Configure hosts with NVMe-oF

ONTAP SAN Host Utilities

NetApp
August 19, 2025

Table of Contents

Configure hosts with NVMe-oF

Overview

You can configure certain SAN hosts for the NVMe over Fabrics (NVMe-oF) protocol, which includes NVMe over Fibre Channel (NVMe/FC) and NVMe over TCP (NVMe/TCP), with ONTAP as the target. Depending on your host operating system and ONTAP version, you configure and validate the NVMe/FC or NVMe/TCP protocol, or both on the host.

Configure AIX with NVMe-oF for ONTAP storage

The IBM AIX and Virtual I/O Server (VIOS)/PowerVM hosts supports the NVMe/FC protocol with Asymmetric Namespace Access (ANA). ANA is equivalent to asymmetric logical unit access (ALUA) multipathing in iSCSI and FCP environments.

For additional details on supported configurations, see the Interoperability Matrix Tool (IMT).

About this task

You can use the following support and features with the NVMe-oF host configuration for AIX hosts. You should also review the known limitations before starting the configuration process.

- Support available:
  - Beginning with ONTAP 9.13.1, NVMe/FC support is added for IBM AIX 7.2 TL5 SP6, AIX 7.3 TL1 SP2, and VIOS 3.1.4.21 with SAN boot support for both physical and virtual stacks. See the IBM documentation for more information on setting up SAN boot support.
  - NVMe/FC is supported with Power9 and Power10 IBM servers.
  - A separate PCM (Path Control Module), such as Host Utilities for AIX SCSI Multipath I/O (MPIO) support, isn't required for NVMe devices.
  - Virtualization support with NetApp (VIOS/PowerVM) is introduced with VIOS 3.1.4.21. This is *only* supported through NPIV (N_PortID Virtualization) storage virtualization mode using the Power10 IBM server.
- Known limitations:
  - Qlogic/Marvel 32G FC HBAs on an AIX host doesn't support NVMe/FC.
  - SAN boot isn't supported for NVMe/FC devices using Power9 IBM server.

Before you begin

- Verify that you have 32GB FC Emulex adapters (EN1A, EN1B, EN1L, EN1M) or 64GB FC adapters (EN1N, EN1P) with adapter firmware 12.4.257.30 and later versions.
- If you have a MetroCluster configuration, NetApp recommends changing the AIX NVMe/FC default APD (All Path Down) time for supporting MetroCluster unplanned switchover events to avoid the AIX operating system enforcing a shorter I/O timeout. For additional information and the recommended changes to default settings, refer to NetApp Bugs Online - 1553249.
- Depending on your AIX version, the Asymmetric Namespace Access Transition Timeout (ANATT) for the AIX host OS is 30 seconds or 60 seconds by default. If the ANATT default for your host is 30 seconds, you need to install an IBM Interim Fix (ifix) from the IBM website that sets the ANATT to 60 seconds to ensure

that all ONTAP workflows are non-disruptive.

> For NVMe/FC AIX support, you must install an ifix on the GA version of the AIX OS. The ifix isn't required for the VIOS/PowerVM OS.
>
> You need to install the ifixes on an AIX version with no previously installed ifixes related to `devices.pciex.pciexclass.010802.rte` on the system. Previously installed ifixes can conflict with the new installation.

Set ANATT to 60 seconds

The default ANATT for the AIX level 72-TL5-SP6-2320 and AIX level 73-TL1-SP2-2320 releases is 30 seconds. IBM provides an ifix that sets the ANATT to 60 seconds. The ifix is available through IBM case ID TS018079082 and you can install it for the following AIX releases:

- For AIX level 72-TL5-SP6-2320, install the `IJ46710s6a.230509.epkg.Z` package.
- For AIX level 73-TL1-SP2-2320, install the `IJ46711s2a.230509.epkg.Z` package.

Default ANATT is 60 seconds

The default ANATT is 60 seconds for the following AIX releases:

- AIX level 73-TL2-SP3-2446
- AIX level 73-TL2-SP2-2420
- AIX level 72-TL5-SP8-2420

Optionally, set ANATT to 120 seconds

IBM provides an ifix that sets the ANATT to 120 seconds. When you set the ANATT to 120 seconds, it enhances performance during ONTAP storage failover events. The ifix is available through IBM case ID TS012877410 and you can install it for the following AIX releases:

- For AIX level 73-TL3-SP0-2446, install the `IJ53487s0a.250130.epkg.Z` package.
- For AIX level 72-TL5-SP9-2446, install the `IJ53445s9a.250130.epkg.Z` package.

> The minimum server firmware version for Power9 servers for NVMe/FC support is FW 950.
>
> The minimum server firmware version for Power10 servers for NVMe/FC support is FW 1010.

For more information on managing ifixes, see [Managing Interim Fixes on AIX].

Step 1: Confirm the multipath configuration for your host

When you install the AIX OS, IBM MPIO used for NVMe multipathing is enabled by default.

Steps

1. Verify that NVMe multipathing is enabled:

```
lsmpio -l hdisk1
```

Show example

```
name     path_id  status   path_status  parent  connection
hdisk1   8        Enabled  Sel,Opt       nvme12  fcnvme0, 9
hdisk1   9        Enabled  Sel,Non       nvme65  fcnvme1, 9
hdisk1   10       Enabled  Sel,Opt       nvme37  fcnvme1, 9
hdisk1   11       Enabled  Sel,Non       nvme60  fcnvme0, 9
```

Step 2: Configure NVMe/FC

You need to configure NVMe/FC for Broadcom/Emulex adapters on VIOS because the NVMe/FC protocol support is disabled in the Virtual Fibre Channel (vFC) on VIOS. The NVMe/FC protocol support is enabled in the physical FC by default.

Steps

1. Verify that you are using the supported adapter.
2. Retrieve a list of virtual adapters:

```
lsmap -all -npiv
```

Show example

```
Name          Physloc                            ClntID ClntName
ClntOS
------------- ---------------------------------- ------
-------------- -------
vfchost0      U9105.22A.785DB61-V2-C2                 4 s1022-iop-
mcc- AIX
Status:LOGGED_IN
FC name:fcs4                    FC loc code:U78DA.ND0.WZS01UY-P0-C7-
T0
Ports logged in:3
Flags:0xea<LOGGED_IN,STRIP_MERGE,SCSI_CLIENT,NVME_CLIENT>
VFC client name:fcs0            VFC client DRC:U9105.22A.785DB61-V4-
C2
```

3. Enable support for the NVMe/FC protocol on an adapter by running the `ioscli vfcctrl` command on the VIOS:

```
vfcctrl -enable -protocol nvme -vadapter vfchost0
```

Example output

```
The "nvme" protocol for "vfchost0" is enabled.
```

4. Verify that the support has been enabled on the adapter:

```
lsattr -El vfchost0
```

Show example

```
alt_site_wwpn        WWPN to use - Only set after migration   False
current_wwpn  0      WWPN to use - Only set after migration   False
enable_nvme   yes    Enable or disable NVME protocol for NPIV True
label                User defined label                       True
limit_intr    false  Limit NPIV Interrupt Sources             True
map_port      fcs4   Physical FC Port                         False
num_per_nvme  0      Number of NPIV NVME queues per range     True
num_per_range 0      Number of NPIV SCSI queues per range     True
```

5. Enable the NVMe/Fc protocol for all adapters:
   a. Change the `dflt_enabl_nvme` attribute value of `viosnpiv0` pseudo device to `yes`.
   b. Set the `enable_nvme` attribute value to `yes` for all the VFC host devices.

```
chdev -l viosnpiv0 -a dflt_enabl_nvme=yes
```

```
lsattr -El viosnpiv0
```

Show example

```
bufs_per_cmd    10  NPIV Number of local bufs per cmd
True
dflt_enabl_nvme yes Default NVME Protocol setting for a new NPIV
adapter True
num_local_cmds  5   NPIV Number of local cmds per channel
True
num_per_nvme    8   NPIV Number of NVME queues per range
True
num_per_range   8   NPIV Number of SCSI queues per range
True
secure_va_info  no  NPIV Secure Virtual Adapter Information
True
```

6. Enable the NVMe/Fc protocol for selected adapters by changing the `enable_nvme` value of the VFC host device attribute to `yes`.
7. Verify that `FC-NVMe Protocol Device` has been created on the server:

```
lsdev |grep fcnvme
```

Exmaple output

```
fcnvme0       Available 00-00-02    FC-NVMe Protocol Device
fcnvme1       Available 00-01-02    FC-NVMe Protocol Device
```

8. Record the host NQN from the server:

```
lsattr -El fcnvme0
```

Show example

```
attach     switch
How this adapter is connected  False
autoconfig available
Configuration State            True
host_nqn   nqn.2014-08.org.nvmexpress:uuid:64e039bd-27d2-421c-858d-
8a378dec31e8 Host NQN (NVMe Qualified Name) True
```

```
lsattr -El fcnvme1
```

Show example

```
attach     switch
How this adapter is connected  False
autoconfig available
Configuration State            True
host_nqn   nqn.2014-08.org.nvmexpress:uuid:64e039bd-27d2-421c-858d-
8a378dec31e8 Host NQN (NVMe Qualified Name) True
```

9. Check the host NQN and verify that it matches the host NQN string for the corresponding subsystem on the ONTAP array:

```
vserver nvme subsystem host show -vserver vs_s922-55-lpar2
```

Example output

```
Vserver         Subsystem                Host NQN
------- ---------
----------------------------------------------------------
vs_s922-55-lpar2 subsystem_s922-55-lpar2 nqn.2014-
08.org.nvmexpress:uuid:64e039bd-27d2-421c-858d-8a378dec31e8
```

10. Verify that the initiator ports are up and running and you can see the target LIFs.

Step 3: Validate NVMe/FC

Verify that the ONTAP namespaces are correct for the NVMe/FC configuration.

Steps

1. Verify that the ONTAP namespaces correctly reflect on the host:

```
lsdev -Cc disk |grep NVMe
```

Example output

```
hdisk1  Available 00-00-02 NVMe 4K Disk
```

2. Optionally, check the multipathing status:

```
lsmpio -l hdisk1
```

Show example

```
name     path_id  status   path_status  parent  connection
hdisk1   8        Enabled  Sel,Opt      nvme12  fcnvme0, 9
hdisk1   9        Enabled  Sel,Non      nvme65  fcnvme1, 9
hdisk1   10       Enabled  Sel,Opt      nvme37  fcnvme1, 9
hdisk1   11       Enabled  Sel,Non      nvme60  fcnvme0, 9
```

Step 4: Review the known issues

The NVMe/FC host configuration for AIX with ONTAP storage has the following known issues:

Burt ID	Title	Description
1553249	AIX NVMe/FC default APD time to be modified for supporting MCC Unplanned Switchover events	By default, AIX operating systems use an all path down (APD) timeout value of 20sec for NVMe/FC. However, ONTAP MetroCluster automatic unplanned switchover (AUSO) and TieBreaker initiated switchover workflows might take a little longer than the APD timeout window, causing I/O errors.
1546017	AIX NVMe/FC caps ANATT at 60s, instead of 120s as advertised by ONTAP	ONTAP advertises the ANA(asymmetric namespace access) transition timeout in controller identify at 120sec. Currently, with ifix, AIX reads the ANA transition timeout from controller identify, but effectively clamps it to 60sec if it is over that limit.
1541386	AIX NVMe/FC hits EIO after ANATT expiry	For any storage failover (SFO) events, if the ANA(asymmetric namespace access) transitioning exceeds the ANA transition timeout cap on a given path, the AIX NVMe/FC host fails with an I/O error despite having alternate healthy paths available to the namespace.
1541380	AIX NVMe/FC waits for half/full ANATT to expire before resuming I/O after ANA AEN	IBM AIX NVMe/FC does not support some Asynchronous notifications (AENs) that ONTAP publishes. This sub-optimal ANA handling will result in sub optimal performance during SFO operations.

Step 5: Troubleshoot

Before troubleshooting any NVMe/FC failures, verify that you are running a configuration that is compliant with the IMT specifications. If you continue to have issues, contact NetApp support.

ESXi

NVMe-oF Host Configuration for ESXi 8.x with ONTAP

You can configure NVMe over Fabrics (NVMe-oF) on initiator hosts running ESXi 8.x and ONTAP as the target.

Supportability

- Beginning with ONTAP 9.16.1 space allocation is enabled by default for all newly created NVMe namespaces.
- Beginning with ONTAP 9.9.1 P3, NVMe/FC protocol is supported for ESXi 8 and later.
- Beginning with ONTAP 9.10.1, NVMe/TCP protocol is supported for ONTAP.

Features

- ESXi initiator hosts can run both NVMe/FC and FCP traffic through the same adapter ports. See the Hardware Universe for a list of supported FC adapters and controllers. See the NetApp Interoperability Matrix Tool for the most current list of supported configurations and versions.
- For ESXi 8.0 and later releases, HPP (high performance plugin) is the default plugin for NVMe devices.

Known limitations

- RDM mapping is not supported.

Enable NVMe/FC

NVMe/FC is enabled by default in vSphere releases.

Verify host NQN

You must check the ESXi host NQN string and verify that it matches with the host NQN string for the corresponding subsystem on the ONTAP array.

```
# esxcli nvme info get
```

Example output:

```
Host NQN: nqn.2014-08.org.nvmexpress:uuid:62a19711-ba8c-475d-c954-
0000c9f1a436
```

```
# vserver nvme subsystem host show -vserver nvme_fc
```

Example output:

```
Vserver Subsystem Host NQN
------- ---------
----------------------------------------------------------
nvme_fc nvme_ss  nqn.2014-08.org.nvmexpress:uuid:62a19711-ba8c-475d-c954-
0000c9f1a436
```

If the host NQN strings do not match, you should use the `vserver nvme subsystem host add` command to update the correct host NQN string on your corresponding ONTAP NVMe subsystem.

Configure Broadcom/Emulex and Marvell/Qlogic

The `lpfc` driver and the `qlnativefc` driver in vSphere 8.x have the NVMe/FC capability enabled by default.

See Interoperability Matrix Tool to check whether the configuration is supported with the driver or firmware.

Validate NVMe/FC

You can use the following procedure to validate NVMe/FC.

Steps

1. Verify that the NVMe/FC adapter is listed on the ESXi host:

   ```
   # esxcli nvme adapter list
   ```

   Example output:

   ```
   Adapter  Adapter Qualified Name           Transport Type  Driver
   Associated Devices
   -------  -------------------------------  --------------  ----------
   ------------------
   vmhba64  aqn:lpfc:100000109b579f11        FC              lpfc
   vmhba65  aqn:lpfc:100000109b579f12        FC              lpfc
   vmhba66  aqn:qlnativefc:2100f4e9d456e286  FC              qlnativefc
   vmhba67  aqn:qlnativefc:2100f4e9d456e287  FC              qlnativefc
   ```

2. Verify that the NVMe/FC namespaces are correctly created:

   The UUIDs in the following example represent the NVMe/FC namespace devices.

```
# esxcfg-mpath -b
uuid.116cb7ed9e574a0faf35ac2ec115969d : NVMe Fibre Channel Disk
(uuid.116cb7ed9e574a0faf35ac2ec115969d)
   vmhba64:C0:T0:L5 LUN:5 state:active fc Adapter: WWNN:
20:00:00:24:ff:7f:4a:50 WWPN: 21:00:00:24:ff:7f:4a:50  Target: WWNN:
20:04:d0:39:ea:3a:b2:1f WWPN: 20:05:d0:39:ea:3a:b2:1f
   vmhba64:C0:T1:L5 LUN:5 state:active fc Adapter: WWNN:
20:00:00:24:ff:7f:4a:50 WWPN: 21:00:00:24:ff:7f:4a:50  Target: WWNN:
20:04:d0:39:ea:3a:b2:1f WWPN: 20:07:d0:39:ea:3a:b2:1f
   vmhba65:C0:T1:L5 LUN:5 state:active fc Adapter: WWNN:
20:00:00:24:ff:7f:4a:51 WWPN: 21:00:00:24:ff:7f:4a:51  Target: WWNN:
20:04:d0:39:ea:3a:b2:1f WWPN: 20:08:d0:39:ea:3a:b2:1f
   vmhba65:C0:T0:L5 LUN:5 state:active fc Adapter: WWNN:
20:00:00:24:ff:7f:4a:51 WWPN: 21:00:00:24:ff:7f:4a:51  Target: WWNN:
20:04:d0:39:ea:3a:b2:1f WWPN: 20:06:d0:39:ea:3a:b2:1f
```

In ONTAP 9.7, the default block size for an NVMe/FC namespace is 4K. This default size is not compatible with ESXi. Therefore, when creating namespaces for ESXi, you must set the namespace block size as **512B**. You can do this using the `vserver nvme namespace create` command.

Example,

```
vserver nvme namespace create -vserver vs_1 -path
/vol/nsvol/namespace1 -size 100g -ostype vmware -block-size 512B
```

Refer to the ONTAP 9 Command man pages for additional details.

3. Verify the status of the individual ANA paths of the respective NVMe/FC namespace devices:

```
# esxcli storage hpp path list -d uuid.df960bebb5a74a3eaaa1ae55e6b3411d

fc.20000024ff7f4a50:21000024ff7f4a50-
fc.2004d039ea3ab21f:2005d039ea3ab21f-
uuid.df960bebb5a74a3eaaa1ae55e6b3411d
   Runtime Name: vmhba64:C0:T0:L3
   Device: uuid.df960bebb5a74a3eaaa1ae55e6b3411d
   Device Display Name: NVMe Fibre Channel Disk
(uuid.df960bebb5a74a3eaaa1ae55e6b3411d)
   Path State: active unoptimized
   Path Config: {ANA_GRP_id=4,ANA_GRP_state=ANO,health=UP}

fc.20000024ff7f4a51:21000024ff7f4a51-
fc.2004d039ea3ab21f:2008d039ea3ab21f-
uuid.df960bebb5a74a3eaaa1ae55e6b3411d
   Runtime Name: vmhba65:C0:T1:L3
   Device: uuid.df960bebb5a74a3eaaa1ae55e6b3411d
   Device Display Name: NVMe Fibre Channel Disk
(uuid.df960bebb5a74a3eaaa1ae55e6b3411d)
   Path State: active
   Path Config: {ANA_GRP_id=4,ANA_GRP_state=AO,health=UP}

fc.20000024ff7f4a51:21000024ff7f4a51-
fc.2004d039ea3ab21f:2006d039ea3ab21f-
uuid.df960bebb5a74a3eaaa1ae55e6b3411d
   Runtime Name: vmhba65:C0:T0:L3
   Device: uuid.df960bebb5a74a3eaaa1ae55e6b3411d
   Device Display Name: NVMe Fibre Channel Disk
(uuid.df960bebb5a74a3eaaa1ae55e6b3411d)
   Path State: active unoptimized
   Path Config: {ANA_GRP_id=4,ANA_GRP_state=ANO,health=UP}

fc.20000024ff7f4a50:21000024ff7f4a50-
fc.2004d039ea3ab21f:2007d039ea3ab21f-
uuid.df960bebb5a74a3eaaa1ae55e6b3411d
   Runtime Name: vmhba64:C0:T1:L3
   Device: uuid.df960bebb5a74a3eaaa1ae55e6b3411d
   Device Display Name: NVMe Fibre Channel Disk
(uuid.df960bebb5a74a3eaaa1ae55e6b3411d)
   Path State: active
   Path Config: {ANA_GRP_id=4,ANA_GRP_state=AO,health=UP}
```

Configure NVMe/TCP

In ESXi 8.x, the required NVMe/TCP modules are loaded by default. To configure the network and the

NVMe/TCP adapter, refer to the VMware vSphere documentation.

Validate NVMe/TCP

You can use the following procedure to validate NVMe/TCP.

Steps

1. Verify the status of the NVMe/TCP adapter:

   ```
   esxcli nvme adapter list
   ```

   Example output:

   ```
   Adapter  Adapter Qualified Name           Transport Type  Driver
   Associated Devices
   -------  -------------------------------  --------------  -------
   ------------------
   vmhba65  aqn:nvmetcp:ec-2a-72-0f-e2-30-T  TCP             nvmetcp
   vmnic0
   vmhba66  aqn:nvmetcp:34-80-0d-30-d1-a0-T  TCP             nvmetcp
   vmnic2
   vmhba67  aqn:nvmetcp:34-80-0d-30-d1-a1-T  TCP             nvmetcp
   vmnic3
   ```

2. Retrieve a list of NVMe/TCP connections:

   ```
   esxcli nvme controller list
   ```

   Example output:

```
Name                                                     Controller Number
Adapter  Transport Type  Is Online  Is VVOL
---------------------------------------------------------------------------------------------------------  -----------------  -------
nqn.2014-08.org.nvmexpress.discovery#vmhba64#192.168.100.166:8009  256
vmhba64  TCP                     true    false
nqn.1992-
08.com.netapp:sn.89bb1a28a89a11ed8a88d039ea263f93:subsystem.nvme_ss#vmhb
a64#192.168.100.165:4420 258  vmhba64  TCP  true    false
nqn.1992-
08.com.netapp:sn.89bb1a28a89a11ed8a88d039ea263f93:subsystem.nvme_ss#vmhb
a64#192.168.100.168:4420 259  vmhba64  TCP  true    false
nqn.1992-
08.com.netapp:sn.89bb1a28a89a11ed8a88d039ea263f93:subsystem.nvme_ss#vmhb
a64#192.168.100.166:4420 260  vmhba64  TCP  true    false
nqn.2014-08.org.nvmexpress.discovery#vmhba64#192.168.100.165:8009  261
vmhba64  TCP                     true    false
nqn.2014-08.org.nvmexpress.discovery#vmhba65#192.168.100.155:8009  262
vmhba65  TCP                     true    false
nqn.1992-
08.com.netapp:sn.89bb1a28a89a11ed8a88d039ea263f93:subsystem.nvme_ss#vmhb
a64#192.168.100.167:4420 264  vmhba64  TCP  true    false
```

3. Retrieve a list of the number of paths to an NVMe namespace:

```
esxcli storage hpp path list -d uuid.f4f14337c3ad4a639edf0e21de8b88bf
```

Example output:

```
tcp.vmnic2:34:80:0d:30:ca:e0-tcp.192.168.100.165:4420-
uuid.f4f14337c3ad4a639edf0e21de8b88bf
   Runtime Name: vmhba64:C0:T0:L5
   Device: uuid.f4f14337c3ad4a639edf0e21de8b88bf
   Device Display Name: NVMe TCP Disk
(uuid.f4f14337c3ad4a639edf0e21de8b88bf)
   Path State: active
   Path Config: {ANA_GRP_id=6,ANA_GRP_state=AO,health=UP}

tcp.vmnic2:34:80:0d:30:ca:e0-tcp.192.168.100.168:4420-
uuid.f4f14337c3ad4a639edf0e21de8b88bf
   Runtime Name: vmhba64:C0:T3:L5
   Device: uuid.f4f14337c3ad4a639edf0e21de8b88bf
   Device Display Name: NVMe TCP Disk
(uuid.f4f14337c3ad4a639edf0e21de8b88bf)
   Path State: active unoptimized
   Path Config: {ANA_GRP_id=6,ANA_GRP_state=ANO,health=UP}

tcp.vmnic2:34:80:0d:30:ca:e0-tcp.192.168.100.166:4420-
uuid.f4f14337c3ad4a639edf0e21de8b88bf
   Runtime Name: vmhba64:C0:T2:L5
   Device: uuid.f4f14337c3ad4a639edf0e21de8b88bf
   Device Display Name: NVMe TCP Disk
(uuid.f4f14337c3ad4a639edf0e21de8b88bf)
   Path State: active unoptimized
   Path Config: {ANA_GRP_id=6,ANA_GRP_state=ANO,health=UP}

tcp.vmnic2:34:80:0d:30:ca:e0-tcp.192.168.100.167:4420-
uuid.f4f14337c3ad4a639edf0e21de8b88bf
   Runtime Name: vmhba64:C0:T1:L5
   Device: uuid.f4f14337c3ad4a639edf0e21de8b88bf
   Device Display Name: NVMe TCP Disk
(uuid.f4f14337c3ad4a639edf0e21de8b88bf)
   Path State: active
   Path Config: {ANA_GRP_id=6,ANA_GRP_state=AO,health=UP}
```

NVMe deallocate

The NVMe deallocate command is supported for ESXi 8.0u2 and later with ONTAP 9.16.1 and later.

Deallocate support is always enabled for NVMe namespaces. Deallocate also allows the guest OS to perform 'UNMAP' (sometimes called 'TRIM') operations on VMFS datastores. Deallocate operations allow a host to identify blocks of data that are no longer required because they no longer contain valid data. The storage system can then remove those data blocks so that the space can be consumed elsewhere.

Steps

1. On your ESXi host, verify the setting for DSM deallocate with TP4040 support:

   ```
   esxcfg-advcfg -g /Scsi/NVmeUseDsmTp4040
   ```

   The expected value is 0.

2. Enable the setting for DSM deallocate with TP4040 support:

   ```
   esxcfg-advcfg -s 1 /Scsi/NvmeUseDsmTp4040
   ```

3. Verify that the setting for DSM deallocate with TP4040 support is enabled:

   ```
   esxcfg-advcfg -g /Scsi/NVmeUseDsmTp4040
   ```

   The expected value is 1.

For more information on NVMe deallocate in VMware vSphere, refer to Storage Space Reclamation in vSphere

Known issues

The NVMe-oF host configuration for ESXi 8.x with ONTAP has the following known issues:

NetApp Bug ID	Title	Description
1420654	ONTAP node non-operational when NVMe/FC protocol is used with ONTAP version 9.9.1	ONTAP 9.9.1 has introduced support for the NVMe "abort" command. When ONTAP receives the "abort" command to abort an NVMe fused command that is waiting for its partner command, an ONTAP node disruption occurs. The issue is noticed only with hosts that use NVMe fused commands (for example, ESX) and Fibre Channel (FC) transport.
1543660	I/O error occurs when Linux VMs using vNVMe adapters encounter a long all paths down (APD) window	Linux VMs running vSphere 8.x and later and using virtual NVMe (vNVME) adapters encounter an I/O error because the vNVMe retry operation is disabled by default. To avoid a disruption on Linux VMs running older kernels during an all paths down (APD) or a heavy I/O load, VMware has introduced a tunable "VSCSIDisableNvmeRetry" to disable the vNVMe retry operation.

Related information

VMware vSphere with ONTAP
VMware vSphere 5.x, 6.x and 7.x support with NetApp MetroCluster (2031038)
VMware vSphere 6.x and 7.x support with NetApp SnapMirror active sync

NVMe-oF Host Configuration for ESXi 7.x with ONTAP

You can configure NVMe over Fabrics (NVMe-oF) on initiator hosts running ESXi 7.x and ONTAP as the target.

Supportability

- Beginning with ONTAP 9.7, NVMe over Fibre Channel (NVMe/FC) support is added for VMWare vSphere releases.
- Beginning with 7.0U3c, NVMe/TCP feature is supported for ESXi Hypervisor.
- Beginning with ONTAP 9.10.1, NVMe/TCP feature is supported for ONTAP.

Features

- ESXi initiator host can run both NVMe/FC and FCP traffic through the same adapter ports. See the Hardware Universe for a list of supported FC adapters and controllers. See the Interoperability Matrix Tool for the current list of supported configurations and versions.
- Beginning with ONTAP 9.9.1 P3, NVMe/FC feature is supported for ESXi 7.0 update 3.
- For ESXi 7.0 and later releases, HPP (high performance plugin) is the default plugin for NVMe devices.

Known limitations

The following configurations are not supported:

- RDM mapping
- VVols

Enable NVMe/FC

1. Check the ESXi host NQN string and verify that it matches with the host NQN string for the corresponding subsystem on the ONTAP array:

```
# esxcli nvme  info get
Host NQN: nqn.2014-08.com.vmware:nvme:nvme-esx

# vserver nvme subsystem host show -vserver vserver_nvme
  Vserver Subsystem             Host NQN
  ------- ------------------- ----------------------------------------
  vserver_nvme ss_vserver_nvme nqn.2014-08.com.vmware:nvme:nvme-esx
```

Configure Broadcom/Emulex

1. Check whether the configuration is supported with required driver/firmware by referring to Interoperability Matrix Tool.
2. Set the lpfc driver parameter `lpfc_enable_fc4_type=3` for enabling NVMe/FC support in the `lpfc` driver and reboot the host.

Starting with vSphere 7.0 update 3, the `brcmnvmefc` driver is no longer available. Therefore, the `lpfc` driver now includes the NVMe over Fibre Channel (NVMe/FC) functionality previously delivered with the `brcmnvmefc` driver.

The `lpfc_enable_fc4_type=3` parameter is set by default for the LPe35000-series adapters. You must perform the following command to set it manually for LPe32000-series and LPe31000-series adapters.

```
# esxcli system module parameters set -m lpfc -p lpfc_enable_fc4_type=3

#esxcli system module parameters list  -m lpfc | grep lpfc_enable_fc4_type
lpfc_enable_fc4_type              int     3      Defines what FC4 types
are supported

#esxcli storage core adapter list
HBA Name  Driver   Link State  UID
Capabilities         Description
--------  -------  ----------  ------------------------------------
-------------------  -----------
vmhba1    lpfc     link-up     fc.200000109b95456f:100000109b95456f
Second Level Lun ID  (0000:86:00.0) Emulex Corporation Emulex LPe36000
Fibre Channel Adapter    FC HBA
vmhba2    lpfc     link-up     fc.200000109b954570:100000109b954570
Second Level Lun ID  (0000:86:00.1) Emulex Corporation Emulex LPe36000
Fibre Channel Adapter    FC HBA
vmhba64   lpfc     link-up     fc.200000109b95456f:100000109b95456f
(0000:86:00.0) Emulex Corporation Emulex LPe36000 Fibre Channel Adapter
NVMe HBA
vmhba65   lpfc     link-up     fc.200000109b954570:100000109b954570
(0000:86:00.1) Emulex Corporation Emulex LPe36000 Fibre Channel Adapter
NVMe HBA
```

Configure Marvell/QLogic

Steps

1. Check whether configuration is supported with required driver/firmware by referring to Interoperability Matrix Tool.
2. Set the `qlnativefc` driver parameter `ql2xnvmesupport=1` for enabling NVMe/FC support in the `qlnativefc` driver and reboot the host.

   ```
   # esxcfg-module -s 'ql2xnvmesupport=1' qlnativefc
   ```

   The `qlnativefc` driver parameter is set by default for the Qle 277x-series adapters. You must perform the following command to set it manually for Qle 277x series adapters.

```
esxcfg-module -l | grep qlnativefc
qlnativefc               4    1912
```

3. Check whether nvme is enabled on the adapter:

```
  #esxcli storage core adapter list
HBA Name  Driver      Link State  UID
Capabilities         Description
--------  ----------  ----------  ------------------------------------
-------------------  -----------
 vmhba3    qlnativefc  link-up     fc.20000024ff1817ae:21000024ff1817ae
Second Level Lun ID  (0000:5e:00.0) QLogic Corp QLE2742 Dual Port 32Gb
Fibre Channel to PCIe Adapter    FC Adapter
vmhba4    qlnativefc  link-up     fc.20000024ff1817af:21000024ff1817af
Second Level Lun ID  (0000:5e:00.1) QLogic Corp QLE2742 Dual Port 32Gb
Fibre Channel to PCIe Adapter FC Adapter
vmhba64   qlnativefc  link-up     fc.20000024ff1817ae:21000024ff1817ae
(0000:5e:00.0) QLogic Corp QLE2742 Dual Port 32Gb Fibre Channel to PCIe
Adapter  NVMe FC Adapter
vmhba65   qlnativefc  link-up     fc.20000024ff1817af:21000024ff1817af
(0000:5e:00.1) QLogic Corp QLE2742 Dual Port 32Gb Fibre Channel to PCIe
Adapter  NVMe FC Adapter
```

Validate NVMe/FC

1. Verify that NVMe/FC adapter is listed on the ESXi host:

```
# esxcli nvme adapter list

Adapter  Adapter Qualified Name           Transport Type  Driver
Associated Devices
-------  -------------------------------  --------------  ----------
------------------
vmhba64  aqn:qlnativefc:21000024ff1817ae  FC              qlnativefc
vmhba65  aqn:qlnativefc:21000024ff1817af  FC              qlnativefc
vmhba66  aqn:lpfc:100000109b579d9c        FC              lpfc
vmhba67  aqn:lpfc:100000109b579d9d        FC              lpfc
```

2. Verify that the NVMe/FC namespaces are properly created:

   The UUIDs in the following example represent the NVMe/FC namespace devices.

```
# esxcfg-mpath -b
uuid.5084e29a6bb24fbca5ba076eda8ecd7e : NVMe Fibre Channel Disk
(uuid.5084e29a6bb24fbca5ba076eda8ecd7e)
   vmhba65:C0:T0:L1 LUN:1 state:active fc Adapter: WWNN:
20:00:34:80:0d:6d:72:69 WWPN: 21:00:34:80:0d:6d:72:69  Target: WWNN:
20:17:00:a0:98:df:e3:d1 WWPN: 20:2f:00:a0:98:df:e3:d1
   vmhba65:C0:T1:L1 LUN:1 state:active fc Adapter: WWNN:
20:00:34:80:0d:6d:72:69 WWPN: 21:00:34:80:0d:6d:72:69  Target: WWNN:
20:17:00:a0:98:df:e3:d1 WWPN: 20:1a:00:a0:98:df:e3:d1
   vmhba64:C0:T0:L1 LUN:1 state:active fc Adapter: WWNN:
20:00:34:80:0d:6d:72:68 WWPN: 21:00:34:80:0d:6d:72:68  Target: WWNN:
20:17:00:a0:98:df:e3:d1 WWPN: 20:18:00:a0:98:df:e3:d1
   vmhba64:C0:T1:L1 LUN:1 state:active fc Adapter: WWNN:
20:00:34:80:0d:6d:72:68 WWPN: 21:00:34:80:0d:6d:72:68  Target: WWNN:
20:17:00:a0:98:df:e3:d1 WWPN: 20:19:00:a0:98:df:e3:d1
```

In ONTAP 9.7, the default block size for a NVMe/FC namespace is 4K. This default size is not compatible with ESXi. Therefore, when creating namespaces for ESXi, you must set the namespace block size as 512b. You can do this using the `vserver nvme namespace create` command.

Example

`vserver nvme namespace create -vserver vs_1 -path /vol/nsvol/namespace1 -size 100g -ostype vmware -block-size 512B`

Refer to the ONTAP 9 Command man pages for additional details.

3. Verify the status of the individual ANA paths of the respective NVMe/FC namespace devices:

```
esxcli storage hpp path list -d uuid.5084e29a6bb24fbca5ba076eda8ecd7e
fc.200034800d6d7268:210034800d6d7268-
fc.201700a098dfe3d1:201800a098dfe3d1-
uuid.5084e29a6bb24fbca5ba076eda8ecd7e
   Runtime Name: vmhba64:C0:T0:L1
   Device: uuid.5084e29a6bb24fbca5ba076eda8ecd7e
   Device Display Name: NVMe Fibre Channel Disk
(uuid.5084e29a6bb24fbca5ba076eda8ecd7e)
   Path State: active
   Path Config: {TPG_id=0,TPG_state=AO,RTP_id=0,health=UP}

fc.200034800d6d7269:210034800d6d7269-
fc.201700a098dfe3d1:201a00a098dfe3d1-
uuid.5084e29a6bb24fbca5ba076eda8ecd7e
   Runtime Name: vmhba65:C0:T1:L1
   Device: uuid.5084e29a6bb24fbca5ba076eda8ecd7e
   Device Display Name: NVMe Fibre Channel Disk
(uuid.5084e29a6bb24fbca5ba076eda8ecd7e)
   Path State: active
   Path Config: {TPG_id=0,TPG_state=AO,RTP_id=0,health=UP}

fc.200034800d6d7269:210034800d6d7269-
fc.201700a098dfe3d1:202f00a098dfe3d1-
uuid.5084e29a6bb24fbca5ba076eda8ecd7e
   Runtime Name: vmhba65:C0:T0:L1
   Device: uuid.5084e29a6bb24fbca5ba076eda8ecd7e
   Device Display Name: NVMe Fibre Channel Disk
(uuid.5084e29a6bb24fbca5ba076eda8ecd7e)
   Path State: active unoptimized
   Path Config: {TPG_id=0,TPG_state=ANO,RTP_id=0,health=UP}

fc.200034800d6d7268:210034800d6d7268-
fc.201700a098dfe3d1:201900a098dfe3d1-
uuid.5084e29a6bb24fbca5ba076eda8ecd7e
   Runtime Name: vmhba64:C0:T1:L1
   Device: uuid.5084e29a6bb24fbca5ba076eda8ecd7e
   Device Display Name: NVMe Fibre Channel Disk
(uuid.5084e29a6bb24fbca5ba076eda8ecd7e)
   Path State: active unoptimized
   Path Config: {TPG_id=0,TPG_state=ANO,RTP_id=0,health=UP}
```

Configure NVMe/TCP

Starting 7.0U3c, the required NVMe/TCP modules will be loaded by default. For configuring the network and the NVMe/TCP adapter, refer to the VMware vSphere documentation.

Validate NVMe/TCP

Steps

1. Verify the status of the NVMe/TCP adapter.

```
[root@R650-8-45:~] esxcli nvme adapter list
Adapter    Adapter Qualified Name
--------- -------------------------------
vmhba64    aqn:nvmetcp:34-80-0d-30-ca-e0-T
vmhba65    aqn:nvmetc:34-80-13d-30-ca-e1-T
list
Transport Type   Driver   Associated Devices
---------------  -------  ------------------
TCP              nvmetcp    vmnzc2
TCP              nvmetcp    vmnzc3
```

2. To list the NVMe/TCP connections, use the following command:

```
[root@R650-8-45:~] esxcli nvme controller list
Name
-----------
nqn.1992-
08.com.netapp:sn.5e347cf68e0511ec9ec2d039ea13e6ed:subsystem.vs_name_tcp_
ss#vmhba64#192.168.100.11:4420
nqn.1992-
08.com.netapp:sn.5e347cf68e0511ec9ec2d039ea13e6ed:subsystem.vs_name_tcp_
ss#vmhba64#192.168.101.11:4420
Controller Number  Adapter   Transport Type   IS Online
----------------- ---------  ---------------  ---------
1580              vmhba64    TCP              true
1588              vmhba65    TCP              true
```

3. To list the number of paths to an NVMe namespace, use the following command:

```
[root@R650-8-45:~] esxcli storage hpp path list -d
uuid.400bf333abf74ab8b96dc18ffadc3f99
tcp.vmnic2:34:80:0d:30:ca:eo-tcp.unknown-
uuid.400bf333abf74ab8b96dc18ffadc3f99
   Runtime Name: vmhba64:C0:T0:L3
   Device: uuid.400bf333abf74ab8b96dc18ffadc3f99
   Device Display Name: NVMe TCP Disk
(uuid.400bf333abf74ab8b96dc18ffadc3f99)
   Path State: active unoptimized
   Path config: {TPG_id=0,TPG_state=ANO,RTP_id=0,health=UP}

tcp.vmnic3:34:80:0d:30:ca:el-tcp.unknown-
uuid.400bf333abf74ab8b96dc18ffadc3f99
   Runtime Name: vmhba65:C0:T1:L3
   Device: uuid.400bf333abf74ab8b96dc18ffadc3f99
   Device Display Name: NVMe TCP Disk
(uuid.400bf333abf74ab8b96dc18ffadc3f99)
   Path State: active
   Path config: {TPG_id=0,TPG_state=AO,RTP_id=0,health=UP}
```

Known issues

The NVMe-oF host configuration for ESXi 7.x with ONTAP has the following known issues:

NetApp Bug ID	Title	Workaround
1420654	ONTAP node non-operational when NVMe/FC protocol is used with ONTAP version 9.9.1	Check and rectify any network issues in the host fabric. If this does not help, upgrade to a patch that fixes this issue.

Related information

VMware vSphere with ONTAP
VMware vSphere 5.x, 6.x and 7.x support with NetApp MetroCluster (2031038)
VMware vSphere 6.x and 7.x support with NetApp® SnapMirror active sync

Oracle Linux

Oracle Linux 9

Configure Oracle Linux 9.5 with NVMe-oF for ONTAP storage

NetApp SAN host configurations support the NVMe over Fabrics (NVMe-oF) protocol with Asymmetric Namespace Access (ANA). In NVMe-oF environments, ANA is equivalent to asymmetric logical unit access (ALUA) multipathing in iSCSI and FCP environments. ANA is implemented using the in-kernel NVMe multipath feature.

About this task

The following support and features are available with the NVMe-oF host configuration for Oracle Linux 9.5 with ONTAP storage.

- Support available:
  - Support for NVMe over TCP (NVMe/TCP) in addition to NVMe over Fibre Channel (NVMe/FC). The NetApp plug-in in the native nvme-cli package displays ONTAP details for both NVMe/FC and NVMe/TCP namespaces.
  - Running both NVMe and SCSI traffic on the same host. For example, you can configure dm-multipath on SCSI mpath devices for SCSI LUNs and use NVMe multipath to configure NVMe-oF namespace devices on the host.
  - Oracle Linux 9.5 enables in-kernel NVMe multipath for NVMe namespaces by default, removing the need for explicit settings.
  - Beginning with ONTAP 9.12.1, support for secure in-band authentication is introduced for NVMe/TCP. You can use secure in-band authentication for NVMe/TCP with Oracle Linux 9.5.

  > The NetApp `sanlun` host utility isn't supported for NVMe-oF. Instead, you can use the NetApp plug-in included in the native `nvme-cli` for all NVMe-oF transports.

  For additional details on supported configurations, see the Interoperability Matrix Tool.

- Features available:
  - There are no new features in this release.
- Known limitations:
  - Avoid issuing the `nvme disconnect-all` command on systems booting from SAN over NVMe-TCP or NVMe-FC namespaces because it disconnects both root and data filesystems and might lead to system instability.

Step 1: Optionally, enable SAN booting

You can configure your host to use SAN booting to simplify deployment and improve scalability.

Before you begin

Use the Interoperability Matrix Tool to verify that your Linux OS, host bus adapter (HBA), HBA firmware, HBA boot BIOS, and ONTAP version support SAN booting.

Steps

1. Create a SAN boot namespace and map it to the host.
2. Enable SAN booting in the server BIOS for the ports to which the SAN boot namespace is mapped.

   For information on how to enable the HBA BIOS, see your vendor-specific documentation.

3. Verify that the configuration was successful by rebooting the host and verifying that the OS is up and running.

Step 2: Validate software versions

Use the following procedure to validate the minimum supported Oracle Linux 9.5 software versions.

Steps

1. Install Oracle Linux 9.5 on the server. After the installation is complete, verify that you are running the specified Oracle Linux 9.5 kernel.

   ```
   uname -r
   ```

   The following example shows an Oracle Linux kernel version:

   ```
   5.15.0-302.167.6.el9uek.x86_64
   ```

2. Install the `nvme-cli` package:

   ```
   rpm -qa|grep nvme-cli
   ```

   The following example shows an `nvme-cli` package version:

   ```
   nvme-cli-2.9.1-6.el9.x86_64
   ```

3. Install the `libnvme` package:

   ```
   rpm -qa|grep libnvme
   ```

   The following example shows an `libnvme` package version:

   ```
   libnvme-1.9-3.el9.x86_64
   ```

4. On the Oracle Linux 9.5 host, check the `hostnqn` string at `/etc/nvme/hostnqn`:

   ```
   cat /etc/nvme/hostnqn
   ```

   The following example shows an `hostnqn` version:

   ```
   nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0050-3410-8035-c2c04f4a5933
   ```

5. Verify that the `hostnqn` string matches the `hostnqn` string for the corresponding subsystem on the ONTAP array:

   ```
   vserver nvme subsystem host show -vserver vs_213_36002
   ```

Show example

```
Vserver Subsystem Priority  Host NQN
------- --------- --------
------------------------------------------------
vs_coexistence_LPE36002
        nvme1
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0050-3410-8035-c2c04f4a5933
        nvme2
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0050-3410-8035-c2c04f4a5933
        nvme3
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0050-3410-8035-c2c04f4a5933
        nvme4
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0050-3410-8035-c2c04f4a5933
4 entries were displayed.
```

If the `hostnqn` strings don't match, you can use the `vserver modify` command to update the `hostnqn` string on your corresponding ONTAP array subsystem to match the `hostnqn` string from `/etc/nvme/hostnqn` on the host.

Step 3: Configure NVMe/FC

Configure NVMe/FC with Broadcom/Emulex FC or Marvell/Qlogic FC adapters.

Broadcom/Emulex

Configure NVMe/FC for a Broadcom/Emulex adapter.

Steps

1. Verify that you’re using the supported adapter model:

   a. Display the model names:

   ```
   cat /sys/class/scsi_host/host*/modelname
   ```

   You should see the following output:

   ```
   LPe36002-M64
   LPe36002-M64
   ```

   b. Display the model descriptions:

   ```
   cat /sys/class/scsi_host/host*/modeldesc
   ```

   You should see an output similar to the following example:

   ```
   Emulex LightPulse LPe36002-M64 2-Port 64Gb Fibre Channel Adapter
   Emulex LightPulse LPe36002-M64 2-Port 64Gb Fibre Channel Adapter
   ```

2. Verify that you are using the recommended Broadcom `lpfc` firmware and inbox driver:

   a. Display the firmware version:

   ```
   cat /sys/class/scsi_host/host*/fwrev
   ```

   The following example shows firmware versions:

   ```
   14.4.393.25, sli-4:6:d
   14.4.393.25, sli-4:6:d
   ```

   b. Display the inbox driver version:

   ```
   cat /sys/module/lpfc/version
   ```

   The following example shows a driver version:

```
0:14.4.0.2
```

For the current list of supported adapter driver and firmware versions, see the Interoperability Matrix Tool.

3. Verify that `lpfc_enable_fc4_type` is set to `3`:

   ```
   cat /sys/module/lpfc/parameters/lpfc_enable_fc4_type
   ```

4. Verify that you can view your initiator ports:

   ```
   cat /sys/class/fc_host/host*/<port_name>
   ```

   The following example shows port identities:

   ```
   0x100000620b3c089c
   0x100000620b3c089d
   ```

5. Verify that your initiator ports are online:

   ```
   cat /sys/class/fc_host/host*/port_state
   ```

   You should see the following output:

   ```
   Online
   Online
   ```

6. Verify that the NVMe/FC initiator ports are enabled and that the target ports are visible:

   ```
   cat /sys/class/scsi_host/host*/nvme_info
   ```

Show example

```
NVME Initiator Enabled
XRI Dist lpfc0 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc0 WWPN x100000620b3c089c WWNN x200000620b3c089c
DID x081300 ONLINE
NVME RPORT       WWPN x2001d039eab0dadc WWNN x2000d039eab0dadc
DID x080101 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2003d039eab0dadc WWNN x2000d039eab0dadc
DID x080401 TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 00000002e9 Cmpl 00000002e9 Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 0000000000078742 Issue 0000000000078740 OutIO
fffffffffffffffe
        abort 000000c2 noxri 00000000 nondlp 00000a23 qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 000000c2 Err 00000238

NVME Initiator Enabled
XRI Dist lpfc1 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc1 WWPN x100000620b3c089d WWNN x200000620b3c089d
DID x081900 ONLINE
NVME RPORT       WWPN x2002d039eab0dadc WWNN x2000d039eab0dadc
DID x080201 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2004d039eab0dadc WWNN x2000d039eab0dadc
DID x080301 TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 00000002d9 Cmpl 00000002d9 Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000000007754f Issue 000000000007754f OutIO
0000000000000000
        abort 000000c2 noxri 00000000 nondlp 00000719 qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 000000c2 Err 0000023d
```

Marvell/QLogic

Configure NVMe/FC for a Marvell/QLogic adapter.

Steps

1. Verify that you are running the supported adapter driver and firmware versions:

```
cat /sys/class/fc_host/host*/symbolic_name
```

The follow example shows driver and firware versions:

```
QLE2772 FW:v9.15.03 DVR:v10.02.09.300-k-debug
```

2. Verify that `ql2xnvmeenable` is set. This enables the Marvell adapter to function as an NVMe/FC initiator:

```
cat /sys/module/qla2xxx/parameters/ql2xnvmeenable
```

The value `1` verifies that `ql2xnvmeenable` is set.

Step 4: Optionally, enable 1MB I/O for NVMe/FC

ONTAP reports an MDTS (Max Data Transfer Size) of 8 in the Identify Controller data. This means the maximum I/O request size can be up to 1MB. To issue I/O requests of size 1MB for a Broadcom NVMe/FC host, you should increase the `lpfc` value of the `lpfc_sg_seg_cnt` parameter to 256 from the default value of 64.

> These steps don’t apply to Qlogic NVMe/FC hosts.

Steps

1. Set the `lpfc_sg_seg_cnt` parameter to 256:

```
cat /etc/modprobe.d/lpfc.conf
```

You should see an output similar to the following example:

```
options lpfc lpfc_sg_seg_cnt=256
```

2. Run the `dracut -f` command, and reboot the host.
3. Verify that the value for `lpfc_sg_seg_cnt` is 256:

```
cat /sys/module/lpfc/parameters/lpfc_sg_seg_cnt
```

Step 5: Verify NVMe boot services

With Oracle Linux 9.5, the `nvmefc-boot-connections.service` and `nvmf-autoconnect.service` boot services included in the NVMe/FC `nvme-cli` package are automatically enabled when the system boots.

After booting completes, verify that the `nvmefc-boot-connections.service` and `nvmf-autoconnect.service` boot services are enabled.

Steps

1. Verify that `nvmf-autoconnect.service` is enabled:

```
systemctl status nvmf-autoconnect.service
```

Show example output

```
nvmf-autoconnect.service - Connect NVMe-oF subsystems automatically
during boot
Loaded: loaded (/usr/lib/systemd/system/nvmf-autoconnect.service;
enabled; preset: disabled)
Active: inactive (dead) since Wed 2025-07-02 16:46:37 IST; 1 day 3h
ago
Main PID: 2129 (code=exited, status=0/SUCCESS)
CPU: 121ms

Jul 02 16:46:37 interop-13-175 nvme[2129]: Failed to write to
/dev/nvme-fabrics: Invalid argument
Jul 02 16:46:37 interop-13-175 nvme[2129]: Failed to write to
/dev/nvme-fabrics: Invalid argument
Jul 02 16:46:37 interop-13-175 nvme[2129]: Failed to write to
/dev/nvme-fabrics: Invalid argument
Jul 02 16:46:37 interop-13-175 nvme[2129]: Failed to write to
/dev/nvme-fabrics: Invalid argument
Jul 02 16:46:37 interop-13-175 nvme[2129]: Failed to write to
/dev/nvme-fabrics: Invalid argument
Jul 02 16:46:37 interop-13-175 nvme[2129]: Failed to write to
/dev/nvme-fabrics: Invalid argument
Jul 02 16:46:37 interop-13-175 nvme[2129]: Failed to open ctrl
nvme0, errno 2
Jul 02 16:46:37 interop-13-175 nvme[2129]: failed to get discovery
log: Bad file descriptor
Jul 02 16:46:37 interop-13-175 systemd[1]: nvmf-autoconnect.service:
Deactivated successfully.
Jul 02 16:46:37 interop-13-175 systemd[1]: Finished Connect NVMe-oF
subsystems automatically during boot.
```

2. Verify that `nvmefc-boot-connections.service` is enabled:

```
systemctl status nvmefc-boot-connections.service
```

Show example output

```
nvmefc-boot-connections.service - Auto-connect to subsystems on FC-
NVME devices found during boot
Loaded: loaded (/usr/lib/systemd/system/nvmefc-boot-
connections.service; enabled; preset: enabled)
Active: inactive (dead) since Wed 2025-07-02 16:45:46 IST; 1 day 3h
ago
Main PID: 1604 (code=exited, status=0/SUCCESS)
CPU: 32ms

Jul 02 16:45:46 interop-13-175 systemd[1]: Starting Auto-connect to
subsystems on FC-NVME devices found during boot...
Jul 02 16:45:46 interop-13-175 systemd[1]: nvmefc-boot-
connections.service: Deactivated successfully.
Jul 02 16:45:46 interop-13-175 systemd[1]: Finished Auto-connect to
subsystems on FC-NVME devices found during boot.
```

Step 6: Configure NVMe/TCP

The NVMe/TCP protocol doesn’t support the `auto-connect` operation. Instead, you can discover the NVMe/TCP subsystems and namespaces by performing the NVMe/TCP `connect` or `connect-all` operations manually.

Steps

1. Verify that the initiator port can fetch the discovery log page data across the supported NVMe/TCP LIFs:

```
nvme discover -t tcp -w host-traddr -a traddr
```

Show example

```
nvme discover -t tcp -w 192.168.165.3 -a 192.168.165.8
Discovery Log Number of Records 8, Generation counter 8
=====Discovery Log Entry 0======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  4
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.4f7af2bd221811f0afadd039eab0dadd:discovery
traddr:  192.168.166.9
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 1======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  2
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.4f7af2bd221811f0afadd039eab0dadd:discovery
traddr:  192.168.165.9
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 2======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  3
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.4f7af2bd221811f0afadd039eab0dadd:discovery
traddr:  192.168.166.8
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 3======
trtype:  tcp
adrfam:  ipv4
```

```
subtype: current discovery subsystem
treq:    not specified
portid:  1
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.4f7af2bd221811f0afadd039eab0dadd:discovery
traddr:  192.168.165.8
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 4======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.4f7af2bd221811f0afadd039eab0dadd:subsystem.nvme
traddr:  192.168.166.9
eflags:  none
sectype: none
=====Discovery Log Entry 5======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.4f7af2bd221811f0afadd039eab0dadd:subsystem.nvme
traddr:  192.168.165.9
eflags:  none
sectype: none
=====Discovery Log Entry 6======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  3
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.4f7af2bd221811f0afadd039eab0dadd:subsystem.nvme
traddr:  192.168.166.8
eflags:  none
sectype: none
```

```
=====Discovery Log Entry 7======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.4f7af2bd221811f0afadd039eab0dadd:subsystem.nvme
traddr:  192.168.165.8
eflags:  none
sectype: none
```

2. Verify that the other NVMe/TCP initiator-target LIF combinations can successfully fetch discovery log page data:

```
nvme discover -t tcp -w host-traddr -a traddr
```

You should see an output similar to the following example:

```
nvme discover -t tcp -w 192.168.166.4 -a 192.168.166.8
nvme discover -t tcp -w 192.168.165.3 -a 192.168.165.8
nvme discover -t tcp -w 192.168.166.4 -a 192.168.166.9
nvme discover -t tcp -w 192.168.165.3 -a 192.168.165.9
```

3. Run the `nvme connect-all` command across all the supported NVMe/TCP initiator-target LIFs across the nodes:

```
nvme connect-all -t tcp -w host-traddr -a traddr
```

You should see an output similar to the following example:

```
nvme connect-all -t tcp -w 192.168.165.3 -a 192.168.165.8
nvme connect-all -t tcp -w 192.168.165.3 -a 192.168.165.9
nvme connect-all -t tcp -w 192.168.166.4 -a 192.168.166.8
nvme connect-all -t tcp -w 192.168.166.4 -a 192.168.166.9
```

Beginning with Oracle Linux 9.4, the setting for the NVMe/TCP `ctrl_loss_tmo timeout` is automatically set to "off". As a result:

- There are no limits on the number of retries (indefinite retry).
- You don't need to manually configure a specific `ctrl_loss_tmo timeout` duration when using the `nvme connect` or `nvme connect-all` commands (option -l).
- The NVMe/TCP controllers don't experience timeouts in the event of a path failure and remain connected indefinitely.

Step 7: Validate NVMe-oF

Verify that the in-kernel NVMe multipath status, ANA status, and ONTAP namespaces are correct for the NVMe-oF configuration.

Steps

1. Verify that the in-kernel NVMe multipath is enabled:

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

   You should see the following output:

   ```
   Y
   ```

2. Verify that the appropriate NVMe-oF settings (such as, model set to NetApp ONTAP Controller and load balancing iopolicy set to round-robin) for the respective ONTAP namespaces correctly reflect on the host:

   a. Display the subsystems:

      ```
      cat /sys/class/nvme-subsystem/nvme-subsys*/model
      ```

      You should see the following output:

      ```
      NetApp ONTAP Controller
      NetApp ONTAP Controller
      ```

   b. Display the policy:

      ```
      cat /sys/class/nvme-subsystem/nvme-subsys*/iopolicy
      ```

      You should see the following output:

```
round-robin
round-robin
```

3. Verify that the namespaces are created and correctly discovered on the host:

```
nvme list
```

Show example

```
Node          Generic       SN                  Model
Namespace  Usage                  Format         FW Rev
------------- ------------- --------------------
------------------------- ---------- -----------------------
-------------- --------
/dev/nvme1n1  /dev/ng1n1    81Mc4FXd1tocAAAAAAAC NetApp ONTAP
Controller   0x1        0.00   B /  10.74  GB   4 KiB +  0 B
9.16.1
```

4. Verify that the controller state of each path is live and has the correct ANA status:

NVMe/FC

```
nvme list-subsys /dev/nvme4n5
```

Show example

```
nvme-subsys7 - NQN=nqn.1992-
08.com.netapp:sn.7d37987be3cb11ef8948d039eab0dadd:subsystem.nvme
6
               hostnqn=nqn.2014-08.org.nvmexpress:uuid:2831093d-
fa7f-4714-a6bf-548796e82053
               iopolicy=round-robin
\
 +- nvme103 fc traddr=nn-0x202cd039eab0dadc:pn-
0x202fd039eab0dadc,host_traddr=nn-0x200034800d767bb0:pn-
0x210034800d767bb0 live optimized
 +- nvme153 fc traddr=nn-0x202cd039eab0dadc:pn-
0x202ed039eab0dadc,host_traddr=nn-0x200034800d767bb1:pn-
0x210034800d767bb1 live non-optimized
 +- nvme55 fc traddr=nn-0x202cd039eab0dadc:pn-
0x202dd039eab0dadc,host_traddr=nn-0x200034800d767bb0:pn-
0x210034800d767bb0 live non-optimized
 +- nvme7 fc traddr=nn-0x202cd039eab0dadc:pn-
0x2030d039eab0dadc,host_traddr=nn-0x200034800d767bb1:pn-
0x210034800d767bb1 live optimized
```

NVMe/TCP

```
nvme list-subsys /dev/nvme1n1
```

Show example

```
nvme-subsys1 - NQN=nqn.1992-
08.com.netapp:sn.4f7af2bd221811f0afadd039eab0dadd:subsystem.nvme
               hostnqn=nqn.2014-08.org.nvmexpress:uuid:9796c1ec-
0d34-11eb-b6b2-3a68dd3bab57
               iopolicy=round-robin\
+- nvme1 tcp
traddr=192.168.165.8,trsvcid=4420,host_traddr=192.168.165.3,
src_addr=192.168.165.3 live optimized
+- nvme2 tcp
traddr=192.168.165.9,trsvcid=4420,host_traddr=192.168.165.3,
src_addr=192.168.165.3 live non-optimized
+- nvme3 tcp
traddr=192.168.166.8,trsvcid=4420,host_traddr=192.168.166.4,
src_addr=192.168.166.4 live optimized
+- nvme4 tcp
traddr=192.168.166.9,trsvcid=4420,host_traddr=192.168.166.4,
src_addr=192.168.166.4 live non-optimized
```

5. Verify that the NetApp plug-in displays the correct values for each ONTAP namespace device:

Column

```
nvme netapp ontapdevices -o column
```

Show example

```
Device           Vserver                   Namespace Path
NSID UUID                                   Size
---------------- -------------------------
-------------------------------------------------- ----
------------------------------------- ---------
/dev/nvme1n1     vs_tcpinband              /vol/volpdc/ns1
1    80eec226-6987-4eb4-bf86-65bf48c5372d   10.74GB
```

JSON

```
nvme netapp ontapdevices -o json
```

Show example

```
{
  "ONTAPdevices":[
    {
      "Device":"/dev/nvme1n1",
      "Vserver":"vs_tcpinband",
      "Namespace_Path":"/vol/volpdc/ns1",
      "NSID":1,
      "UUID":"80eec226-6987-4eb4-bf86-65bf48c5372d",
      "Size":"10.74GB",
      "LBA_Data_Size":4096,
      "Namespace_Size":2621440
    }
  ]
}
```

Step 8: Set up secure in-band authentication

Beginning with ONTAP 9.12.1, secure in-band authentication is supported over NVMe/TCP between an Oracle Linux 9.5 host and an ONTAP controller.

To set up secure authentication, each host or controller must be associated with a `DH-HMAC-CHAP` key, which is a combination of the NQN of the NVMe host or controller and an authentication secret configured by the administrator. To authenticate its peer, an NVMe host or controller must recognize the key associated with the peer.

You can set up secure in-band authentication using the CLI or a config JSON file. If you need to specify different dhchap keys for different subsystems, you must use a config JSON file.

CLI

Set up secure in-band authentication using the CLI.

Steps

1. Obtain the host NQN:

```
cat /etc/nvme/hostnqn
```

2. Generate the dhchap key for the Linux host.

   The following output describes the `gen-dhchap-key` command paramters:

```
nvme gen-dhchap-key -s optional_secret -l key_length {32|48|64} -m
HMAC_function {0|1|2|3} -n host_nqn
•   -s secret key in hexadecimal characters to be used to initialize
the host key
•   -l length of the resulting key in bytes
•   -m HMAC function to use for key transformation
0 = none, 1- SHA-256, 2 = SHA-384, 3=SHA-512
•   -n host NQN to use for key transformation
```

   In the following example, a random dhchap key with HMAC set to 3 (SHA-512) is generated.

```
# nvme gen-dhchap-key -m 3 -n nqn.2014-
08.org.nvmexpress:uuid:9796c1ec-0d34-11eb-b6b2-3a68dd3bab57
DHHC-
1:03:Y5VkkESgmtTGNdX842qemNpFK6BXYVwwnqErgt3IQKP5Fbjje\/JSBOjG5Ea3NB
LRfuiAuUSDUto6eY\/GwKoRp6AwGkw=:
```

3. On the ONTAP controller, add the host and specify both dhchap keys:

```
vserver nvme subsystem host add -vserver <svm_name> -subsystem
<subsystem> -host-nqn <host_nqn> -dhchap-host-secret
<authentication_host_secret> -dhchap-controller-secret
<authentication_controller_secret> -dhchap-hash-function {sha-
256|sha-512} -dhchap-group {none|2048-bit|3072-bit|4096-bit|6144-
bit|8192-bit}
```

4. A host supports two types of authentication methods, unidirectional and bidirectional. On the host, connect to the ONTAP controller and specify dhchap keys based on the chosen authentication method:

```
nvme connect -t tcp -w <host-traddr> -a <tr-addr> -n <host_nqn> -S
<authentication_host_secret> -C <authentication_controller_secret>
```

5. Validate the `nvme connect authentication` command by verifying the host and controller dhchap keys:

   a. Verify the host dhchap keys:

   ```
   cat /sys/class/nvme-subsystem/<nvme-subsysX>/nvme*/dhchap_secret
   ```

   Show example output for a unidirectional configuration

   ```
   cat /sys/class/nvme-subsystem/nvme-subsys1/nvme*/dhchap_secret
   DHHC-
   1:03:Y5VkkESgmtTGNdX842qemNpFK6BXYVwwnqErgt3IQKP5Fbjje\/JSBOjG
   5Ea3NBLRfuiAuUSDUto6eY\/GwKoRp6AwGkw=:
   DHHC-
   1:03:Y5VkkESgmtTGNdX842qemNpFK6BXYVwwnqErgt3IQKP5Fbjje\/JSBOjG
   5Ea3NBLRfuiAuUSDUto6eY\/GwKoRp6AwGkw=:
   DHHC-
   1:03:Y5VkkESgmtTGNdX842qemNpFK6BXYVwwnqErgt3IQKP5Fbjje\/JSBOjG
   5Ea3NBLRfuiAuUSDUto6eY\/GwKoRp6AwGkw=:
     DHHC-
   1:03:Y5VkkESgmtTGNdX842qemNpFK6BXYVwwnqErgt3IQKP5Fbjje\/JSBOjG
   5Ea3NBLRfuiAuUSDUto6eY\/GwKoRp6AwGkw=:
   ```

   b. Verify the controller dhchap keys:

   ```
   cat /sys/class/nvme-subsystem/<nvme-
   subsysX>/nvme*/dhchap_ctrl_secret
   ```

Show example output for a bidirectional configuration

```
cat /sys/class/nvme-subsystem/nvme-
subsys6/nvme*/dhchap_ctrl_secret
DHHC-
1:03:frpLlTrnOYtcWDxPzq4ccxU1UrH2FjV7hYw5s2XEDB+lo+TjMsOwHR\/N
FtM0nBBidx+gdoyUcC5s6hOOtTLDGcz0Kbs=:
DHHC-
1:03:frpLlTrnOYtcWDxPzq4ccxU1UrH2FjV7hYw5s2XEDB+lo+TjMsOwHR\/N
FtM0nBBidx+gdoyUcC5s6hOOtTLDGcz0Kbs=:
DHHC-
1:03:frpLlTrnOYtcWDxPzq4ccxU1UrH2FjV7hYw5s2XEDB+lo+TjMsOwHR\/N
FtM0nBBidx+gdoyUcC5s6hOOtTLDGcz0Kbs=:
DHHC-
1:03:frpLlTrnOYtcWDxPzq4ccxU1UrH2FjV7hYw5s2XEDB+lo+TjMsOwHR\/N
FtM0nBBidx+gdoyUcC5s6hOOtTLDGcz0Kbs=:
```

JSON file

When multiple NVMe subsystems are available on the ONTAP controller configuration, you can use the `/etc/nvme/config.json` file with the `nvme connect-all` command.

Use the `-o` option to generate the JSON file. See the NVMe connect-all manual pages for more syntax options.

Steps

1. Configure the JSON file:

Show example

```
 cat /etc/nvme/config.json
[
  {
    "hostnqn":"nqn.2014-08.org.nvmexpress:uuid:9796c1ec-0d34-
11eb-b6b2-3a68dd3bab57",
    "hostid":"9796c1ec-0d34-11eb-b6b2-3a68dd3bab57",
    "dhchap_key":"DHHC-
1:03:Y5VkkESgmtTGNdX842qemNpFK6BXYVwwnqErgt3IQKP5Fbjje\/JSBOjG5E
a3NBLRfuiAuUSDUto6eY\/GwKoRp6AwGkw=:",
    "subsystems":[
      {
        "nqn":"nqn.1992-
08.com.netapp:sn.4f7af2bd221811f0afadd039eab0dadd:subsystem.nvme
",
        "ports":[
          {
            "transport":"tcp",
            "traddr":"192.168.165.9",
            "host_traddr":"192.168.165.3",
            "trsvcid":"4420",
            "dhchap_key":"DHHC-
1:03:Y5VkkESgmtTGNdX842qemNpFK6BXYVwwnqErgt3IQKP5Fbjje\/JSBOjG5E
a3NBLRfuiAuUSDUto6eY\/GwKoRp6AwGkw=:",
            "dhchap_ctrl_key":"DHHC-
1:03:frpLlTrnOYtcWDxPzq4ccxU1UrH2FjV7hYw5s2XEDB+lo+TjMsOwHR\/NFt
M0nBBidx+gdoyUcC5s6hOOtTLDGcz0Kbs=:"          },
          {
            "transport":"tcp",
            "traddr":"192.168.166.9",
            "host_traddr":"192.168.166.4",
            "trsvcid":"4420",
                        "dhchap_key":"DHHC-
1:03:Y5VkkESgmtTGNdX842qemNpFK6BXYVwwnqErgt3IQKP5Fbjje\/JSBOjG5E
a3NBLRfuiAuUSDUto6eY\/GwKoRp6AwGkw=:",
            "dhchap_ctrl_key":"DHHC-
1:03:frpLlTrnOYtcWDxPzq4ccxU1UrH2FjV7hYw5s2XEDB+lo+TjMsOwHR\/NFt
M0nBBidx+gdoyUcC5s6hOOtTLDGcz0Kbs=:"
          },
          {
            "transport":"tcp",
            "traddr":"192.168.166.8",
            "host_traddr":"192.168.166.4",
            "trsvcid":"4420",
                        "dhchap_key":"DHHC-
```

```
1:03:Y5VkkESgmtTGNdX842qemNpFK6BXYVwwnqErgt3IQKP5Fbjje\/JSBOjG5E
a3NBLRfuiAuUSDUto6eY\/GwKoRp6AwGkw=:",
            "dhchap_ctrl_key":"DHHC-
1:03:frpLlTrnOYtcWDxPzq4ccxU1UrH2FjV7hYw5s2XEDB+lo+TjMsOwHR\/NFt
M0nBBidx+gdoyUcC5s6hOOtTLDGcz0Kbs=:"
          },
          {
            "transport":"tcp",
            "traddr":"192.168.165.8",
            "host_traddr":"192.168.165.3",
            "trsvcid":"4420",
                        "dhchap_key":"DHHC-
1:03:Y5VkkESgmtTGNdX842qemNpFK6BXYVwwnqErgt3IQKP5Fbjje\/JSBOjG5E
a3NBLRfuiAuUSDUto6eY\/GwKoRp6AwGkw=:",
            "dhchap_ctrl_key":"DHHC-
1:03:frpLlTrnOYtcWDxPzq4ccxU1UrH2FjV7hYw5s2XEDB+lo+TjMsOwHR\/NFt
M0nBBidx+gdoyUcC5s6hOOtTLDGcz0Kbs=:"
          }
        ]
      }
    ]
  }
]
```

In the preceding example, `dhchap_key` corresponds to `dhchap_secret` and `dhchap_ctrl_key` corresponds to `dhchap_ctrl_secret`.

2. Connect to the ONTAP controller using the config JSON file:

```
nvme connect-all -J /etc/nvme/config.json
```

3. Verify that the dhchap secrets have been enabled for the respective controllers for each subsystem:

   a. Verify the host dhchap keys:

   ```
   cat /sys/class/nvme-subsystem/nvme-subsys0/nvme0/dhchap_secret
   ```

   The following example shows a dhchap key:

   ```
   DHHC-
   1:03:Y5VkkESgmtTGNdX842qemNpFK6BXYVwwnqErgt3IQKP5Fbjje\/JSBOjG5Ea
   3NBLRfuiAuUSDUto6eY\/GwKoRp6AwGkw=:
   ```

b. Verify the controller dhchap keys:

```
cat /sys/class/nvme-subsystem/nvme-
subsys0/nvme0/dhchap_ctrl_secret
```

You should see an output similar to the following example:

```
DHHC-
1:03:frpLlTrnOYtcWDxPzq4ccxU1UrH2FjV7hYw5s2XEDB+lo+TjMsOwHR\/NFtM
0nBBidx+gdoyUcC5s6hOOtTLDGcz0Kbs=:
```

Step 9: Review the known issues

There are no known issues.

NVMe-oF Host Configuration for Oracle Linux 9.4 with ONTAP

NetApp SAN host configurations support the NVMe over Fabrics (NVMe-oF) protocol with Asymmetric Namespace Access (ANA). In NVMe-oF environments, ANA is equivalent to asymmetric logical unit access (ALUA) multipathing in iSCSI and FCP environments. ANA is implemented using the in-kernel NVMe multipath feature.

About this task

The following support and features are available with the NVMe-oF host configuration for Oracle Linux 9.4 with ONTAP storage. You should also review the known limitations before starting the configuration process.

- Support available:
  - Support for NVMe over TCP (NVMe/TCP) in addition to NVMe over Fibre Channel (NVMe/FC). The NetApp plug-in in the native nvme-cli package displays ONTAP details for both NVMe/FC and NVMe/TCP namespaces.
  - Running both NVMe and SCSI traffic on the same host. For example, you can configure dm-multipath for SCSI mpath devices on SCSI LUNs and use NVMe multipath to configure NVMe-oF namespace devices on the host.

    For additional details on supported configurations, see the Interoperability Matrix Tool.
- Features available:
  - Beginning with ONTAP 9.12.1, support for secure in-band authentication is introduced for NVMe/TCP. You can use secure in-band authentication for NVMe/TCP with Oracle Linux 9.4
  - Support for in-kernel NVMe multipath enabled for NVMe namespaces by default, therefore, there is no need for explicit settings.
- Known limitations:
  - SAN booting using the NVMe-oF protocol is currently not supported.

Validate software versions

You can use the following procedure to validate the minimum supported Oracle Linux 9.4 software versions.

Steps

1. Install Oracle Linux 9.4 GA on the server. After the installation is complete, verify that you are running the specified Oracle Linux 9.4 GA kernel.

   ```
   uname -r
   ```

   ```
   5.15.0-205.149.5.1.el9uek.x86_64
   ```

2. Install the `nvme-cli` package:

   ```
   rpm -qa|grep nvme-cli
   ```

   ```
   nvme-cli-2.6-5.el9.x86_64
   ```

3. Install the `libnvme` package:

   ```
   rpm -qa|grep libnvme
   ```

   ```
   libnvme-1.6-1.el9.x86_64
   ```

4. On the Oracle Linux 9.4 host, check the `hostnqn` string at `/etc/nvme/hostnqn`:

   ```
   cat /etc/nvme/hostnqn
   ```

   ```
   nqn.2014-08.org.nvmexpress:uuid:9c5d23fe-21c5-472f-9aa4-dc68de0882e9
   ```

5. Verify that the `hostnqn` string matches the `hostnqn` string for the corresponding subsystem on the ONTAP array:

   ```
   vserver nvme subsystem host show -vserver vs_coexistence_149
   ```

Show example

```
Vserver Subsystem Priority  Host NQN
------- --------- --------
------------------------------------------------
vs_coexistence_149
        nvme
                  regular   nqn.2014-
08.org.nvmexpress:uuid:9c5d23fe-21c5-472f-9aa4-dc68de0882e9
        nvme_1
                  regular   nqn.2014-
08.org.nvmexpress:uuid:9c5d23fe-21c5-472f-9aa4-dc68de0882e9
        nvme_2
                  regular   nqn.2014-
08.org.nvmexpress:uuid:9c5d23fe-21c5-472f-9aa4-dc68de0882e9
        nvme_3
                  regular   nqn.2014-
08.org.nvmexpress:uuid:9c5d23fe-21c5-472f-9aa4-dc68de0882e9
4 entries were displayed.
```

If the `hostnqn` strings don't match, you can use the `vserver modify` command to update the `hostnqn` string on your corresponding ONTAP array subsystem to match the `hostnqn` string from `/etc/nvme/hostnqn` on the host.

Configure NVMe/FC

You can configure NVMe/FC with Broadcom/Emulex FC or Marvell/Qlogic FC adapters. For NVMe/FC configured with a Broadcom adapter, you can enable I/O requests of size 1MB.

Broadcom/Emulex

Configure NVMe/FC for a Broadcom/Emulex adapter.

Steps

1. Verify that you are using the supported adapter model:

   a. `cat /sys/class/scsi_host/host*/modelname`

   ```
   LPe32002-M2
   LPe32002-M2
   ```

   b. `cat /sys/class/scsi_host/host*/modeldesc`

   ```
   Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
   Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
   ```

2. Verify that you are using the recommended Broadcom `lpfc` firmware and inbox driver:

   a. `cat /sys/class/scsi_host/host*/fwrev`

   ```
   14.4.317.7, sli-4:2:c
   14.4.317.7, sli-4:2:c
   ```

   b. `cat /sys/module/lpfc/version`

   ```
   0:14.2.0.13
   ```

   For the current list of supported adapter driver and firmware versions, see the Interoperability Matrix Tool.

3. Verify that `lpfc_enable_fc4_type` is set to `3`:

   `cat /sys/module/lpfc/parameters/lpfc_enable_fc4_type`

   ```
   3
   ```

4. Verify that you can view your initiator ports:

   `cat /sys/class/fc_host/host*/port_name`

   ```
   0x100000109b3c081f
   0x100000109b3c0820
   ```

5. Verify that your initiator ports are online:

   ```
   cat /sys/class/fc_host/host*/port_state
   ```

   ```
   Online
   Online
   ```

6. Verify that the NVMe/FC initiator ports are enabled and that the target ports are visible:

   ```
   cat /sys/class/scsi_host/host*/nvme_info
   ```

Show example

```
NVME Initiator Enabled
XRI Dist lpfc0 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc0 WWPN x100000109b3c081f WWNN x200000109b3c081f
DID x081600 ONLINE
NVME RPORT       WWPN x2020d039eab0dadc WWNN x201fd039eab0dadc
DID x08010c TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2024d039eab0dadc WWNN x201fd039eab0dadc
DID x08030c TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 00000027d8 Cmpl 00000027d8 Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 00000000315454fa Issue 00000000314de6a4 OutIO
fffffffffff991aa
        abort 00000be4 noxri 00000000 nondlp 00001903 qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 00000c92 Err 0000bda4

NVME Initiator Enabled
XRI Dist lpfc1 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc1 WWPN x100000109b3c0820 WWNN x200000109b3c0820
DID x081b00 ONLINE
NVME RPORT       WWPN x2027d039eab0dadc WWNN x201fd039eab0dadc
DID x08020c TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2025d039eab0dadc WWNN x201fd039eab0dadc
DID x08040c TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 00000026ac Cmpl 00000026ac Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 00000000312a5478 Issue 00000000312465a2 OutIO
fffffffffffa112a
        abort 00000b01 noxri 00000000 nondlp 00001ae4 qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 00000b53 Err 0000ba63
```

Marvell/QLogic

Configure NVMe/FC for a Marvell/QLogic adapter.

ⓘ The native inbox qla2xxx driver included in the Oracle Linux 9.4 GA kernel has the latest fixes. These fixes are essential for ONTAP support.

Steps

1. Verify that you are running the supported adapter driver and firmware versions:

```
cat /sys/class/fc_host/host*/symbolic_name
```

```
QLE2872 FW:v9.15.00 DVR:v10.02.09.100-k
QLE2872 FW:v9.15.00 DVR:v10.02.09.100-k
```

2. Verify that `ql2xnvmeenable` is set. This enables the Marvell adapter to function as an NVMe/FC initiator:

```
cat /sys/module/qla2xxx/parameters/ql2xnvmeenable
```

```
1
```

Enable 1MB I/O size (Optional)

ONTAP reports an MDTS (Max Data Transfer Size) of 8 in the Identify Controller data. This means the maximum I/O request size can be up to 1MB. To issue I/O requests of size 1MB for a Broadcom NVMe/FC host, you should increase the `lpfc` value of the `lpfc_sg_seg_cnt` parameter to 256 from the default value of 64.

> These steps don't apply to Qlogic NVMe/FC hosts.

Steps

1. Set the `lpfc_sg_seg_cnt` parameter to 256:

```
cat /etc/modprobe.d/lpfc.conf
```

You should see an output similar to the following example:

```
options lpfc lpfc_sg_seg_cnt=256
```

2. Run the `dracut -f` command, and reboot the host.
3. Verify that the value for `lpfc_sg_seg_cnt` is 256:

```
cat /sys/module/lpfc/parameters/lpfc_sg_seg_cnt
```

Configure NVMe/TCP

The NVMe/TCP protocol doesn't support the `auto-connect` operation. Instead, you can discover the NVMe/TCP subsystems and namespaces by performing the NVMe/TCP `connect` or `connect-all` operations manually.

Steps

1. Verify that the initiator port can fetch the discovery log page data across the supported NVMe/TCP LIFs:

    ```
    nvme discover -t tcp -w host-traddr -a traddr
    ```

Show example

```
nvme discover -t tcp -w 192.168.166.4 -a 192.168.166.56

Discovery Log Number of Records 10, Generation counter 15
=====Discovery Log Entry 0======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  13
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.cf84a53c81b111ef8446d039ea9ea481:discovery
traddr:  192.168.165.56
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 1======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  9
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.cf84a53c81b111ef8446d039ea9ea481:discovery
traddr:  192.168.166.56
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 2======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  13
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.cf84a53c81b111ef8446d039ea9ea481:subsystem.nvme_tcp
_2
traddr:  192.168.165.56
eflags:  none
sectype: none
```

2. Verify that the other NVMe/TCP initiator-target LIF combinations can successfully fetch discovery log page data:

```
nvme discover -t tcp -w host-traddr -a traddr
```

```
nvme discover -t tcp -w 192.168.166.4 -a 192.168.166.56
nvme discover -t tcp -w 192.168.165.3 -a 192.168.165.56
```

3. Run the `nvme connect-all` command across all the supported NVMe/TCP initiator-target LIFs across the nodes:

```
nvme connect-all -t tcp -w host-traddr -a traddr
```

```
nvme connect-all -t tcp -w 192.168.166.4 -a 192.168.166.56
nvme connect-all -t tcp -w 192.168.165.3 -a 192.168.165.56
```

> Beginning with Oracle Linux 9.4, the setting for the NVMe/TCP `ctrl_loss_tmo timeout` is automatically set to "off". As a result:
>
> - There are no limits on the number of retries (indefinite retry).
> - You don't need to manually configure a specific `ctrl_loss_tmo timeout` duration when using the `nvme connect` or `nvme connect-all` commands (option -l).
> - The NVMe/TCP controllers don't experience timeouts in the event of a path failure and remain connected indefinitely.

Validate NVMe-oF

Verify that the in-kernel NVMe multipath status, ANA status, and ONTAP namespaces are correct for the NVMe-oF configuration.

Steps

1. Verify that the in-kernel NVMe multipath is enabled:

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

You should see the following output:

```
Y
```

2. Verify that the appropriate NVMe-oF settings (such as, model set to NetApp ONTAP Controller and load balancing iopolicy set to round-robin) for the respective ONTAP namespaces correctly reflect on the host:

a. Display the subsystems:

```
cat /sys/class/nvme-subsystem/nvme-subsys*/model
```

You should see the following output:

```
NetApp ONTAP Controller
NetApp ONTAP Controller
```

b. Display the policy:

```
cat /sys/class/nvme-subsystem/nvme-subsys*/iopolicy
```

You should see the following output:

```
round-robin
round-robin
```

3. Verify that the namespaces are created and correctly discovered on the host:

```
nvme list
```

Show example

```
Node         SN                   Model
---------------------------------------------------------
/dev/nvme4n1 81Ix2BVuekWcAAAAAAAB   NetApp ONTAP Controller


Namespace Usage    Format             FW             Rev
-----------------------------------------------------------
1                 21.47 GB / 21.47 GB   4 KiB + 0 B   FFFFFFFF
```

Steps

1. Verify the following NVMe/FC settings on the Oracle Liniux 9.4 host:

a. `cat /sys/module/nvme_core/parameters/multipath`

```
Y
```

b. `cat /sys/class/nvme-subsystem/nvme-subsys*/model`

```
NetApp ONTAP Controller
NetApp ONTAP Controller
```

c. `cat /sys/class/nvme-subsystem/nvme-subsys*/iopolicy`

```
round-robin
round-robin
```

2. Verify that the namespaces are created and correctly discovered on the host:

```
nvme list
```

Show example

```
Node         SN                   Model
---------------------------------------------------------
/dev/nvme0n1 81K2iBXAYSG6AAAAAAAB NetApp ONTAP Controller
/dev/nvme0n2 81K2iBXAYSG6AAAAAAAB NetApp ONTAP Controller
/dev/nvme0n3 81K2iBXAYSG6AAAAAAAB NetApp ONTAP Controller


Namespace Usage    Format             FW             Rev
-----------------------------------------------------------
1                 3.78GB/10.74GB 4 KiB + 0 B       FFFFFFFF
2                 3.78GB/10.74GB 4 KiB +  0 B      FFFFFFFF
3                 3.78GB/10.74GB 4 KiB + 0 B       FFFFFFFF
```

3. Verify that the controller state of each path is live and has the correct ANA status:

NVMe/FC

```
nvme list-subsys /dev/nvme0n1
```

Show example

```
nvme-subsys0 - NQN=nqn.1992-
08.com.netapp:sn.5f074d527b7011ef8446d039ea9ea481:subsystem.nvme
               hostnqn=nqn.2014-08.org.nvmexpress:uuid:060fd513-
83be-4c3e-aba1-52e169056dcf
               iopolicy=round-robin
\
 +- nvme10 fc traddr=nn-0x201fd039eab0dadc:pn-
0x2024d039eab0dadc,host_traddr=nn-0x200000109b3c081f:pn-
0x100000109b3c081f live non-optimized
 +- nvme15 fc traddr=nn-0x201fd039eab0dadc:pn-
0x2020d039eab0dadc,host_traddr=nn-0x200000109b3c081f:pn-
0x100000109b3c081f live optimized
 +- nvme7 fc traddr=nn-0x201fd039eab0dadc:pn-
0x2025d039eab0dadc,host_traddr=nn-0x200000109b3c0820:pn-
0x100000109b3c0820 live non-optimized
 +- nvme9 fc traddr=nn-0x201fd039eab0dadc:pn-
0x2027d039eab0dadc,host_traddr=nn-0x200000109b3c0820:pn-
0x100000109b3c0820 live optimized
```

NVMe/TCP

```
nvme list-subsys /dev/nvme1n22
```

Show example

```
nvme-subsys0 - NQN=nqn.1992-
08.com.netapp:sn.cf84a53c81b111ef8446d039ea9ea481:subsystem.nvme
_tcp_1
               hostnqn=nqn.2014-08.org.nvmexpress:uuid:9796c1ec-
0d34-11eb-b6b2-3a68dd3bab57
               iopolicy=round-robin
\
 +- nvme2 tcp
traddr=192.168.166.56,trsvcid=4420,host_traddr=192.168.166.4,src
_addr=192.168.166.4 live optimized
 +- nvme4 tcp
traddr=192.168.165.56,trsvcid=4420,host_traddr=192.168.165.3,src
_addr=192.168.165.3 live non-optimized
```

4. Verify that the NetApp plug-in displays the correct values for each ONTAP namespace device:

Column

```
nvme netapp ontapdevices -o column
```

Show example

```
Device        Vserver   Namespace Path
----------------------- ------------------------------
/dev/nvme0n1     vs_coexistence_147 /vol/fcnvme_1_1_0/fcnvme_ns
/dev/nvme0n2     vs_coexistence_147 /vol/fcnvme_1_1_1/fcnvme_ns
/dev/nvme0n3     vs_coexistence_147 /vol/fcnvme_1_1_2/fcnvme_ns




NSID       UUID                                   Size
------------------------------------------------------------
1   e605babf-1b54-417d-843b-bc14355b70c5    10.74GB
2   b8dbecc7-14c5-4d84-b948-73c7abf5af43    10.74GB
3   ba24d1a3-1911-4351-83a9-1c843d04633c    10.74GB
```

JSON

```
nvme netapp ontapdevices -o json
```

Show example


```
{
  "ONTAPdevices":[
    {
      "Device":"/dev/nvme0n1",
      "Vserver":"vs_coexistence_147",
      "Namespace_Path":"/vol/fcnvme_1_1_0/fcnvme_ns",
      "NSID":1,
      "UUID":"e605babf-1b54-417d-843b-bc14355b70c5",
      "Size":"10.74GB",
      "LBA_Data_Size":4096,
      "Namespace_Size":2621440
    },
    {
      "Device":"/dev/nvme0n2",
      "Vserver":"vs_coexistence_147",
      "Namespace_Path":"/vol/fcnvme_1_1_1/fcnvme_ns",
      "NSID":2,
      "UUID":"b8dbecc7-14c5-4d84-b948-73c7abf5af43",
      "Size":"10.74GB",
      "LBA_Data_Size":4096,
      "Namespace_Size":2621440
    },
    {
      "Device":"/dev/nvme0n3",
      "Vserver":"vs_coexistence_147",
      "Namespace_Path":"/vol/fcnvme_1_1_2/fcnvme_ns",
      "NSID":3,
      "UUID":"c236905d-a335-47c4-a4b1-89ae30de45ae",
      "Size":"10.74GB",
      "LBA_Data_Size":4096,
      "Namespace_Size":2621440
    },
    ]
}
```


Set up secure in-band authentication

Beginning with ONTAP 9.12.1, secure in-band authentication is supported over NVMe/TCP between an Oracle Linux 9.4 host and an ONTAP controller.

To set up secure authentication, each host or controller must be associated with a `DH-HMAC-CHAP` key, which is a combination of the NQN of the NVMe host or controller and an authentication secret configured by the administrator. To authenticate its peer, an NVMe host or controller must recognize the key associated with the peer.

You can set up secure in-band authentication using the CLI or a config JSON file. If you need to specify different dhchap keys for different subsystems, you must use a config JSON file.

CLI

Set up secure in-band authentication using the CLI.

Steps

1. Obtain the host NQN:

```
cat /etc/nvme/hostnqn
```

2. Generate the dhchap key for the OL 9.4 host.

   The following output describes the `gen-dhchap-key` command paramters:

```
nvme gen-dhchap-key -s optional_secret -l key_length {32|48|64} -m
HMAC_function {0|1|2|3} -n host_nqn
•   -s secret key in hexadecimal characters to be used to initialize
the host key
•   -l length of the resulting key in bytes
•   -m HMAC function to use for key transformation
0 = none, 1- SHA-256, 2 = SHA-384, 3=SHA-512
•   -n host NQN to use for key transformation
```

   In the following example, a random dhchap key with HMAC set to 3 (SHA-512) is generated.

```
# nvme gen-dhchap-key -m 3 -n nqn.2014-
08.org.nvmexpress:uuid:9796c1ec-0d34-11eb-b6b2-3a68dd3bab57
DHHC-
1:03:zSq3+upTmknih8+6Ro0yw6KBQNAXjHFrOxQJaE5i916YdM/xsUSTdLkHw2MMmdF
uGEslj6+LhNdf5HF0qfroFPgoQpU=:
```

3. On the ONTAP controller, add the host and specify both dhchap keys:

```
vserver nvme subsystem host add -vserver <svm_name> -subsystem
<subsystem> -host-nqn <host_nqn> -dhchap-host-secret
<authentication_host_secret> -dhchap-controller-secret
<authentication_controller_secret> -dhchap-hash-function {sha-
256|sha-512} -dhchap-group {none|2048-bit|3072-bit|4096-bit|6144-
bit|8192-bit}
```

4. A host supports two types of authentication methods, unidirectional and bidirectional. On the host, connect to the ONTAP controller and specify dhchap keys based on the chosen authentication method:

```
nvme connect -t tcp -w <host-traddr> -a <tr-addr> -n <host_nqn> -S
<authentication_host_secret> -C <authentication_controller_secret>
```

5. Validate the `nvme connect authentication` command by verifying the host and controller dhchap keys:
    a. Verify the host dhchap keys:

    ```
    cat /sys/class/nvme-subsystem/<nvme-subsysX>/nvme*/dhchap_secret
    ```

    Show example output for a unidirectional configuration

    ```
    cat /sys/class/nvme-subsystem/nvme-subsys0/nvme*/dhchap_secret
    DHHC-1:01:OKIc4l+fs+fmpAj0hMK7ay8tTIzjccUWSCak/G2XjgJpKZeK:
    DHHC-1:01:OKIc4l+fs+fmpAj0hMK7ay8tTIzjccUWSCak/G2XjgJpKZeK:
    ```

    b. Verify the controller dhchap keys:

    ```
    cat /sys/class/nvme-subsystem/<nvme-
    subsysX>/nvme*/dhchap_ctrl_secret
    ```

    Show example output for a bidirectional configuration

    ```
    cat /sys/class/nvme-subsystem/nvme-
    subsys0/nvme*/dhchap_ctrl_secret
    DHHC-
    1:03:zSq3+upTmknih8+6Ro0yw6KBQNAXjHFrOxQJaE5i916YdM/xsUSTdLkHw
    2MMmdFuGEslj6+LhNdf5HF0qfroFPgoQpU=:
    DHHC-
    1:03:zSq3+upTmknih8+6Ro0yw6KBQNAXjHFrOxQJaE5i916YdM/xsUSTdLkHw
    2MMmdFuGEslj6+LhNdf5HF0qfroFPgoQpU=:
    ```

JSON file

When multiple NVMe subsystems are available on the ONTAP controller configuration, you can use the `/etc/nvme/config.json` file with the `nvme connect-all` command.

To generate the JSON file, you can use the `-o` option. See the NVMe connect-all manual pages for more syntax options.

Steps

1. Configure the JSON file:

Show example

```
cat /etc/nvme/config.json
[
  {
    "hostnqn":"nqn.2014-08.org.nvmexpress:uuid:9796c1ec-0d34-
11eb-b6b2-3a68dd3bab57",
    "hostid":"9796c1ec-0d34-11eb-b6b2-3a68dd3bab57",
    "dhchap_key":"DHHC-
1:01:OKIc4l+fs+fmpAj0hMK7ay8tTIzjccUWSCak\/G2XjgJpKZeK:",
    "subsystems":[
      {
        "nqn":"nqn.1992-
08.com.netapp:sn.cf84a53c81b111ef8446d039ea9ea481:subsystem.nvme
_tcp_1",
        "ports":[
          {
            "transport":"tcp",
            "traddr":"192.168.165.56",
            "host_traddr":"192.168.165.3",
            "trsvcid":"4420",
            "dhchap_key":"DHHC-
1:01:OKIc4l+fs+fmpAj0hMK7ay8tTIzjccUWSCak\/G2XjgJpKZeK:",
            "dhchap_ctrl_key":"DHHC-
1:03:zSq3+upTmknih8+6Ro0yw6KBQNAXjHFrOxQJaE5i916YdM\/xsUSTdLkHw2
MMmdFuGEslj6+LhNdf5HF0qfroFPgoQpU=:"
          },
          {
            "transport":"tcp",
            "traddr":"192.168.166.56",
            "host_traddr":"192.168.166.4",
            "trsvcid":"4420",
            "dhchap_key":"DHHC-
1:01:OKIc4l+fs+fmpAj0hMK7ay8tTIzjccUWSCak\/G2XjgJpKZeK:",
            "dhchap_ctrl_key":"DHHC-
1:03:zSq3+upTmknih8+6Ro0yw6KBQNAXjHFrOxQJaE5i916YdM\/xsUSTdLkHw2
MMmdFuGEslj6+LhNdf5HF0qfroFPgoQpU=:"
          }
        ]
      }
    ]
  }
]
```

In the preceding example, `dhchap_key` corresponds to `dhchap_secret` and `dhchap_ctrl_key` corresponds to `dhchap_ctrl_secret`.

2. Connect to the ONTAP controller using the config JSON file:

```
nvme connect-all -J /etc/nvme/config.json
```

Show example

```
traddr=192.168.165.56 is already connected
traddr=192.168.165.56 is already connected
traddr=192.168.165.56 is already connected
traddr=192.168.165.56 is already connected
traddr=192.168.165.56 is already connected
traddr=192.168.165.56 is already connected
traddr=192.168.166.56 is already connected
traddr=192.168.166.56 is already connected
traddr=192.168.166.56 is already connected
traddr=192.168.166.56 is already connected
traddr=192.168.166.56 is already connected
traddr=192.168.166.56 is already connected
```

3. Verify that the dhchap secrets have been enabled for the respective controllers for each subsystem:
   a. Verify the host dhchap keys:

```
cat /sys/class/nvme-subsystem/nvme-subsys0/nvme0/dhchap_secret
```

```
DHHC-1:01:OKIc4l+fs+fmpAj0hMK7ay8tTIzjccUWSCak/G2XjgJpKZeK:
```

   b. Verify the controller dhchap keys:

```
cat /sys/class/nvme-subsystem/nvme-
subsys0/nvme0/dhchap_ctrl_secret
```

```
DHHC-
1:03:zSq3+upTmknih8+6Ro0yw6KBQNAXjHFrOxQJaE5i916YdM/xsUSTdLkHw2MM
mdFuGEslj6+LhNdf5HF0qfroFPgoQpU=:
```

Known issues

There are no known issues for the Oracle Linux 9.4 with ONTAP release.

NVMe-oF Host Configuration for Oracle Linux 9.3 with ONTAP

NVMe over Fabrics (NVMe-oF), including NVMe over Fibre Channel (NVMe/FC) and other transports, is supported with Oracle Linux (OL) 9.3 with Asymmetric Namespace Access (ANA). In NVMe-oF environments, ANA is the equivalent of ALUA multipathing in iSCSI and FC environments and is implemented with in-kernel NVMe multipath.

The following support is available for the NVMe-oF host configuration for OL 9.3 with ONTAP:

- Support for NVMe over TCP (NVMe/TCP) in addition to NVMe/FC. The NetApp plug-in in the native nvme-cli package displays ONTAP details for both NVMe/FC and NVMe/TCP namespaces.
- Use of NVMe and SCSI co-existent traffic on the same host on a given host bus adapter (HBA), without the explicit dm-multipath settings to prevent claiming NVMe namespaces.

For additional details on supported configurations, see the Interoperability Matrix Tool.

Features

Oracle Linux 9.3 has in-kernel NVMe multipath enabled for NVMe namespaces by default, therefore, there is no need for explicit settings.

Known limitations

SAN booting using the NVMe-oF protocol is currently not supported.

Validate software versions

You can use the following procedure to validate the minimum supported OL 9.3 software versions.

Steps

1. Install OL 9.3 GA on the server. After the installation is complete, verify that you are running the specified OL 9.3 GA kernel.

   ```
   # uname -r
   ```

   Example output:

   ```
   5.15.0-200.131.27.el9uek.x86_64
   ```

2. Install the `nvme-cli` package:

   ```
   # rpm -qa|grep nvme-cli
   ```

   Example output:

```
nvme-cli-2.4-10.el9.x86_64
```

3. Install the `libnvme` package:

```
#rpm -qa|grep libnvme
```

Example output

```
libnvme-1.4-7.el9.x86_64
```

4. On the Oracle Linux 9.3 host, check the `hostnqn` string at `/etc/nvme/hostnqn`:

```
# cat /etc/nvme/hostnqn
```

Example output:

```
nqn.2014-08.org.nvmexpress:uuid:2831093d-fa7f-4714-a6bf-548796e82053
```

5. Verify that the `hostnqn` string matches the `hostnqn` string for the corresponding subsystem on the ONTAP array:

```
::> vserver nvme subsystem host show -vserver vs_ol_nvme
```

Example output:

```
Vserver     Subsystem          Host NQN
----------- ---------------
----------------------------------------------------------
vs_ol_nvme   nvme              nqn.2014-08.org.nvmexpress:uuid:2831093d-
fa7f-4714-a6bf-548796e82053
```

> If the `hostnqn` strings do not match, you can use the `vserver modify` command to update the `hostnqn` string on your corresponding ONTAP array subsystem to match the `hostnqn` string from `/etc/nvme/hostnqn` on the host.

Configure NVMe/FC

You can configure NVMe/FC for Broadcom/Emulex adapters or Marvell/Qlogic adapters.

Broadcom/Emulex

Steps

1. Verify that you are using the supported adapter model:

   ```
   # cat /sys/class/scsi_host/host*/modelname
   ```

   Example output:

   ```
   LPe36002-M2
   LPe36002-M2
   ```

   ```
   # cat /sys/class/scsi_host/host*/modeldesc
   ```

   Example output:

   ```
   Emulex LightPulse LPe36002-M2 2-Port 64Gb Fibre Channel Adapter
   Emulex LightPulse LPe36002-M2 2-Port 64Gb Fibre Channel Adapter
   ```

2. Verify that you are using the recommended Broadcom `lpfc` firmware and inbox driver:

   ```
   # cat /sys/class/scsi_host/host*/fwrev
   14.2.673.40, sli-4:2:c
   14.2.673.40, sli-4:2:c
   ```

   ```
   # cat /sys/module/lpfc/version
   0:14.2.0.13
   ```

   For the current list of supported adapter driver and firmware versions, see the Interoperability Matrix Tool.

3. Verify that `lpfc_enable_fc4_type` is set to `3`:

   ```
   # cat /sys/module/lpfc/parameters/lpfc_enable_fc4_type
   3
   ```

4. Verify that the initiator ports are up and running, and that you can see the target LIFs:

```
# cat /sys/class/fc_host/host*/port_name
0x100000620b3c089c
0x100000620b3c089d
```

```
# cat /sys/class/fc_host/host*/port_state
Online
Online
```

Show example output

```
# cat /sys/class/scsi_host/host*/nvme_info
NVME Initiator Enabled
XRI Dist lpfc0 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc0 WWPN x100000620b3c089c WWNN x200000620b3c089c
DID x062f00 ONLINE
NVME RPORT       WWPN x2019d039ea9ea480 WWNN x2018d039ea9ea480
DID x061b06 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x201cd039ea9ea480 WWNN x2018d039ea9ea480
DID x062706 TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 0000000f03 Cmpl 0000000efa Abort 0000004a
LS XMIT: Err 00000009  CMPL: xb 0000004a Err 0000004a
Total FCP Cmpl 00000000b9b3486a Issue 00000000b97ba0d2 OutIO
fffffffffffc85868
abort 00000afc noxri 00000000 nondlp 00002e34 qdepth 00000000
wqerr 00000000 err 00000000
FCP CMPL: xb 0000138c Err 00014750

NVME Initiator Enabled
XRI Dist lpfc1 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc1 WWPN x100000620b3c089d WWNN x200000620b3c089d
DID x062400 ONLINE
NVME RPORT       WWPN x201ad039ea9ea480 WWNN x2018d039ea9ea480
DID x060206 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x201dd039ea9ea480 WWNN x2018d039ea9ea480
DID x061305 TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 0000000b40 Cmpl 0000000b40 Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 00000000b9a9f03f Issue 00000000b96e622e OutIO
fffffffffffc471ef
abort 0000090d noxri 00000000 nondlp 00003b3f qdepth 00000000
wqerr 00000000 err 00000000
FCP CMPL: xb 000010a5 Err 000147e4
```

Marvell/QLogic FC Adapter for NVMe/FC

The native inbox qla2xxx driver included in the OL 9.3 GA kernel has the latest upstream fixes. These fixes are essential for ONTAP support.

Steps

1. Verify that you are running the supported adapter driver and firmware versions:

```
# cat /sys/class/fc_host/host*/symbolic_name
QLE2872 FW:v9.14.02 DVR:v 10.02.09.100-k
QLE2872 FW:v9.14.02 DVR:v 10.02.09.100-k
```

2. Verify that `ql2xnvmeenable` is set. This enables the Marvell adapter to function as an NVMe/FC initiator:

```
# cat /sys/module/qla2xxx/parameters/ql2xnvmeenable
1
```

Enable 1MB I/O size (Optional)

ONTAP reports an MDTS (Max Data Transfer Size) of 8 in the Identify Controller data. This means the maximum I/O request size can be up to 1MB. To issue I/O requests of size 1MB for a Broadcom NVMe/FC host, you should increase the `lpfc` value of the `lpfc_sg_seg_cnt` parameter to 256 from the default value of 64.

> These steps don't apply to Qlogic NVMe/FC hosts.

Steps

1. Set the `lpfc_sg_seg_cnt` parameter to 256:

```
cat /etc/modprobe.d/lpfc.conf
```

You should see an output similar to the following example:

```
options lpfc lpfc_sg_seg_cnt=256
```

2. Run the `dracut -f` command, and reboot the host.
3. Verify that the value for `lpfc_sg_seg_cnt` is 256:

```
cat /sys/module/lpfc/parameters/lpfc_sg_seg_cnt
```

Configure NVMe/TCP

NVMe/TCP does not have an auto-connect functionality. Therefore, you need to perform the NVMe/TCP connect or connect-all functionality manually to discover the NVMe/TCP subsystems and namespaces. You can use the following procedure to configure NVMe/TCP.

Steps

1. Verify that the initiator port can fetch the discovery log page data across the supported NVMe/TCP LIFs:

```
nvme discover -t tcp -w host-traddr -a traddr
```

Show example

```
# nvme discover -t tcp -w 192.168.166.4 -a 192.168.166.56

Discovery Log Number of Records 4, Generation counter 10
=====Discovery Log Entry 0======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  2
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.337a0392d58011ee9764d039eab0dadd:discovery
traddr:  192.168.165.56
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 1======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  1
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.337a0392d58011ee9764d039eab0dadd:discovery
traddr:  192.168.166.56
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 2======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.337a0392d58011ee9764d039eab0dadd:subsystem.rhel_95
traddr:  192.168.165.56
eflags:  none
sectype: none
..........
```

2. Verify that the other NVMe/TCP initiator-target LIF combinations can successfully fetch discovery log page data:

```
nvme discover -t tcp -w host-traddr -a traddr
```

Example output:

```
# nvme discover -t tcp -w 192.168.166.4 -a 192.168.166.56
# nvme discover -t tcp -w 192.168.165.3 -a 192.168.165.56
```

3. Run the `nvme connect-all` command across all the supported NVMe/TCP initiator-target LIFs across the nodes:

```
nvme connect-all -t tcp -w host-traddr -a traddr -l
<ctrl_loss_timeout_in_seconds>
```

Example output:

```
# nvme connect-all -t tcp -w 192.168.166.4 -a 192.168.166.56 -l -1
# nvme connect-all -t tcp -w 192.168.165.3 -a 192.168.165.56 -l -1
```

> NetApp recommends setting the `ctrl-loss-tmo` option to `-1` so that the NVMe/TCP initiator attempts to reconnect indefinitely in the event of a path loss.

Validate NVMe-oF

You can use the following procedure to validate NVMe-oF.

Steps

1. Verify the following NVMe/FC settings on the OL 9.3 host:

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

```
# cat /sys/class/nvme-subsystem/nvme-subsys*/model
NetApp ONTAP Controller
NetApp ONTAP Controller
```

```
# cat /sys/class/nvme-subsystem/nvme-subsys*/iopolicy
round-robin
round-robin
```

2. Verify that the namespaces are created and correctly discovered on the host:

```
# nvme list
```

Example output:

```
Node         SN                   Model
---------------------------------------------------------
/dev/nvme0n1 814vWBNRwf9HAAAAAAAB NetApp ONTAP Controller
/dev/nvme0n2 814vWBNRwf9HAAAAAAAB NetApp ONTAP Controller
/dev/nvme0n3 814vWBNRwf9HAAAAAAAB NetApp ONTAP Controller


Namespace Usage    Format             FW             Rev
-----------------------------------------------------------
1                 21.47 GB / 21.47 GB  4 KiB + 0 B   FFFFFFFF
2                 21.47 GB / 21.47 GB  4 KiB + 0 B  FFFFFFFF
3                   21.47 GB/ 21.47 GB 4 KiB + 0 B   FFFFFFFF
```

3. Verify that the controller state of each path is live and has the correct ANA status:

NVMe/FC

```
# nvme list-subsys /dev/nvme0n1
```

Example output:

```
nvme-subsys5 - NQN=nqn.1992-
08.com.netapp:sn.4aa0fa76c92c11eeb301d039eab0dadd:subsystem.rhel_213
\
 +- nvme3 fc traddr=nn-0x2018d039ea9ea480:pn-
0x201dd039ea9ea480,host_traddr=nn-0x200000620b3c089d:pn-
0x100000620b3c089d live non-optimized
 +- nvme4 fc traddr=nn-0x2018d039ea9ea480:pn-
0x201cd039ea9ea480,host_traddr=nn-0x200000620b3c089c:pn-
0x100000620b3c089c live non-optimized
 +- nvme6 fc traddr=nn-0x2018d039ea9ea480:pn-
0x2019d039ea9ea480,host_traddr=nn-0x200000620b3c089c:pn-
0x100000620b3c089c live optimized
 +- nvme7 fc traddr=nn-0x2018d039ea9ea480:pn-
0x201ad039ea9ea480,host_traddr=nn-0x200000620b3c089d:pn-
0x100000620b3c089d live optimized
```

NVMe/TCP

```
nvme list-subsys /dev/nvme1n22
```

Example output

```
nvme-subsys1 - NQN=nqn.1992-
08.com.netapp:sn.337a0392d58011ee9764d039eab0dadd:subsystem.rhel_95
\
 +- nvme2 tcp
traddr=192.168.166.56,trsvcid=4420,host_traddr=192.168.166.4,src_add
r=192.168.166.4 live optimized
 +- nvme3 tcp
traddr=192.168.165.56,trsvcid=4420,host_traddr=192.168.165.3,src_add
r=192.168.165.3 live non-optimized
```

4. Verify that the NetApp plug-in displays the correct values for each ONTAP namespace device:

Column

```
# nvme netapp ontapdevices -o column
```

Example output:

```
Device        Vserver   Namespace Path
----------------------- ------------------------------
/dev/nvme5n6     vs_nvme175                /vol/vol6/ns
/dev/nvme5n7     vs_nvme175                /vol/vol7/ns
/dev/nvme5n8     vs_nvme175                /vol/vol8/ns




NSID       UUID                                   Size
------------------------------------------------------------
6          72b887b1-5fb6-47b8-be0b-33326e2542e2   21.47GB
7          04bf9f6e-9031-40ea-99c7-a1a61b2d7d08   21.47GB
8          264823b1-8e03-4155-80dd-e904237014a4   21.47GB
```

JSON

```
# nvme netapp ontapdevices -o json
```

Example output

```
{
  "ONTAPdevices":[
    {
      "Device":"/dev/nvme5n1",
      "Vserver":"vs_nvme175",
      "Namespace_Path":"/vol/vol1/ns",
      "NSID":1,
      "UUID":"d4791955-07c9-44fc-b41c-d1c39d3d9b5b",
      "Size":"21.47GB",
      "LBA_Data_Size":4096,
      "Namespace_Size":5242880
    },
    {
      "Device":"/dev/nvme5n10",
      "Vserver":"vs_nvme175",
      "Namespace_Path":"/vol/vol10/ns",
      "NSID":10,
      "UUID":"f3a4ce94-bcc5-4ff0-9e52-e59030bbc97f",
      "Size":"21.47GB",
      "LBA_Data_Size":4096,
      "Namespace_Size":5242880
    },
    {
      "Device":"/dev/nvme5n11",
      "Vserver":"vs_nvme175",
      "Namespace_Path":"/vol/vol11/ns",
      "NSID":11,
      "UUID":"0bf171d2-51f7-4a00-8f6a-0ea2190885a2",
      "Size":"21.47GB",
      "LBA_Data_Size":4096,
      "Namespace_Size":5242880
    },
  ]
}
```

Known issues

There are no known issues for the Oracle Linux 9.3 with ONTAP release.

NVMe-oF Host Configuration for Oracle Linux 9.2 with ONTAP

NVMe over Fabrics (NVMe-oF), including NVMe over Fibre Channel (NVMe/FC) and other transports, is supported with Oracle Linux (OL) 9.2 with Asymmetric Namespace Access (ANA). In NVMe-oF environments, ANA is the equivalent of ALUA multipathing in iSCSI and FC environments and is implemented with in-kernel NVMe multipath.

The following support is available for the NVMe-oF host configuration for OL 9.2 with ONTAP:

- Support for NVMe over TCP (NVMe/TCP) in addition to NVMe/FC. The NetApp plug-in in the native nvme-cli package displays ONTAP details for both NVMe/FC and NVMe/TCP namespaces.
- Use of NVMe and SCSI co-existent traffic on the same host on a given host bus adapter (HBA), without the explicit dm-multipath settings to prevent claiming NVMe namespaces.

For additional details on supported configurations, see the Interoperability Matrix Tool.

Features

- Oracle Linux 9.2 has in-kernel NVMe multipath enabled for NVMe namespaces by default, therefore, there is no need for explicit settings.

Known limitations

SAN booting using the NVMe-oF protocol is currently not supported.

Validate software versions

You can use the following procedure to validate the minimum supported OL 9.2 software versions.

Steps

1. Install OL 9.2 GA on the server. After the installation is complete, verify that you are running the specified OL 9.2 GA kernel.

   ```
   # uname -r
   ```

   Example output:

   ```
   5.15.0-101.103.2.1.el9uek.x86_64
   ```

2. Install the `nvme-cli` package:

   ```
   # rpm -qa|grep nvme-cli
   ```

   Example output:

   ```
   nvme-cli-2.2.1-2.el9.x86_64
   ```

3. Install the `libnvme` package:

   ```
   #rpm -qa|grep libnvme
   ```

Example output

```
libnvme-1.2-2.el9.x86_64
```

4. On the Oracle Linux 9.2 host, check the `hostnqn` string at `/etc/nvme/hostnqn`:

```
# cat /etc/nvme/hostnqn
```

Example output:

```
nqn.2014-08.org.nvmexpress:uuid:bc59d14c-47f3-11eb-b93c-3a68dd48673f
```

5. Verify that the `hostnqn` string matches the `hostnqn` string for the corresponding subsystem on the ONTAP array:

```
::> vserver nvme subsystem host show -vserver vs_ol_nvme
```

Example output:

```
Vserver     Subsystem          Host NQN
----------- ---------------
----------------------------------------------------------
vs_nvme207   nvme_ss_ol_1    nqn.2014-08.org.nvmexpress:uuid:bc59d14c-
47f3-11eb-b93c-3a68dd48673f
```

If the `hostnqn` strings do not match, you can use the `vserver modify` command to update the `hostnqn` string on your corresponding ONTAP array subsystem to match the `hostnqn` string from `/etc/nvme/hostnqn` on the host.

Configure NVMe/FC

You can configure NVMe/FC for Broadcom/Emulex adapters or Marvell/Qlogic adapters.

Broadcom/Emulex

Steps

1. Verify that you are using the supported adapter model:

   ```
   # cat /sys/class/scsi_host/host*/modelname
   ```

   Example output:

   ```
   LPe32002-M2
   LPe32002-M2
   ```

   ```
   # cat /sys/class/scsi_host/host*/modeldesc
   ```

   Example output:

   ```
   Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
   Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
   ```

2. Verify that you are using the recommended Broadcom `lpfc` firmware and inbox driver:

   ```
   # cat /sys/class/scsi_host/host*/fwrev
   14.2.455.11, sli-4:2:c
   14.2.455.11, sli-4:2:c
   ```

   ```
   # cat /sys/module/lpfc/version
   0:14.2.0.5
   ```

   For the current list of supported adapter driver and firmware versions, see the Interoperability Matrix Tool.

3. Verify that `lpfc_enable_fc4_type` is set to `3`:

   ```
   # cat /sys/module/lpfc/parameters/lpfc_enable_fc4_type
   3
   ```

4. Verify that the initiator ports are up and running, and that you can see the target LIFs:

```
# cat /sys/class/fc_host/host*/port_name
0x100000109b3c081f
0x100000109b3c0820
```

```
# cat /sys/class/fc_host/host*/port_state
Online
Online
```

```
# cat /sys/class/scsi_host/host*/nvme_info
NVME Initiator Enabled
XRI Dist lpfc0 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc0 WWPN x100000109b1c1204 WWNN x200000109b1c1204 DID
x011d00 ONLINE
NVME RPORT WWPN x203800a098dfdd91 WWNN x203700a098dfdd91 DID x010c07
TARGET DISCSRVC ONLINE
NVME RPORT WWPN x203900a098dfdd91 WWNN x203700a098dfdd91 DID x011507
TARGET DISCSRVC ONLINE
NVME Statistics
LS: Xmt 0000000f78 Cmpl 0000000f78 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000002fe29bba Issue 000000002fe29bc4 OutIO
000000000000000a
abort 00001bc7 noxri 00000000 nondlp 00000000 qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 00001e15 Err 0000d906
NVME Initiator Enabled
XRI Dist lpfc1 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc1 WWPN x100000109b1c1205 WWNN x200000109b1c1205 DID
x011900 ONLINE
NVME RPORT WWPN x203d00a098dfdd91 WWNN x203700a098dfdd91 DID x010007
TARGET DISCSRVC ONLINE
NVME RPORT WWPN x203a00a098dfdd91 WWNN x203700a098dfdd91 DID x012a07
TARGET DISCSRVC ONLINE
NVME Statistics
LS: Xmt 0000000fa8 Cmpl 0000000fa8 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000002e14f170 Issue 000000002e14f17a OutIO
000000000000000a
abort 000016bb noxri 00000000 nondlp 00000000 qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 00001f50 Err 0000d9f8
```

Marvell/QLogic FC Adapter for NVMe/FC

The native inbox qla2xxx driver included in the OL 9.2 GA kernel has the latest upstream fixes. These fixes are essential for ONTAP support.

Steps

1. Verify that you are running the supported adapter driver and firmware versions:

```
# cat /sys/class/fc_host/host*/symbolic_name
QLE2742 FW:v9.12.00 DVR:v10.02.08.100-k
QLE2742 FW:v9.12.00 DVR:v10.02.08.100-k
```

2. Verify that `ql2xnvmeenable` is set. This enables the Marvell adapter to function as an NVMe/FC initiator:

```
# cat /sys/module/qla2xxx/parameters/ql2xnvmeenable
1
```

Enable 1MB I/O size (Optional)

ONTAP reports an MDTS (Max Data Transfer Size) of 8 in the Identify Controller data. This means the maximum I/O request size can be up to 1MB. To issue I/O requests of size 1MB for a Broadcom NVMe/FC host, you should increase the `lpfc` value of the `lpfc_sg_seg_cnt` parameter to 256 from the default value of 64.

> These steps don't apply to Qlogic NVMe/FC hosts.

Steps

1. Set the `lpfc_sg_seg_cnt` parameter to 256:

```
cat /etc/modprobe.d/lpfc.conf
```

You should see an output similar to the following example:

```
options lpfc lpfc_sg_seg_cnt=256
```

2. Run the `dracut -f` command, and reboot the host.
3. Verify that the value for `lpfc_sg_seg_cnt` is 256:

```
cat /sys/module/lpfc/parameters/lpfc_sg_seg_cnt
```

Configure NVMe/TCP

NVMe/TCP does not have auto-connect functionality. Therefore, if a path goes down and is not reinstated within the default time out period of 10 minutes, NVMe/TCP cannot automatically reconnect. To prevent a time out, you should set the retry period for failover events to at least 30 minutes.

Steps

1. Verify that the initiator port can fetch the discovery log page data across the supported NVMe/TCP LIFs:

   ```
   nvme discover -t tcp -w host-traddr -a traddr
   ```

   Example output:

```
# nvme discover -t tcp -w 192.168.167.5 -a 192.168.167.22

Discovery Log Number of Records 8, Generation counter 18
=====Discovery Log Entry 0======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  0
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.c680f5bcae1411ed8639d039ea951c46:discovery
traddr:  192.168.166.23
eflags:  explicit discovery connections, duplicate discovery information
sectype: none
=====Discovery Log Entry 1======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  1
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.c680f5bcae1411ed8639d039ea951c46:discovery
traddr:  192.168.166.22
eflags:  explicit discovery connections, duplicate discovery information
sectype: none
=====Discovery Log Entry 2======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  2
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.c680f5bcae1411ed8639d039ea951c46:discovery
traddr:  192.168.167.23
eflags:  explicit discovery connections, duplicate discovery information
sectype: none
..........
```

2. Verify that the other NVMe/TCP initiator-target LIF combinations can successfully fetch discovery log page data:

```
nvme discover -t tcp -w host-traddr -a traddr
```

Example output:

```
# nvme discover -t tcp -w 192.168.8.1 -a 192.168.8.48
# nvme discover -t tcp -w 192.168.8.1 -a 192.168.8.49
# nvme discover -t tcp -w 192.168.9.1 -a 192.168.9.48
# nvme discover -t tcp -w 192.168.9.1 -a 192.168.9.49
```

3. Run the `nvme connect-all` command across all the supported NVMe/TCP initiator-target LIFs across the nodes, and set the controller loss timeout period for at least 30 minutes or 1800 seconds:

```
nvme connect-all -t tcp -w host-traddr -a traddr -l 1800
```

Example output:

```
# nvme connect-all -t tcp -w 192.168.8.1 -a 192.168.8.48 -l 1800
# nvme connect-all -t tcp -w 192.168.8.1 -a 192.168.8.49 -l 1800
# nvme connect-all -t tcp -w 192.168.9.1 -a 192.168.9.48 -l 1800
# nvme connect-all -t tcp -w 192.168.9.1 -a 192.168.9.49 -l 1800
```

Validate NVMe-oF

You can use the following procedure to validate NVMe-oF.

Steps

1. Verify the following NVMe/FC settings on the OL 9.2 host:

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

```
# cat /sys/class/nvme-subsystem/nvme-subsys*/model
NetApp ONTAP Controller
NetApp ONTAP Controller
```

```
# cat /sys/class/nvme-subsystem/nvme-subsys*/iopolicy
round-robin
round-robin
```

2. Verify that the namespaces are created and correctly discovered on the host:

```
# nvme list
```

Example output:

```
Node          SN                   Model
---------------------------------------------------------
/dev/nvme0n1 814vWBNRwf9HAAAAAAAB NetApp ONTAP Controller
/dev/nvme0n2 814vWBNRwf9HAAAAAAAB NetApp ONTAP Controller
/dev/nvme0n3 814vWBNRwf9HAAAAAAAB NetApp ONTAP Controller



Namespace Usage    Format             FW             Rev
-----------------------------------------------------------
1                 85.90 GB / 85.90 GB  4 KiB + 0 B   FFFFFFFF
2                 85.90 GB / 85.90 GB  24 KiB + 0 B  FFFFFFFF
3                   85.90 GB / 85.90 GB  4 KiB + 0 B   FFFFFFFF
```

3. Verify that the controller state of each path is live and has the correct ANA status:

NVMe/FC

```
# nvme list-subsys /dev/nvme0n1
```

Example output:

```
nvme-subsys0 - NQN=nqn.1992-
08.com.netapp:sn.5f5f2c4aa73b11e9967e00a098df41bd:subsystem.nvme_ss_
ol_1
\
+- nvme0 fc traddr=nn-0x203700a098dfdd91:pn-0x203800a098dfdd91
host_traddr=nn-0x200000109b1c1204:pn-0x100000109b1c1204 live non-
optimized
+- nvme1 fc traddr=nn-0x203700a098dfdd91:pn-0x203900a098dfdd91
host_traddr=nn-0x200000109b1c1204:pn-0x100000109b1c1204 live non-
optimized
+- nvme2 fc traddr=nn-0x203700a098dfdd91:pn-0x203a00a098dfdd91
host_traddr=nn-0x200000109b1c1205:pn-0x100000109b1c1205 live
optimized
+- nvme3 fc traddr=nn-0x203700a098dfdd91:pn-0x203d00a098dfdd91
host_traddr=nn-0x200000109b1c1205:pn-0x100000109b1c1205 live
optimized
```

NVMe/TCP

```
nvme list-subsys /dev/nvme1n22
```

Example output

```
nvme-subsys1 - NQN=nqn.1992-
08.com.netapp:sn.68c036aaa3cf11edbb95d039ea243511:subsystem.tcp
\
 +- nvme2 tcp
traddr=192.168.8.49,trsvcid=4420,host_traddr=192.168.8.1 live
optimized
 +- nvme3 tcp
traddr=192.168.8.48,trsvcid=4420,host_traddr=192.168.8.1 live
optimized
 +- nvme6 tcp
traddr=192.168.9.49,trsvcid=4420,host_traddr=192.168.9.1 live non-
optimized
 +- nvme7 tcp
traddr=192.168.9.48,trsvcid=4420,host_traddr=192.168.9.1 live non-
optimized
```

4. Verify that the NetApp plug-in displays the correct values for each ONTAP namespace device:

Column

```
# nvme netapp ontapdevices -o column
```

Example output:

```
Device         Vserver   Namespace Path
----------------------- ------------------------------
/dev/nvme0n1   vs_ol_nvme  /vol/ol_nvme_vol_1_1_0/ol_nvme_ns
/dev/nvme0n2   vs_ol_nvme  /vol/ol_nvme_vol_1_0_0/ol_nvme_ns
/dev/nvme0n3   vs_ol_nvme  /vol/ol_nvme_vol_1_1_1/ol_nvme_ns




NSID       UUID                                   Size
------------------------------------------------------------
1          72b887b1-5fb6-47b8-be0b-33326e2542e2   85.90GB
2          04bf9f6e-9031-40ea-99c7-a1a61b2d7d08   85.90GB
3          264823b1-8e03-4155-80dd-e904237014a4   85.90GB
```

JSON

```
# nvme netapp ontapdevices -o json
```

Example output

```
{
"ONTAPdevices" : [
    {
        "Device" : "/dev/nvme0n1",
        "Vserver" : "vs_ol_nvme",
        "Namespace_Path" : "/vol/ol_nvme_vol_1_1_0/ol_nvme_ns",
        "NSID" : 1,
        "UUID" : "72b887b1-5fb6-47b8-be0b-33326e2542e2",
        "Size" : "85.90GB",
        "LBA_Data_Size" : 4096,
        "Namespace_Size" : 20971520
    },
    {
        "Device" : "/dev/nvme0n2",
        "Vserver" : "vs_ol_nvme",
        "Namespace_Path" : "/vol/ol_nvme_vol_1_0_0/ol_nvme_ns",
        "NSID" : 2,
        "UUID" : "04bf9f6e-9031-40ea-99c7-a1a61b2d7d08",
        "Size" : "85.90GB",
        "LBA_Data_Size" : 4096,
        "Namespace_Size" : 20971520
      },
      {
         "Device" : "/dev/nvme0n3",
         "Vserver" : "vs_ol_nvme",
         "Namespace_Path" : "/vol/ol_nvme_vol_1_1_1/ol_nvme_ns",
         "NSID" : 3,
         "UUID" : "264823b1-8e03-4155-80dd-e904237014a4",
         "Size" : "85.90GB",
         "LBA_Data_Size" : 4096,
         "Namespace_Size" : 20971520
       },
  ]
}
```


Known issues

There are no known issues.

NVMe-oF Host Configuration for Oracle Linux 9.1 with ONTAP

NVMe over Fabrics (NVMe-oF), including NVMe over Fibre Channel (NVMe/FC) and other transports, is supported with Oracle Linux (OL) 9.1 with Asymmetric Namespace Access (ANA). In NVMe-oF environments, ANA is the equivalent of ALUA multipathing in iSCSI and FC environments and is implemented with in-kernel NVMe multipath.

The following support is available for the NVMe-oF host configuration for OL 9.1 with ONTAP:

- Support for NVMe over TCP (NVMe/TCP) in addition to NVMe/FC. The NetApp plug-in in the native nvme-cli package displays ONTAP details for both NVMe/FC and NVMe/TCP namespaces.
- Use of NVMe and SCSI co-existent traffic on the same host on a given host bus adapter (HBA), without the explicit dm-multipath settings to prevent claiming NVMe namespaces.

For additional details on supported configurations, see the Interoperability Matrix Tool.

Features

- Oracle Linux 9.1 has in-kernel NVMe multipath enabled for NVMe namespaces by default, therefore, there is no need for explicit settings.

Known limitations

SAN booting using the NVMe-oF protocol is currently not supported.

Validate software versions

You can use the following procedure to validate the minimum supported OL 9.1 software versions.

Steps

1. Install OL 9.1 GA on the server. After the installation is complete, verify that you are running the specified OL 9.1 GA kernel.

   ```
   # uname -r
   ```

   Example output:

   ```
   5.15.0-3.60.5.1.el9uek.x86_64
   ```

2. Install the `nvme-cli` package:

   ```
   # rpm -qa|grep nvme-cli
   ```

   Example output:

   ```
   nvme-cli-2.0-4.el9.x86_64
   ```

3. Install the `libnvme` package:

   ```
   #rpm -qa|grep libnvme
   ```

Example output

```
libnvme-1.0-5.el9.x86_64.rpm
```

4. On the Oracle Linux 9.1 host, check the `hostnqn` string at `/etc/nvme/hostnqn`:

```
# cat /etc/nvme/hostnqn
```

Example output:

```
nqn.2014-08.org.nvmexpress:uuid:bc59d14c-47f3-11eb-b93c-3a68dd48673f
```

5. Verify that the `hostnqn` string matches the `hostnqn` string for the corresponding subsystem on the ONTAP array:

```
::> vserver nvme subsystem host show -vserver vs_ol_nvme
```

Example output:

```
Vserver     Subsystem          Host NQN
----------- ---------------
----------------------------------------------------------
vs_ol_nvme  nvme_ss_ol_1    nqn.2014-08.org.nvmexpress:uuid:bc59d14c-
47f3-11eb-b93c-3a68dd48673f
```

> If the `hostnqn` strings do not match, you can use the `vserver modify` command to update the `hostnqn` string on your corresponding ONTAP array subsystem to match the `hostnqn` string from `/etc/nvme/hostnqn` on the host.

Configure NVMe/FC

You can configure NVMe/FC for Broadcom/Emulex adapters or Marvell/Qlogic adapters.

Broadcom/Emulex

Steps

1. Verify that you are using the supported adapter model:

```
# cat /sys/class/scsi_host/host*/modelname
```

Example output:

```
LPe32002-M2
LPe32002-M2
```

```
# cat /sys/class/scsi_host/host*/modeldesc
```

Example output:

```
Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
```

2. Verify that you are using the recommended Broadcom `lpfc` firmware and inbox driver:

```
# cat /sys/class/scsi_host/host*/fwrev
12.8.614.23, sli-4:2:c
12.8.614.23, sli-4:2:c


# cat /sys/module/lpfc/version
0:14.0.0.1
```

For the current list of supported adapter driver and firmware versions, see the Interoperability Matrix Tool.

3. Verify that `lpfc_enable_fc4_type` is set to `3`:

```
# cat /sys/module/lpfc/parameters/lpfc_enable_fc4_type
3
```

4. Verify that the initiator ports are up and running, and that you can see the target LIFs:

```
# cat /sys/class/fc_host/host*/port_name
0x100000109b3c081f
0x100000109b3c0820

# cat /sys/class/fc_host/host*/port_state
Online
Online
# cat /sys/class/scsi_host/host*/nvme_info
# cat /sys/class/scsi_host/host*/nvme_info
NVME Initiator Enabled
XRI Dist lpfc0 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc0 WWPN x100000109b1c1204 WWNN x200000109b1c1204 DID
x011d00 ONLINE
NVME RPORT WWPN x203800a098dfdd91 WWNN x203700a098dfdd91 DID x010c07
TARGET DISCSRVC ONLINE
NVME RPORT WWPN x203900a098dfdd91 WWNN x203700a098dfdd91 DID x011507
TARGET DISCSRVC ONLINE
NVME Statistics
LS: Xmt 0000000f78 Cmpl 0000000f78 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000002fe29bba Issue 000000002fe29bc4 OutIO
000000000000000a
abort 00001bc7 noxri 00000000 nondlp 00000000 qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 00001e15 Err 0000d906
NVME Initiator Enabled
XRI Dist lpfc1 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc1 WWPN x100000109b1c1205 WWNN x200000109b1c1205 DID
x011900 ONLINE
NVME RPORT WWPN x203d00a098dfdd91 WWNN x203700a098dfdd91 DID x010007
TARGET DISCSRVC ONLINE
NVME RPORT WWPN x203a00a098dfdd91 WWNN x203700a098dfdd91 DID x012a07
TARGET DISCSRVC ONLINE
NVME Statistics
LS: Xmt 0000000fa8 Cmpl 0000000fa8 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000002e14f170 Issue 000000002e14f17a OutIO
000000000000000a
abort 000016bb noxri 00000000 nondlp 00000000 qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 00001f50 Err 0000d9f8
```

Marvell/QLogic FC Adapter for NVMe/FC

The native inbox qla2xxx driver included in the OL 9.1 GA kernel has the latest upstream fixes. These fixes are essential for ONTAP support.

Steps

1. Verify that you are running the supported adapter driver and firmware versions:

```
# cat /sys/class/fc_host/host*/symbolic_name
QLE2742 FW:v9.18.02 DVR:v10.02.00.106-k
QLE2742 FW:v9.18.02 DVR:v10.02.00.106-k
```

2. Verify that `ql2xnvmeenable` is set. This enables the Marvell adapter to function as an NVMe/FC initiator:

```
# cat /sys/module/qla2xxx/parameters/ql2xnvmeenable
1
```

Enable 1MB I/O size (Optional)

ONTAP reports an MDTS (Max Data Transfer Size) of 8 in the Identify Controller data. This means the maximum I/O request size can be up to 1MB. To issue I/O requests of size 1MB for a Broadcom NVMe/FC host, you should increase the `lpfc` value of the `lpfc_sg_seg_cnt` parameter to 256 from the default value of 64.

> These steps don't apply to Qlogic NVMe/FC hosts.

Steps

1. Set the `lpfc_sg_seg_cnt` parameter to 256:

```
cat /etc/modprobe.d/lpfc.conf
```

You should see an output similar to the following example:

```
options lpfc lpfc_sg_seg_cnt=256
```

2. Run the `dracut -f` command, and reboot the host.
3. Verify that the value for `lpfc_sg_seg_cnt` is 256:

```
cat /sys/module/lpfc/parameters/lpfc_sg_seg_cnt
```

Configure NVMe/TCP

NVMe/TCP does not have auto-connect functionality. Therefore, if a path goes down and is not reinstated within the default time out period of 10 minutes, NVMe/TCP cannot automatically reconnect. To prevent a time out, you should set the retry period for failover events to at least 30 minutes.

Steps

1. Verify that the initiator port can fetch the discovery log page data across the supported NVMe/TCP LIFs:

```
nvme discover -t tcp -w host-traddr -a traddr
```

Example output:

```
#  nvme discover -t tcp -w 192.168.6.13 -a 192.168.6.15
Discovery Log Number of Records 6, Generation counter 8
=====Discovery Log Entry 0======
trtype: tcp
adrfam: ipv4
subtype: unrecognized
treq: not specified
portid: 0
trsvcid: 8009
subnqn: nqn.1992-
08.com.netapp:sn.1c6ac66338e711eda41dd039ea3ad566:discovery
traddr: 192.168.6.17
sectype: none
=====Discovery Log Entry 1======
trtype: tcp
adrfam: ipv4
subtype: unrecognized
treq: not specified
portid: 1
trsvcid: 8009
subnqn: nqn.1992-
08.com.netapp:sn.1c6ac66338e711eda41dd039ea3ad566:discovery
traddr: 192.168.5.17
sectype: none
=====Discovery Log Entry 2======
trtype: tcp
adrfam: ipv4
subtype: unrecognized
treq: not specified
portid: 2
trsvcid: 8009
subnqn: nqn.1992-
08.com.netapp:sn.1c6ac66338e711eda41dd039ea3ad566:discovery
traddr: 192.168.6.15
sectype: none
=====Discovery Log Entry 3======
trtype: tcp
adrfam: ipv4
```

```
subtype: nvme subsystem
treq: not specified
portid: 0
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.1c6ac66338e711eda41dd039ea3ad566:subsystem.host_95
traddr: 192.168.6.17
sectype: none
..........
```

2. Verify that the other NVMe/TCP initiator-target LIF combinations can successfully fetch discovery log page data:

```
nvme discover -t tcp -w host-traddr -a traddr
```

Example output:

```
# nvme discover -t tcp -w 192.168.5.13 -a 192.168.5.15
# nvme discover -t tcp -w 192.168.5.13 -a 192.168.5.17
# nvme discover -t tcp -w 192.168.6.13 -a 192.168.6.15
# nvme discover -t tcp -w 192.168.6.13 -a 192.168.6.17
```

3. Run the `nvme connect-all` command across all the supported NVMe/TCP initiator-target LIFs across the nodes, and set the controller loss timeout period for at least 30 minutes or 1800 seconds:

```
nvme connect-all -t tcp -w host-traddr -a traddr -l 1800
```

Example output:

```
# nvme connect-all -t tcp -w 192.168.5.13 -a 192.168.5.15 -l 1800
# nvme connect-all -t tcp -w 192.168.5.13 -a 192.168.5.17 -l 1800
# nvme connect-all -t tcp -w 192.168.6.13 -a 192.168.6.15 -l 1800
# nvme connect-all -t tcp -w 192.168.6.13 -a 192.168.6.17 -l 1800
```

Validate NVMe-oF

You can use the following procedure to validate NVMe-oF.

Steps

1. Verify the following NVMe/FC settings on the OL 9.1 host:

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

```
# cat /sys/class/nvme-subsystem/nvme-subsys*/model
NetApp ONTAP Controller
NetApp ONTAP Controller
```

```
# cat /sys/class/nvme-subsystem/nvme-subsys*/iopolicy
round-robin
round-robin
```

2. Verify that the namespaces are created and correctly discovered on the host:

```
# nvme list
```

Example output:

```
Node         SN                   Model
---------------------------------------------------------
/dev/nvme0n1 814vWBNRwf9HAAAAAAAB NetApp ONTAP Controller
/dev/nvme0n2 814vWBNRwf9HAAAAAAAB NetApp ONTAP Controller
/dev/nvme0n3 814vWBNRwf9HAAAAAAAB NetApp ONTAP Controller


Namespace Usage    Format             FW             Rev
-----------------------------------------------------------
1                 85.90 GB / 85.90 GB  4 KiB + 0 B   FFFFFFFF
2                 85.90 GB / 85.90 GB  24 KiB + 0 B  FFFFFFFF
3                 85.90 GB / 85.90 GB  4 KiB + 0 B   FFFFFFFF
```

3. Verify that the controller state of each path is live and has the correct ANA status:

NVMe/FC

```
# nvme list-subsys /dev/nvme0n1
```

Example output:

```
nvme-subsys0 - NQN=nqn.1992-
08.com.netapp:sn.5f5f2c4aa73b11e9967e00a098df41bd:subsystem.nvme_ss_
ol_1
\
+- nvme0 fc traddr=nn-0x203700a098dfdd91:pn-0x203800a098dfdd91
host_traddr=nn-0x200000109b1c1204:pn-0x100000109b1c1204 live
inaccessible
+- nvme1 fc traddr=nn-0x203700a098dfdd91:pn-0x203900a098dfdd91
host_traddr=nn-0x200000109b1c1204:pn-0x100000109b1c1204 live
inaccessible
+- nvme2 fc traddr=nn-0x203700a098dfdd91:pn-0x203a00a098dfdd91
host_traddr=nn-0x200000109b1c1205:pn-0x100000109b1c1205 live
optimized
+- nvme3 fc traddr=nn-0x203700a098dfdd91:pn-0x203d00a098dfdd91
host_traddr=nn-0x200000109b1c1205:pn-0x100000109b1c1205 live
optimized
```

NVMe/TCP

```
nvme list-subsys /dev/nvme1n22
```

Example output

```
nvme-subsys1 - NQN=nqn.1992-
08.com.netapp:sn.68c036aaa3cf11edbb95d039ea243511:subsystem.tcp
\
+- nvme2 tcp
traddr=192.168.8.49,trsvcid=4420,host_traddr=192.168.8.1 live
optimized
+- nvme3 tcp
traddr=192.168.8.48,trsvcid=4420,host_traddr=192.168.8.1 live
optimized
+- nvme6 tcp
traddr=192.168.9.49,trsvcid=4420,host_traddr=192.168.9.1 live non-
optimized
+- nvme7 tcp
traddr=192.168.9.48,trsvcid=4420,host_traddr=192.168.9.1 live non-
optimized
```

4. Verify that the NetApp plug-in displays the correct values for each ONTAP namespace device:

Column

```
# nvme netapp ontapdevices -o column
```

Example output:

```
Device          Vserver   Namespace Path
----------------------- ------------------------------
/dev/nvme0n1   vs_ol_nvme  /vol/ol_nvme_vol_1_1_0/ol_nvme_ns
/dev/nvme0n2   vs_ol_nvme  /vol/ol_nvme_vol_1_0_0/ol_nvme_ns
/dev/nvme0n3   vs_ol_nvme  /vol/ol_nvme_vol_1_1_1/ol_nvme_ns


NSID       UUID                                   Size
------------------------------------------------------------
1          72b887b1-5fb6-47b8-be0b-33326e2542e2   85.90GB
2          04bf9f6e-9031-40ea-99c7-a1a61b2d7d08   85.90GB
3          264823b1-8e03-4155-80dd-e904237014a4   85.90GB
```

JSON

```
# nvme netapp ontapdevices -o json
```

Example output

```
{
"ONTAPdevices" : [
    {
        "Device" : "/dev/nvme0n1",
        "Vserver" : "vs_ol_nvme",
        "Namespace_Path" : "/vol/ol_nvme_vol_1_1_0/ol_nvme_ns",
        "NSID" : 1,
        "UUID" : "72b887b1-5fb6-47b8-be0b-33326e2542e2",
        "Size" : "85.90GB",
        "LBA_Data_Size" : 4096,
        "Namespace_Size" : 20971520
    },
    {
        "Device" : "/dev/nvme0n2",
        "Vserver" : "vs_ol_nvme",
        "Namespace_Path" : "/vol/ol_nvme_vol_1_0_0/ol_nvme_ns",
        "NSID" : 2,
        "UUID" : "04bf9f6e-9031-40ea-99c7-a1a61b2d7d08",
        "Size" : "85.90GB",
        "LBA_Data_Size" : 4096,
        "Namespace_Size" : 20971520
      },
      {
         "Device" : "/dev/nvme0n3",
         "Vserver" : "vs_ol_nvme",
         "Namespace_Path" : "/vol/ol_nvme_vol_1_1_1/ol_nvme_ns",
         "NSID" : 3,
         "UUID" : "264823b1-8e03-4155-80dd-e904237014a4",
         "Size" : "85.90GB",
         "LBA_Data_Size" : 4096,
         "Namespace_Size" : 20971520
       },
  ]
}
```

Known issues

The NVMe-oF host configuration for OL 9.1 with ONTAP release has the following known issues:

NetApp Bug ID	Title	Description
1536937	`nvme list-subsys` command prints repeated NVMe controllers for a subsystem	The `nvme list-subsys` command should return a unique list of NVMe controllers associated with a given subsystem. In Oracle Linux 9.1, the `nvme list-subsys` command returns NVMe controllers with the respective asymmetric namespace access (ANA) state for all namespaces that belong to a given subsystem. However, it would be useful to display unique NVMe controller entries with the path state if you list the subsystem command syntax for a given namespace because the ANA state is a per-namespace attribute.
1539101	Oracle Linux 9.1 NVMe-oF hosts fail to create a persistent discovery controller	On Oracle Linux 9.1 NVMe-oF hosts, you can use the `nvme discover -p` command to create Persistent Discovery Controllers (PDCs). When this command is used, one PDC should be created per initiator-target combination. However, if you are running Oracle Linux 9.1 on an NVMe-oF host, PDC creation fails when the `nvme discover -p` command is executed.

NVMe/FC Host Configuration for Oracle Linux 9.0 with ONTAP

NVMe over Fabrics (NVMe-oF), including NVMe/FC and other transports, is supported with Oracle Linux (OL) 9.0 with Asymmetric Namespace Access (ANA). In NVMe-oF environments, ANA is the equivalent of ALUA multipathing in iSCSI and FC environments and is implemented with in-kernel NVMe multipath.

For additional details on supported configurations, see the Interoperability Matrix Tool.

Features

- Oracle Linux 9.0 has in-kernel NVMe multipath enabled for NVMe namespaces by default, therefore, there is no need for explicit settings.

Known limitations

SAN booting using the NVMe-oF protocol is currently not supported.

Validate software versions

You can validate the minimum supported OL 9.0 software versions using the following procedure.

Steps

1. Install OL 9.0 GA on the server. After the installation is complete, verify that you are running the specified OL 9.0 GA kernel.

   ```
   # uname -r
   ```

   Example output:

   ```
   5.15.0-0.30.19.el9uek.x86_64
   ```

2. Install the `nvme-cli` package:

   ```
   # rpm -qa|grep nvme-cli
   ```

   Example output:

   ```
   nvme-cli-1.16-3.el9.x86_64
   ```

3. On the Oracle Linux 9.0 host, check the `hostnqn` string at `/etc/nvme/hostnqn`:

   ```
   # cat /etc/nvme/hostnqn
   ```

   Example output:

   ```
   nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0032-3310-8033-b8c04f4c5132
   ```

4. Verify that the `hostnqn` string matches the `hostnqn` string for the corresponding subsystem on the ONTAP array:

   ```
   ::> vserver nvme subsystem host show -vserver vs_ol_nvme
   ```

   Example output:

   ```
   Vserver     Subsystem          Host NQN
   ----------- ---------------
   ----------------------------------------------------------
   vs_ol_nvme  nvme_ss_ol_1    nqn.2014-08.org.nvmexpress:uuid:4c4c4544-
   0032-3310-8033-b8c04f4c5132
   ```

> If the `hostnqn` strings do not match, you can use the `vserver modify` command to update the `hostnqn` string on your corresponding ONTAP array subsystem to match the `hostnqn` string from `/etc/nvme/hostnqn` on the host.

Configure NVMe/FC

You can configure NVMe/FC for Broadcom/Emulex or Marvell/Qlogic adapters.

Broadcom/Emulex

Steps

1. Verify that you are using the supported adapter model.

   ```
   # cat /sys/class/scsi_host/host*/modelname
   ```

   Example output:

   ```
   LPe32002-M2
   LPe32002-M2
   ```

   ```
   # cat /sys/class/scsi_host/host*/modeldesc
   ```

   Example output:

   ```
   Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
   Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
   ```

2. Verify that you are using the recommended Broadcom `lpfc` firmware and inbox driver:

   ```
   # cat /sys/class/scsi_host/host*/fwrev
   14.0.505.11, sli-4:2:c
   14.0.505.11, sli-4:2:c

   # cat /sys/module/lpfc/version
   0:12.8.0.11
   ```

   For the current list of supported adapter driver and firmware versions, see the Interoperability Matrix Tool.

3. Verify that `lpfc_enable_fc4_type` is set to `3`:

   ```
   # cat /sys/module/lpfc/parameters/lpfc_enable_fc4_type
   3
   ```

4. Verify that the initiator ports are up and running, and that you can see the target LIFs:

```
# cat /sys/class/fc_host/host*/port_name
0x100000109b1c1204
0x100000109b1c1205
# cat /sys/class/fc_host/host*/port_state
Online
Online
# cat /sys/class/scsi_host/host*/nvme_info
NVME Initiator Enabled
XRI Dist lpfc0 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc0 WWPN x100000109b1c1204 WWNN x200000109b1c1204 DID
x011d00 ONLINE
NVME RPORT WWPN x203800a098dfdd91 WWNN x203700a098dfdd91 DID x010c07
TARGET DISCSRVC ONLINE
NVME RPORT WWPN x203900a098dfdd91 WWNN x203700a098dfdd91 DID x011507
TARGET DISCSRVC ONLINE
NVME Statistics
LS: Xmt 0000000f78 Cmpl 0000000f78 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000002fe29bba Issue 000000002fe29bc4 OutIO
000000000000000a
abort 00001bc7 noxri 00000000 nondlp 00000000 qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 00001e15 Err 0000d906
NVME Initiator Enabled
XRI Dist lpfc1 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc1 WWPN x100000109b1c1205 WWNN x200000109b1c1205 DID
x011900 ONLINE
NVME RPORT WWPN x203d00a098dfdd91 WWNN x203700a098dfdd91 DID x010007
TARGET DISCSRVC ONLINE
NVME RPORT WWPN x203a00a098dfdd91 WWNN x203700a098dfdd91 DID x012a07
TARGET DISCSRVC ONLINE
NVME Statistics
LS: Xmt 0000000fa8 Cmpl 0000000fa8 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000002e14f170 Issue 000000002e14f17a OutIO
000000000000000a
abort 000016bb noxri 00000000 nondlp 00000000 qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 00001f50 Err 0000d9f8
```

Marvell/QLogic FC Adapter for NVMe/FC

The native inbox qla2xxx driver included in the OL 9.0 GA kernel has the latest upstream fixes. These fixes are essential for ONTAP support.

Steps

1. Verify that you are running the supported adapter driver and firmware versions:

```
# cat /sys/class/fc_host/host*/symbolic_name
QLE2742 FW:v9.08.02 DVR:v10.02.00.106-k
QLE2742 FW:v9.08.02 DVR:v10.02.00.106-k
```

2. Verify that `ql2xnvmeenable` is set which enables the Marvell adapter to function as an NVMe/FC initiator:

```
# cat /sys/module/qla2xxx/parameters/ql2xnvmeenable
1
```

Enable 1MB I/O size (Optional)

ONTAP reports an MDTS (Max Data Transfer Size) of 8 in the Identify Controller data. This means the maximum I/O request size can be up to 1MB. To issue I/O requests of size 1MB for a Broadcom NVMe/FC host, you should increase the `lpfc` value of the `lpfc_sg_seg_cnt` parameter to 256 from the default value of 64.

> These steps don't apply to Qlogic NVMe/FC hosts.

Steps

1. Set the `lpfc_sg_seg_cnt` parameter to 256:

```
cat /etc/modprobe.d/lpfc.conf
```

You should see an output similar to the following example:

```
options lpfc lpfc_sg_seg_cnt=256
```

2. Run the `dracut -f` command, and reboot the host.
3. Verify that the value for `lpfc_sg_seg_cnt` is 256:

```
cat /sys/module/lpfc/parameters/lpfc_sg_seg_cnt
```

Configure NVMe/TCP

NVMe/TCP does not have auto-connect functionality. Therefore, if a path goes down and is not reinstated within the default time out period of 10 minutes, NVMe/TCP cannot automatically reconnect. To prevent a time out, you should set the retry period for failover events to at least 30 minutes.

Steps

1. Verify that the initiator port can fetch the discovery log page data across the supported NVMe/TCP LIFs:

```
nvme discover -t tcp -w host-traddr -a traddr
```

Example output:

```
#  nvme discover -t tcp -w 192.168.6.13 -a 192.168.6.15
Discovery Log Number of Records 6, Generation counter 8
=====Discovery Log Entry 0======
trtype: tcp
adrfam: ipv4
subtype: unrecognized
treq: not specified
portid: 0
trsvcid: 8009
subnqn: nqn.1992-
08.com.netapp:sn.1c6ac66338e711eda41dd039ea3ad566:discovery
traddr: 192.168.6.17
sectype: none
=====Discovery Log Entry 1======
trtype: tcp
adrfam: ipv4
subtype: unrecognized
treq: not specified
portid: 1
trsvcid: 8009
subnqn: nqn.1992-
08.com.netapp:sn.1c6ac66338e711eda41dd039ea3ad566:discovery
traddr: 192.168.5.17
sectype: none
=====Discovery Log Entry 2======
trtype: tcp
adrfam: ipv4
subtype: unrecognized
treq: not specified
portid: 2
trsvcid: 8009
subnqn: nqn.1992-
08.com.netapp:sn.1c6ac66338e711eda41dd039ea3ad566:discovery
traddr: 192.168.6.15
sectype: none
=====Discovery Log Entry 3======
trtype: tcp
adrfam: ipv4
```

```
subtype: nvme subsystem
treq: not specified
portid: 0
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.1c6ac66338e711eda41dd039ea3ad566:subsystem.host_95
traddr: 192.168.6.17
sectype: none
..........
```

2. Verify that the other NVMe/TCP initiator-target LIF combinations are able to successfully fetch discovery log page data.

   ```
   nvme discover -t tcp -w host-traddr -a traddr
   ```

   Example output:

   ```
   # nvme discover -t tcp -w 192.168.5.13 -a 192.168.5.15
   # nvme discover -t tcp -w 192.168.5.13 -a 192.168.5.17
   # nvme discover -t tcp -w 192.168.6.13 -a 192.168.6.15
   # nvme discover -t tcp -w 192.168.6.13 -a 192.168.6.17
   ```

3. Run the `nvme connect-all` command across all the supported NVMe/TCP initiator-target LIFs across the nodes, and set the controller loss timeout period for at least 30 minutes or 1800 seconds:

   ```
   nvme connect-all -t tcp -w host-traddr -a traddr -l 1800
   ```

   Example output:

   ```
   # nvme connect-all -t tcp -w 192.168.5.13 -a 192.168.5.15 -l 1800
   # nvme connect-all -t tcp -w 192.168.5.13 -a 192.168.5.17 -l 1800
   # nvme connect-all -t tcp -w 192.168.6.13 -a 192.168.6.15 -l 1800
   # nvme connect-all -t tcp -w 192.168.6.13 -a 192.168.6.17 -l 1800
   ```

Validate NVMe-oF

You can use the following procedure to validate NVMe-oF.

Steps

1. Verify the following NVMe/FC settings on the OL 9.0 host:

```
# cat /sys/module/nvme_core/parameters/multipath
Y
# cat /sys/class/nvme-subsystem/nvme-subsys*/model
NetApp ONTAP Controller
NetApp ONTAP Controller
# cat /sys/class/nvme-subsystem/nvme-subsys*/iopolicy
round-robin
round-robin
```

2. Verify that the namespaces are created and correctly discovered on the host:

```
# nvme list
```

Example output:

```
Node         SN                   Model
---------------------------------------------------------
/dev/nvme0n1 814vWBNRwf9HAAAAAAAB NetApp ONTAP Controller
/dev/nvme0n2 814vWBNRwf9HAAAAAAAB NetApp ONTAP Controller
/dev/nvme0n3 814vWBNRwf9HAAAAAAAB NetApp ONTAP Controller


Namespace Usage    Format             FW             Rev
-----------------------------------------------------------
1                 85.90 GB / 85.90 GB  4 KiB + 0 B   FFFFFFFF
2                 85.90 GB / 85.90 GB  24 KiB + 0 B  FFFFFFFF
3                 85.90 GB / 85.90 GB  4 KiB + 0 B   FFFFFFFF
```

3. Verify that the controller state of each path is live and has the correct ANA status:

```
# nvme list-subsys /dev/nvme0n1
```

Example output:

```
nvme-subsys0 - NQN=nqn.1992-
08.com.netapp:sn.5f5f2c4aa73b11e9967e00a098df41bd:subsystem.nvme_ss_ol_1
\
+- nvme0 fc traddr=nn-0x203700a098dfdd91:pn-0x203800a098dfdd91
host_traddr=nn-0x200000109b1c1204:pn-0x100000109b1c1204 live
inaccessible
+- nvme1 fc traddr=nn-0x203700a098dfdd91:pn-0x203900a098dfdd91
host_traddr=nn-0x200000109b1c1204:pn-0x100000109b1c1204 live
inaccessible
+- nvme2 fc traddr=nn-0x203700a098dfdd91:pn-0x203a00a098dfdd91
host_traddr=nn-0x200000109b1c1205:pn-0x100000109b1c1205 live optimized
+- nvme3 fc traddr=nn-0x203700a098dfdd91:pn-0x203d00a098dfdd91
host_traddr=nn-0x200000109b1c1205:pn-0x100000109b1c1205 live optimized
```

4. Verify that the NetApp plug-in displays the correct values for each ONTAP namespace device:

```
# nvme netapp ontapdevices -o column
```

Example output:

```
Device        Vserver   Namespace Path
----------------------- ------------------------------
/dev/nvme0n1   vs_ol_nvme  /vol/ol_nvme_vol_1_1_0/ol_nvme_ns
/dev/nvme0n2   vs_ol_nvme  /vol/ol_nvme_vol_1_0_0/ol_nvme_ns
/dev/nvme0n3   vs_ol_nvme  /vol/ol_nvme_vol_1_1_1/ol_nvme_ns


NSID       UUID                                   Size
------------------------------------------------------------
1          72b887b1-5fb6-47b8-be0b-33326e2542e2   85.90GB
2          04bf9f6e-9031-40ea-99c7-a1a61b2d7d08   85.90GB
3          264823b1-8e03-4155-80dd-e904237014a4   85.90GB
```

```
# nvme netapp ontapdevices -o json
{
"ONTAPdevices" : [
    {
        "Device" : "/dev/nvme0n1",
        "Vserver" : "vs_ol_nvme",
        "Namespace_Path" : "/vol/ol_nvme_vol_1_1_0/ol_nvme_ns",
        "NSID" : 1,
        "UUID" : "72b887b1-5fb6-47b8-be0b-33326e2542e2",
        "Size" : "85.90GB",
        "LBA_Data_Size" : 4096,
        "Namespace_Size" : 20971520
    },
    {
        "Device" : "/dev/nvme0n2",
        "Vserver" : "vs_ol_nvme",
        "Namespace_Path" : "/vol/ol_nvme_vol_1_0_0/ol_nvme_ns",
        "NSID" : 2,
        "UUID" : "04bf9f6e-9031-40ea-99c7-a1a61b2d7d08",
        "Size" : "85.90GB",
        "LBA_Data_Size" : 4096,
        "Namespace_Size" : 20971520
      },
      {
         "Device" : "/dev/nvme0n3",
         "Vserver" : "vs_ol_nvme",
         "Namespace_Path" : "/vol/ol_nvme_vol_1_1_1/ol_nvme_ns",
         "NSID" : 3,
         "UUID" : "264823b1-8e03-4155-80dd-e904237014a4",
         "Size" : "85.90GB",
         "LBA_Data_Size" : 4096,
         "Namespace_Size" : 20971520
       },
  ]
}
```

Known issues

The NVMe-oF host configuration for Oracle Linux 9.0 with ONTAP has the following known issues:

NetApp Bug ID	Title	Description
1517321	Oracle Linux 9.0 NVMe-oF Hosts create duplicate Persistent Discovery Controllers	On Oracle Linux 9.0 NVMe over Fabrics (NVMe-oF) hosts, you can use the `nvme discover -p` command to create Persistent Discovery Controllers (PDCs). When this command is used, only one PDC should be created per initiator-target combination. However, if you are running ONTAP 9.10.1 and Oracle Linux 9.0 with an NVMe-oF host, a duplicate PDC is created each time `nvme discover -p` is executed. This leads to unnecessary usage of resources on both the host and the target.

Oracle Linux 8

NVMe-oF Host Configuration for Oracle Linux 8.10 with ONTAP

NetApp SAN host configurations support the NVMe over Fabrics (NVMe-oF) protocol with Asymmetric Namespace Access (ANA). In NVMe-oF environments, ANA is equivalent to asymmetric logical unit access (ALUA) multipathing in iSCSI and FCP environments. ANA is implemented using the in-kernel NVMe multipath feature.

About this task

You can use the following support and features with the NVMe-oF host configuration for Oracle Linux 8.10. You should also review the known limitations before starting the configuration process.

- Support available:
  - Support for NVMe over TCP (NVMe/TCP) and NVMe over Fibre Channel (NVMe/FC). This gives the NetApp plug-in in the native nvme-cli package the capability to display the ONTAP information for both NVMe/FC and NVMe/TCP namespaces.

    Depending on your host configuration, you configure NNMe/FC, NVMe/TCP, or both protocols.
  - Running NVMe and SCSI traffic simultaneously on the same host. For example, you can configure dm-multipath for SCSI `mpath` devices for SCSI LUNs and use NVMe multipath to configure NVMe-oF namespace devices on the host.

  For additional details on supported configurations, see the Interoperability Matrix Tool.
- Features available:
  - The in-kernel NVMe multipath feature is enabled for NVMe namespaces by default in Oracle Linux 8.10. You don’t need to configure explicit settings.
- Known limitations:
  - SAN booting using the NVMe-oF protocol is currently not supported.

    ◦ NetApp sanlun host utility support isn't available for NVMe-oF on an Oracle Linux 8.10 host. Instead, you can rely on the NetApp plug-in included in the native `nvme-cli` package for all NVMe-oF transports.

Validate software versions

Validate the minimum supported software versions for Oracle Linux 8.10.

Steps

1. Install Oracle Linux 8.10 GA on the server. After the installation is complete, verify that you are running the specified Oracle Linux 8.10 GA kernel:

```
uname -r
```

```
5.15.0-206.153.7.1.el8uek.x86_64
```

2. Install the `nvme-cli` package:

```
rpm -qa|grep nvme-cli
```

```
nvme-cli-1.16-9.el8.x86_64
```

3. On the Oracle Linux 8.10 host, check the `hostnqn` string at `/etc/nvme/hostnqn`:

```
cat /etc/nvme/hostnqn
```

```
nqn.2014-08.org.nvmexpress:uuid:edd38060-00f7-47aa-a9dc-4d8ae0cd969a
```

4. Verify that `hostnqn` on the Oracle Linux 8.10 host matches `hostnqn` for the corresponding subsystem on the ONTAP array:

```
vserver nvme subsystem host show -vserver vs_coexistence_LPE36002
```

Show example

```
Vserver Subsystem Priority  Host NQN
------- --------- --------
------------------------------------------------
vs_coexistence_LPE36002
        nvme
                  regular   nqn.2014-
08.org.nvmexpress:uuid:edd38060-00f7-47aa-a9dc-4d8ae0cd969a
        nvme1
                  regular   nqn.2014-
08.org.nvmexpress:uuid:edd38060-00f7-47aa-a9dc-4d8ae0cd969a
        nvme2
                  regular   nqn.2014-
08.org.nvmexpress:uuid:edd38060-00f7-47aa-a9dc-4d8ae0cd969a
        nvme3
                  regular   nqn.2014-
08.org.nvmexpress:uuid:edd38060-00f7-47aa-a9dc-4d8ae0cd969a
4 entries were displayed.
```

If the `hostnqn` strings don't match, use the `vserver modify` command to update the `hostnqn` string on your corresponding ONTAP array subsystem to match the `hostnqn` string from `/etc/nvme/hostnqn` on the host.

5. If you intend to run both NVMe and SCSI co-existent traffic on the same host, NetApp recommends using the in-kernel NVMe multipath for ONTAP namespaces and `dm-multipath` for ONTAP LUNs respectively. This should exclude the ONTAP namespaces from `dm-multipath` and prevent `dm-multipath` from claiming the ONTAP namespace devices:
   a. Add the `enable_foreign` setting to the `/etc/multipath.conf` file:

   ```
   # cat /etc/multipath.conf
   defaults {
     enable_foreign     NONE
   }
   ```

   b. Restart the `multipathd` daemon to apply the new setting:

   ```
   systemctl restart multipathd
   ```

Configure NVMe/FC

You can configure NVMe/FC with Broadcom/Emulex FC or Marvell/Qlogic FC adapters. For NVMe/FC configured with a Broadcom adapter, you can enable I/O requests of size 1MB.

Broadcom/Emulex

Configure NVMe/FC for a Broadcom/Emulex adapter.

Steps

1. Verify that you are using the supported adapter model:

   a. `cat /sys/class/scsi_host/host*/modelname`

   ```
   LPe36002-M64
   LPe36002-M64
   ```

   b. `cat /sys/class/scsi_host/host*/modeldesc`

   ```
   Emulex LPe36002-M64 2-Port 64Gb Fibre Channel Adapter
   Emulex LPe36002-M64 2-Port 64Gb Fibre Channel Adapter
   ```

2. Verify that you are using the recommended Broadcom `lpfc` firmware and inbox driver:

   a. `cat /sys/class/scsi_host/host*/fwrev`

   ```
   14.4.317.10, sli-4:6:d
   14.4.317.10, sli-4:6:d
   ```

   b. `cat /sys/module/lpfc/version`

   ```
   0:14.2.0.13
   ```

   For the current list of supported adapter driver and firmware versions, see the Interoperability Matrix Tool.

3. Verify that `lpfc_enable_fc4_type` is set to "3":

   `cat /sys/module/lpfc/parameters/lpfc_enable_fc4_type`

4. Verify that the initiator ports are up and running and that you can see the target LIFs:

   a. `cat /sys/class/fc_host/host*/port_name`

   ```
   0x100000109bf0449c
   0x100000109bf0449d
   ```

   b. `cat /sys/class/fc_host/host*/port_state`

```
Online
Online
```

c. `cat /sys/class/scsi_host/host*/nvme_info`

Show example

```
NVME Initiator Enabled
XRI Dist lpfc0 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc0 WWPN x100000109bf0449c WWNN x200000109bf0449c
DID x061500 ONLINE
NVME RPORT       WWPN x200bd039eab31e9c WWNN x2005d039eab31e9c
DID x020e06 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2006d039eab31e9c WWNN x2005d039eab31e9c
DID x020a0a TARGET DISCSRVC ONLINE
NVME Statistics
LS: Xmt 000000002c Cmpl 000000002c Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 00000000008ffe8 Issue 00000000008ffb9 OutIO
fffffffffffffd1
        abort 0000000c noxri 00000000 nondlp 00000000 qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 0000000c Err 0000000c
NVME Initiator Enabled
XRI Dist lpfc1 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc1 WWPN x100000109bf0449d WWNN x200000109bf0449d
DID x062d00 ONLINE
NVME RPORT       WWPN x201fd039eab31e9c WWNN x2005d039eab31e9c
DID x02090a TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x200cd039eab31e9c WWNN x2005d039eab31e9c
DID x020d06 TARGET DISCSRVC ONLINE
NVME Statistics
LS: Xmt 0000000041 Cmpl 0000000041 Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 00000000000936bf Issue 000000000009369a OutIO
ffffffffffffffdb
        abort 00000016 noxri 00000000 nondlp 00000000 qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 00000016 Err 00000016
```

Marvell/QLogic

Configure NVMe/FC for a Marvell/QLogic adapter.

The native inbox qla2xxx driver included in the Oracle Linux 10 GA kernel has the latest fixes. These fixes are essential for ONTAP support.

Steps

1. Verify that you are running the supported adapter driver and firmware versions:

   `cat /sys/class/fc_host/host*/symbolic_name`

   ```
   QLE2772 FW:v9.15.00 DVR:v10.02.09.100-k
   QLE2772 FW:v9.15.00 DVR:v10.02.09.100-k
   ```

2. Verify that `ql2xnvmeenable` is set to "1". This enables the Marvell adapter to function as an NVMe/FC initiator:

   `cat /sys/module/qla2xxx/parameters/ql2xnvmeenable`

Enable 1MB I/O size (Optional)

ONTAP reports an MDTS (Max Data Transfer Size) of 8 in the Identify Controller data. This means the maximum I/O request size can be up to 1MB. To issue I/O requests of size 1MB for a Broadcom NVMe/FC host, you should increase the `lpfc` value of the `lpfc_sg_seg_cnt` parameter to 256 from the default value of 64.

These steps don't apply to Qlogic NVMe/FC hosts.

Steps

1. Set the `lpfc_sg_seg_cnt` parameter to 256:

   ```
   cat /etc/modprobe.d/lpfc.conf
   ```

   You should see an output similar to the following example:

   ```
   options lpfc lpfc_sg_seg_cnt=256
   ```

2. Run the `dracut -f` command, and reboot the host.
3. Verify that the value for `lpfc_sg_seg_cnt` is 256:

   ```
   cat /sys/module/lpfc/parameters/lpfc_sg_seg_cnt
   ```

Configure NVMe/TCP

The NVMe/TCP protocol doesn't support the `auto-connect` operation. Instead, you can discover the

NVMe/TCP subsystems and namespaces by performing the NVMe/TCP `connect` or `connect-all` operations manually.

Steps

1. Verify that the initiator port can fetch the discovery log page data across the supported NVMe/TCP LIFs:

```
nvme discover -t tcp -w <host-traddr> -a <traddr>
```

Show example

```
#   nvme discover -t tcp -w 192.168.6.1 -a 192.168.6.24 Discovery
Log Number of Records 20, Generation counter 45
=====Discovery Log Entry 0======
trtype:  tcp
adrfam:  ipv4
subtype: unrecognized
treq:    not specified
portid:  6
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.e6c438e66ac211ef9ab8d039eab31e9d:discovery
traddr:  192.168.6.25
sectype: none
=====Discovery Log Entry 1======
trtype:  tcp
adrfam:  ipv4
subtype: unrecognized
treq:    not specified
portid:  1
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.e6c438e66ac211ef9ab8d039eab31e9d:discovery
traddr:  192.168.5.24
sectype: none
=====Discovery Log Entry 2======
trtype:  tcp
adrfam:  ipv4
subtype: unrecognized
treq:    not specified
portid:  4
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.e6c438e66ac211ef9ab8d039eab31e9d:discovery
traddr:  192.168.6.24
sectype: none
=====Discovery Log Entry 3======
trtype:  tcp
adrfam:  ipv4
subtype: unrecognized
treq:    not specified
portid:  2
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.e6c438e66ac211ef9ab8d039eab31e9d:discovery
```

```
traddr:  192.168.5.25
sectype: none
=====Discovery Log Entry 4======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  6
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.e6c438e66ac211ef9ab8d039eab31e9d:subsystem.nvme_tcp
_4
traddr:  192.168.6.25
sectype: none
=====Discovery Log Entry 5======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.e6c438e66ac211ef9ab8d039eab31e9d:subsystem.nvme_tcp
_4
..........
```

2. Verify that all other NVMe/TCP initiator-target LIF combinations can successfully fetch discovery log page data:

```
nvme discover -t tcp -w <host-traddr> -a <traddr>
```

Show example

```
# nvme discover -t tcp -w 192.168.6.1 -a 192.168.6.24
# nvme discover -t tcp -w 192.168.6.1 -a 192.168.6.25
# nvme discover -t tcp -w 192.168.5.1 -a 192.168.5.24
# nvme discover -t tcp -w 192.168.5.1 -a 192.168.5.25
```

3. Run the `nvme connect-all` command across all supported NVMe/TCP initiator-target LIFs across the nodes:

```
nvme connect-all -t tcp -w <host-traddr> -a <traddr> -l
<ctrl_loss_timeout_in_seconds>
```

Show example

```
#   nvme    connect-all -t  tcp -w  192.168.5.1 -a  192.168.5.24
-l -1
#   nvme    connect-all -t  tcp -w  192.168.5.1 -a  192.168.5.25
-l -1
#   nvme    connect-all -t  tcp -w  192.168.6.1 -a  192.168.6.24
-l -1
#   nvme    connect-all -t  tcp -w  192.168.6.1 -a  192.168.6.25
-l -1
```

NetApp recommends setting the `ctrl-loss-tmo` option to "-1" so that the NVMe/TCP initiator attempts to reconnect indefinitely in the event of a path loss.

Validate NVMe-oF

Verify that the in-kernel NVMe multipath status, ANA status, and ONTAP namespaces are correct for the NVMe-oF configuration.

Steps

1. Verify that the in-kernel NVMe multipath is enabled:

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

   You should see the following output:

   ```
   Y
   ```

2. Verify that the appropriate NVMe-oF settings (such as, model set to NetApp ONTAP Controller and load balancing iopolicy set to round-robin) for the respective ONTAP namespaces correctly reflect on the host:

   a. Display the subsystems:

      ```
      cat /sys/class/nvme-subsystem/nvme-subsys*/model
      ```

      You should see the following output:

```
NetApp ONTAP Controller
NetApp ONTAP Controller
```

b. Display the policy:

```
cat /sys/class/nvme-subsystem/nvme-subsys*/iopolicy
```

You should see the following output:

```
round-robin
round-robin
```

3. Verify that the namespaces are created and correctly discovered on the host:

```
nvme list
```

Show example

```
Node         SN                   Model
---------------------------------------------------------
/dev/nvme4n1 81Ix2BVuekWcAAAAAAB   NetApp ONTAP Controller


Namespace Usage    Format             FW             Rev
-----------------------------------------------------------
1                 21.47 GB / 21.47 GB   4 KiB + 0 B   FFFFFFFF
```

Steps

1. Verify that in-kernel NVMe multipath is enabled:

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

```
Y
```

2. Verify that the NVMe-oF settings (such as model set to "NetApp ONTAP Controller" and load balancing iopolicy set to "round-robin") for the respective ONTAP namespaces correctly display on the host:
   a. `cat /sys/class/nvme-subsystem/nvme-subsys*/model`

```
NetApp ONTAP Controller
NetApp ONTAP Controller
```

b. `cat /sys/class/nvme-subsystem/nvme-subsys*/iopolicy`

```
round-robin
round-robin
```

3. Verify that the namespaces are created and correctly discovered on the host:

   `nvme list`

   Show example

```
Node         SN                   Model
---------------------------------------------------------
/dev/nvme0n1 814vWBNRwf9HAAAAAAAB NetApp ONTAP Controller
/dev/nvme0n2 814vWBNRwf9HAAAAAAAB NetApp ONTAP Controller
/dev/nvme0n3 814vWBNRwf9HAAAAAAAB NetApp ONTAP Controller

Namespace Usage   Format               FW            Rev
-----------------------------------------------------------
1                 85.90 GB / 85.90 GB  4 KiB + 0 B   FFFFFFFF
2                 85.90 GB / 85.90 GB  24 KiB + 0 B  FFFFFFFF
3                   85.90 GB / 85.90 GB  4 KiB + 0 B   FFFFFFFF
```

4. Verify that the controller state of each path is live and has the correct ANA status:

NVMe/FC

```
nvme list-subsys /dev/nvme0n1
```

Show example

```
nvme-subsys0 - NQN=nqn.1992- 08.com.netapp:
4b4d82566aab11ef9ab8d039eab31e9d:subsystem.nvme\
+-  nvme1 fc traddr=nn-0x2038d039eab31e9c:pn-0x203ad039eab31e9c
host_traddr=nn-0x200034800d756a89:pn-0x210034800d756a89 live
optimized
+-  nvme2 fc traddr=nn-0x2038d039eab31e9c:pn-0x203cd039eab31e9c
host_traddr=nn-0x200034800d756a88:pn-0x210034800d756a88 live
optimized
+- nvme3 fc traddr=nn-0x2038d039eab31e9c:pn-0x203ed039eab31e9c
host_traddr=nn-0x200034800d756a89:pn-0x210034800d756a89 live
non-optimized
+-  nvme7 fc traddr=nn-0x2038d039eab31e9c:pn-0x2039d039eab31e9c
host_traddr=nn-0x200034800d756a88:pn-0x210034800d756a88 live
non-optimized
```

NVMe/TCP

```
nvme list-subsys /dev/nvme0n1
```

Show example

```
nvme-subsys0 - NQN=nqn.1992- 08.com.netapp:
sn.e6c438e66ac211ef9ab8d039eab31e9d:subsystem.nvme_tcp_4
\
+- nvme1 tcp traddr=192.168.5.25 trsvcid=4420
host_traddr=192.168.5.1 src_addr=192.168.5.1 live optimized
+- nvme10 tcp traddr=192.168.6.24 trsvcid=4420
host_traddr=192.168.6.1 src_addr=192.168.6.1 live optimized
+- nvme2 tcp traddr=192.168.5.24 trsvcid=4420
host_traddr=192.168.5.1 src_addr=192.168.5.1 live non-optimized
+- nvme9 tcp traddr=192.168.6.25 trsvcid=4420
host_traddr=192.168.6.1 src_addr=192.168.6.1 live non-optimized
```

5. Verify that the NetApp plug-in displays the correct values for each ONTAP namespace device:

Column

```
nvme netapp ontapdevices -o column
```

Show example

```
Device          Vserver                   Namespace Path
NSID UUID                                   Size
-------------- ------------------------
----------------------------- ----
------------------------------------- ---------
/dev/nvme0n1   vs_coexistence_QLE2772
/vol/fcnvme_1_1_0/fcnvme_ns   1    159f9f88-be00-4828-aef6-
197d289d4bd9  10.74GB
/dev/nvme0n2   vs_coexistence_QLE2772
/vol/fcnvme_1_1_1/fcnvme_ns   2    2c1ef769-10c0-497d-86d7-
e84811ed2df6  10.74GB
/dev/nvme0n3   vs_coexistence_QLE2772
/vol/fcnvme_1_1_2/fcnvme_ns   3    9b49bf1a-8a08-4fa8-baf0-
6ec6332ad5a4  10.74GB
```

JSON

```
nvme netapp ontapdevices -o json
```

Show example


```
{
  "ONTAPdevices" : [
    {
      "Device" : "/dev/nvme0n1",
      "Vserver" : "vs_coexistence_QLE2772",
      "Namespace_Path" : "/vol/fcnvme_1_1_0/fcnvme_ns",
      "NSID" : 1,
      "UUID" : "159f9f88-be00-4828-aef6-197d289d4bd9",
      "Size" : "10.74GB",
      "LBA_Data_Size" : 4096,
      "Namespace_Size" : 2621440
    },
    {
      "Device" : "/dev/nvme0n2",
      "Vserver" : "vs_coexistence_QLE2772",
      "Namespace_Path" : "/vol/fcnvme_1_1_1/fcnvme_ns",
      "NSID" : 2,
      "UUID" : "2c1ef769-10c0-497d-86d7-e84811ed2df6",
      "Size" : "10.74GB",
      "LBA_Data_Size" : 4096,
      "Namespace_Size" : 2621440
    },
    {
      "Device" : "/dev/nvme0n4",
      "Vserver" : "vs_coexistence_QLE2772",
      "Namespace_Path" : "/vol/fcnvme_1_1_3/fcnvme_ns",
      "NSID" : 4,
      "UUID" : "f3572189-2968-41bc-972a-9ee442dfaed7",
      "Size" : "10.74GB",
      "LBA_Data_Size" : 4096,
      "Namespace_Size" : 2621440
    },
```


Known issues

The NVMe-oF host configuration for Oracle Linux 8.10 with ONTAP release has the following known issue:

NetApp Bug ID	Title	Description
CONTAPEXT-1082	Oracle Linux 8.10 NVMe-oF hosts create duplicate PDCs	On Oracle Linux 8.10 NVMe-oF hosts, Persistent Discovery Controllers (PDCs) are created by using the `-p` option with the `nvme discover` command. For a given initiator-target combination, each execution of the `nvme discover` command is expected to create one PDC. However, beginning with Oracle Linux 8.x, NVMe-oF hosts create a duplicate PDC. This wastes resources on both the host and the target.

NVMe-oF Host Configuration for Oracle Linux 8.9 with ONTAP

NVMe over Fabrics (NVMe-oF), including NVMe over Fibre Channel (NVMe/FC) and other transports, is supported with Oracle Linux 8.9 with Asymmetric Namespace Access (ANA). In NVMe-oF environments, ANA is the equivalent of ALUA multipathing in iSCSI and FC environments and is implemented with in-kernel NVMe multipath.

The following support is available for the NVMe-oF host configuration for Oracle Linux 8.9 with ONTAP:

- Support for NVMe over TCP (NVMe/TCP) in addition to NVMe/FC. The NetApp plug-in in the native `nvme-cli` package displays the ONTAP details for both NVMe/FC and NVMe/TCP namespaces.
- Both NVMe and SCSI traffic can be run on the same host. Therefore, you can configure dm-multipath for SCSI mpath devices for SCSI LUNs, whereas you might use NVMe multipath to configure NVMe-oF namespace devices on the host.
- There is no sanlun support for NVMe-oF. Therefore, there is no host utility support for NVMe-oF on an Oracle Linux 8.9 host. You can rely on the NetApp plug-in included in the native `nvme-cli` package for all NVMe-oF transports.

For additional details on supported configurations, see the Interoperability Matrix Tool.

Features

Oracle Linux 8.9 has in-kernel NVMe multipath enabled for NVMe namespaces by default; therefore, there is no need for explicit settings.

Known limitations

SAN booting using the NVMe-oF protocol is currently not supported.

Validate software versions

Validate the minimum supported Oracle Linux 8.9 software versions.

Steps

1. Install Oracle Linux 8.9 GA on the server. After the installation is complete, verify that you are running the specified Oracle Linux 8.9 GA kernel:

```
# uname -r
```

Example output:

```
5.15.0-200.131.27.el8uek.x86_64
```

2. Install the `nvme-cli` package:

```
# rpm -qa|grep nvme-cli
```

Example output:

```
nvme-cli-1.16-9.el8.x86_64
```

3. On the Oracle Linux 8.9 host, check the `hostnqn` string at `/etc/nvme/hostnqn`:

```
# cat /etc/nvme/hostnqn
```

Example output:

```
nqn.2014-08.org.nvmexpress:uuid:edd38060-00f7-47aa-a9dc-4d8ae0cd969a
```

4. Verify that the `hostnqn` string matches the `hostnqn` string for the corresponding subsystem on the ONTAP array:

```
::> vserver nvme subsystem host show -vserver vs_nvme177
```

Example output:

```
Vserver     Subsystem          Host NQN
----------- ---------------
--------------------------------------------------------------
vs_nvme177  nvme_ss_ol_1       nqn.2014-08.org.nvmexpress:uuid:edd38060-
00f7-47aa-a9dc-4d8ae0cd969a
```

If the `hostnqn` strings do not match, you can use the `vserver modify` command to update the `hostnqn` string on your corresponding ONTAP array subsystem to match the `hostnqn` string from `/etc/nvme/hostnqn` on the host.

5. Reboot the host.

> If you intend to run both NVMe and SCSI co-existent traffic on the same host, NetApp recommends using the in-kernel NVMe multipath for ONTAP namespaces and `dm-multipath` for ONTAP LUNs respectively. This means that the ONTAP namespaces should be excluded from `dm-multipath` to prevent `dm-multipath` from claiming these namespace devices. You can add the `enable_foreign` setting to the `/etc/multipath.conf` file:
>
> ```
> # cat /etc/multipath.conf
>
> defaults {
>   enable_foreign     NONE
> }
> ```
>
> Restart the multipathd daemon by running a `systemctl restart multipathd` command. This allows the new setting to take effect.

Configure NVMe/FC

Configure NVMe/FC for Broadcom/Emulex adapters or Marvell/Qlogic adapters.

Broadcom/Emulex

Steps

1. Verify that you are using the supported adapter model:

   ```
   # cat /sys/class/scsi_host/host*/modelname
   ```

   Example output:

   ```
   LPe32002-M2
   LPe32002-M2
   ```

   ```
   # cat /sys/class/scsi_host/host*/modeldesc
   ```

   Example output:

   ```
   Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
   Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
   ```

2. Verify that you are using the recommended Broadcom `lpfc` firmware and inbox driver:

   ```
   # cat /sys/class/scsi_host/host*/fwrev
   14.2.539.16, sli-4:2:c
   14.2.539.16, sli-4:2:c
   ```

   ```
   # cat /sys/module/lpfc/version
   0:14.2.0.5
   ```

   For the current list of supported adapter driver and firmware versions, see the Interoperability Matrix Tool.

3. Verify that `lpfc_enable_fc4_type` is set to `3`:

   ```
   # cat /sys/module/lpfc/parameters/lpfc_enable_fc4_type
   3
   ```

4. Verify that the initiator ports are up and running, and that you can see the target LIFs:

```
# cat /sys/class/fc_host/host*/port_name
0x100000109b3c081f
0x100000109b3c0820
```

```
# cat /sys/class/fc_host/host*/port_state
Online
Online
```

Show example

```
# cat /sys/class/scsi_host/host*/nvme_info
NVME Initiator Enabled
XRI Dist lpfc0 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc0 WWPN x100000109b1c1204 WWNN x200000109b1c1204
DID x011d00 ONLINE
NVME RPORT WWPN x203800a098dfdd91 WWNN x203700a098dfdd91 DID
x010c07 TARGET DISCSRVC ONLINE
NVME RPORT WWPN x203900a098dfdd91 WWNN x203700a098dfdd91 DID
x011507 TARGET DISCSRVC ONLINE
NVME Statistics
LS: Xmt 0000000f78 Cmpl 0000000f78 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000002fe29bba Issue 000000002fe29bc4 OutIO
000000000000000a
abort 00001bc7 noxri 00000000 nondlp 00000000 qdepth 00000000
wqerr 00000000 err 00000000
FCP CMPL: xb 00001e15 Err 0000d906
NVME Initiator Enabled
XRI Dist lpfc1 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc1 WWPN x100000109b1c1205 WWNN x200000109b1c1205
DID x011900 ONLINE
NVME RPORT WWPN x203d00a098dfdd91 WWNN x203700a098dfdd91 DID
x010007 TARGET DISCSRVC ONLINE
NVME RPORT WWPN x203a00a098dfdd91 WWNN x203700a098dfdd91 DID
x012a07 TARGET DISCSRVC ONLINE
NVME Statistics
LS: Xmt 0000000fa8 Cmpl 0000000fa8 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000002e14f170 Issue 000000002e14f17a OutIO
000000000000000a
abort 000016bb noxri 00000000 nondlp 00000000 qdepth 00000000
wqerr 00000000 err 00000000
FCP CMPL: xb 00001f50 Err 0000d9f8
```

Marvell/QLogic FC Adapter for NVMe/FC

The native inbox qla2xxx driver included in the Oracle Linux 8.9 GA kernel has the latest fixes. These fixes are essential for ONTAP support.

Steps

1. Verify that you are running the supported adapter driver and firmware versions:

```
# cat /sys/class/fc_host/host*/symbolic_name
QLE2742 FW:v9.12.00 DVR:v10.02.08.100-k
QLE2742 FW:v9.12.00 DVR:v10.02.08.100-k
```

2. Verify that `ql2xnvmeenable` is set. This enables the Marvell adapter to function as an NVMe/FC initiator:

```
# cat /sys/module/qla2xxx/parameters/ql2xnvmeenable
1
```

Enable 1MB I/O size (Optional)

ONTAP reports an MDTS (Max Data Transfer Size) of 8 in the Identify Controller data. This means the maximum I/O request size can be up to 1MB. To issue I/O requests of size 1MB for a Broadcom NVMe/FC host, you should increase the `lpfc` value of the `lpfc_sg_seg_cnt` parameter to 256 from the default value of 64.

> These steps don't apply to Qlogic NVMe/FC hosts.

Steps

1. Set the `lpfc_sg_seg_cnt` parameter to 256:

```
cat /etc/modprobe.d/lpfc.conf
```

You should see an output similar to the following example:

```
options lpfc lpfc_sg_seg_cnt=256
```

2. Run the `dracut -f` command, and reboot the host.
3. Verify that the value for `lpfc_sg_seg_cnt` is 256:

```
cat /sys/module/lpfc/parameters/lpfc_sg_seg_cnt
```

Configure NVMe/TCP

NVMe/TCP does not have auto-connect functionality. Therefore, if a path goes down and is not reinstated within the default time out period of 10 minutes, NVMe/TCP cannot automatically reconnect. To avoid this scenario, you should set the retry period for storage failover events by using the following procedure.

Steps

1. Verify that the initiator port can fetch the discovery log page data across the supported NVMe/TCP LIFs:

```
nvme discover -t tcp -w host-traddr -a traddr
```

Show example output

```
#  nvme discover -t tcp -w 192.168.6.13 -a 192.168.6.15
Discovery Log Number of Records 6, Generation counter 8
=====Discovery Log Entry 0======
trtype: tcp
adrfam: ipv4
subtype: unrecognized
treq: not specified
portid: 0
trsvcid: 8009
subnqn: nqn.1992-
08.com.netapp:sn.1c6ac66338e711eda41dd039ea3ad566:discovery
traddr: 192.168.6.17
sectype: none
=====Discovery Log Entry 1======
trtype: tcp
adrfam: ipv4
subtype: unrecognized
treq: not specified
portid: 1
trsvcid: 8009
subnqn: nqn.1992-
08.com.netapp:sn.1c6ac66338e711eda41dd039ea3ad566:discovery
traddr: 192.168.5.17
sectype: none
=====Discovery Log Entry 2======
trtype: tcp
adrfam: ipv4
subtype: unrecognized
treq: not specified
portid: 2
trsvcid: 8009
subnqn: nqn.1992-
08.com.netapp:sn.1c6ac66338e711eda41dd039ea3ad566:discovery
traddr: 192.168.6.15
sectype: none
=====Discovery Log Entry 3======
trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 0
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.1c6ac66338e711eda41dd039ea3ad566:subsystem.host_95
```

```
traddr: 192.168.6.17
sectype: none
..........
```

2. Verify that the other NVMe/TCP initiator-target LIF combinations can successfully fetch discovery log page data:

```
nvme discover -t tcp -w host-traddr -a traddr
```

Example output:

```
# nvme discover -t tcp -w 192.168.6.1 -a 192.168.6.10
# nvme discover -t tcp -w 192.168.6.1 -a 192.168.6.11
# nvme discover -t tcp -w 192.168.5.1 -a 192.168.5.10
# nvme discover -t tcp -w 192.168.5.1 -a 192.168.5.11
```

3. Run the `nvme connect-all` command across all the supported NVMe/TCP initiator-target LIFs across the nodes:

```
nvme connect-all -t tcp -w host-traddr -a traddr -l
<ctrl_loss_timeout_in_seconds>
```

Example output:

```
#   nvme    connect-all -t  tcp -w  192.168.5.1 -a  192.168.5.10    -l
-1
#   nvme    connect-all -t  tcp -w  192.168.5.1 -a  192.168.5.11    -l
-1
#   nvme    connect-all -t  tcp -w  192.168.6.1 -a  192.168.6.10    -l
-1
#   nvme    connect-all -t  tcp -w  192.168.6.1 -a  192.168.6.11    -l
-1
```

NetApp recommends setting the `ctrl-loss-tmo` option to `-1` so that the NVMe/TCP initiator attempts to reconnect indefinitely in the event of a path loss.

Validate NVMe-oF

You can use the following procedure to validate NVMe-oF.

Steps

1. Verify that in-kernel NVMe multipath is enabled:

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

2. Verify that the appropriate NVMe-oF settings (such as `model` set to `NetApp ONTAP Controller` and load balancing `iopolicy` set to `round-robin`) for the respective ONTAP namespaces correctly reflect on the host:

```
# cat /sys/class/nvme-subsystem/nvme-subsys*/model
NetApp ONTAP Controller
NetApp ONTAP Controller
```

```
# cat /sys/class/nvme-subsystem/nvme-subsys*/iopolicy
round-robin
round-robin
```

3. Verify that the namespaces are created and correctly discovered on the host:

```
# nvme list
```

Example output:

```
Node         SN                   Model
---------------------------------------------------------
/dev/nvme0n1 814vWBNRwf9HAAAAAAAB NetApp ONTAP Controller
/dev/nvme0n2 814vWBNRwf9HAAAAAAAB NetApp ONTAP Controller
/dev/nvme0n3 814vWBNRwf9HAAAAAAAB NetApp ONTAP Controller



Namespace Usage    Format             FW             Rev
-----------------------------------------------------------
1                 85.90 GB / 85.90 GB  4 KiB + 0 B   FFFFFFFF
2                 85.90 GB / 85.90 GB  24 KiB + 0 B  FFFFFFFF
3                   85.90 GB / 85.90 GB  4 KiB + 0 B   FFFFFFFF
```

4. Verify that the controller state of each path is live and has the correct ANA status:

NVMe/FC

```
# nvme list-subsys /dev/nvme0n1
```

Example output:

```
nvme-subsys0 - NQN=nqn.1992-
08.com.netapp:sn.5f5f2c4aa73b11e9967e00a098df41bd:subsystem.nvme_ss_
ol_1
\
+- nvme0 fc traddr=nn-0x203700a098dfdd91:pn-0x203800a098dfdd91
host_traddr=nn-0x200000109b1c1204:pn-0x100000109b1c1204 live non-
optimized
+- nvme1 fc traddr=nn-0x203700a098dfdd91:pn-0x203900a098dfdd91
host_traddr=nn-0x200000109b1c1204:pn-0x100000109b1c1204 live non-
optimized
+- nvme2 fc traddr=nn-0x203700a098dfdd91:pn-0x203a00a098dfdd91
host_traddr=nn-0x200000109b1c1205:pn-0x100000109b1c1205 live
optimized
+- nvme3 fc traddr=nn-0x203700a098dfdd91:pn-0x203d00a098dfdd91
host_traddr=nn-0x200000109b1c1205:pn-0x100000109b1c1205 live
optimized
```

NVMe/TCP

```
nvme list-subsys /dev/nvme1n22
```

Example output

```
nvme-subsys0 - NQN=nqn.1992- 08.com.netapp:
sn.44986b09cadc11eeb309d039eab31e9d:subsystem.ol_nvme
\
+- nvme1 tcp traddr=192.168.5.11 trsvcid=4420
host_traddr=192.168.5.1 src_addr=192.168.5.1 live non-optimized
+- nvme2 tcp traddr=192.168.5.10 trsvcid=4420
host_traddr=192.168.5.1 src_addr=192.168.5.1 live optimized
+- nvme3 tcp traddr=192.168.6.11 trsvcid=4420
host_traddr=192.168.6.1 src_addr=192.168.6.1 live non-optimized
+- nvme4 tcp traddr=192.168.6.10 trsvcid=4420
host_traddr=192.168.6.1 src_addr=192.168.6.1 live  optimized
```

5. Verify that the NetApp plug-in displays the correct values for each ONTAP namespace device:

Column

```
# nvme netapp ontapdevices -o column
```

Example output:

```
Device        Vserver     Namespace Path
------------  ----------  -----------------
/dev/nvme0n1    vs_nvme177  /vol/vol1/ns1
/dev/nvme0n2    vs_nvme177  /vol/vol2/ns2
/dev/nvme0n3    vs_nvme177  /vol/vol3/ns3



NSID     UUID                                   Size
-------- -------------------------------------- -----------
1          72b887b1-5fb6-47b8-be0b-33326e2542e2   85.90GB
2          04bf9f6e-9031-40ea-99c7-a1a61b2d7d08   85.90GB
3          264823b1-8e03-4155-80dd-e904237014a4   85.90GB
```

JSON

```
# nvme netapp ontapdevices -o json
```

Example output

```
{
"ONTAPdevices" : [
{
"Device" : "/dev/nvme0n1", "Vserver" : "vs_nvme177",
"Namespace_Path" : "/vol/vol1/ns1",
"NSID" : 1,
"UUID" : "72b887b1-5fb6-47b8-be0b-33326e2542e2", "Size" : "85.90GB",
"LBA_Data_Size" : 4096,
"Namespace_Size" : 5242880
},
{
"Device" : "/dev/nvme0n2", "Vserver" : "vs_nvme177",
"Namespace_Path" : "/vol/vol2/ns2",
 "NSID" : 2,
"UUID" : "04bf9f6e-9031-40ea-99c7-a1a61b2d7d08", "Size" : "85.90GB",
"LBA_Data_Size" : 4096,
"Namespace_Size" : 20971520
},
{
"Device" : "/dev/nvme0n3", "Vserver" : "vs_nvme177",
"Namespace_Path" : "/vol/vol3/ns3",
 "NSID" : 3,
"UUID" : "264823b1-8e03-4155-80dd-e904237014a4", "Size" : "85.90GB",
"LBA_Data_Size" : 4096,
"Namespace_Size" : 20971520
},
]
}
```

Known issues

The NVMe-oF host configuration for Oracle Linux 8.9 with ONTAP release has the following known issues:

NetApp Bug ID	Title	Description

1517321	Oracle Linux 8.9 NVMe-oF hosts create duplicate PDCs	On Oracle Linux 8.9 NVMe-oF hosts, Persistent Discovery Controllers (PDCs) are created by passing the `-p` option to the `nvme discover` command. For a given initiator-target combination, each execution of the `nvme discover` command is expected to create one PDC. However, beginning with Oracle Linux 8.x, NVMe-oF hosts create duplicate. This wastes resources on both the host and the target.

NVMe-oF Host Configuration for Oracle Linux 8.8 with ONTAP

NVMe over Fabrics (NVMe-oF), including NVMe over Fibre Channel (NVMe/FC) and other transports, is supported with Oracle Linux (OL) 8.8 with Asymmetric Namespace Access (ANA). In NVMe-oF environments, ANA is the equivalent of ALUA multipathing in iSCSI and FC environments and is implemented with in-kernel NVMe multipath.

The following support is available for the NVMe-oF host configuration for OL 8.8 with ONTAP:

- Support for NVMe over TCP (NVMe/TCP) in addition to NVMe/FC. The NetApp plug-in in the native `nvme-cli` package displays ONTAP details for both NVMe/FC and NVMe/TCP namespaces.
- Both NVMe and SCSI traffic can be run on the same host. Therefore, for SCSI LUNs, you can configure dm-multipath for SCSI mpath devices, whereas you might use NVMe multipath to configure NVMe-oF namespace devices on the host.
- There is no sanlun support for NVMe-oF. Therefore, there is no host utility support for NVMe-oF on an OL 8.8 host. You can rely on the NetApp plug-in included in the native `nvme-cli` package for all NVMe-oF transports.

For additional details on supported configurations, see the Interoperability Matrix Tool.

Features

Oracle Linux 8.8 has in-kernel NVMe multipath enabled for NVMe namespaces by default, therefore, there is no need for explicit settings.

Known limitations

SAN booting using the NVMe-oF protocol is currently not supported.

Validate software versions

You can use the following procedure to validate the minimum supported OL 8.8 software versions.

Steps

1. Install OL 8.8 GA on the server. After the installation is complete, verify that you are running the specified OL 8.8 GA kernel.

```
# uname -r
```

Example output:

```
5.15.0-101.103.2.1.el8uek.x86_64
```

2. Install the `nvme-cli` package:

```
# rpm -qa|grep nvme-cli
```

Example output:

```
nvme-cli-1.16-7.el8.x86_64
```

3. On the Oracle Linux 8.8 host, check the `hostnqn` string at `/etc/nvme/hostnqn`:

```
# cat /etc/nvme/hostnqn
```

Example output:

```
nqn.2014-08.org.nvmexpress:uuid:bc59d14c-47f3-11eb-b93c-3a68dd48673f
```

4. Verify that the `hostnqn` string matches the `hostnqn` string for the corresponding subsystem on the ONTAP array:

```
::> vserver nvme subsystem host show -vserver vs_ol_nvme
```

Example output:

```
Vserver     Subsystem          Host NQN
----------- ---------------
----------------------------------------------------------
vs_nvme207   nvme_ss_ol_1    nqn.2014-08.org.nvmexpress:uuid:bc59d14c-
47f3-11eb-b93c-3a68dd48673f
```

If the `hostnqn` strings do not match, you can use the `vserver modify` command to update the `hostnqn` string on your corresponding ONTAP array subsystem to match the `hostnqn` string from `/etc/nvme/hostnqn` on the host.

5. Reboot the host.

If you intend to run both NVMe and SCSI co-existent traffic on the same host, NetApp recommends using the in-kernel NVMe multipath for ONTAP namespaces and `dm-multipath` for ONTAP LUNs respectively. This means that the ONTAP namespaces should be excluded from `dm-multipath` to prevent `dm-multipath` from claiming these namespace devices. You can add the `enable_foreign` setting to the `/etc/multipath.conf` file:

```
# cat /etc/multipath.conf

defaults {
  enable_foreign     NONE
}
```

Restart the multipathd daemon by running a `systemctl restart multipathd` command. This allows the new setting to take effect.

Configure NVMe/FC

You can configure NVMe/FC for Broadcom/Emulex adapters or Marvell/Qlogic adapters.

Broadcom/Emulex

Steps

1. Verify that you are using the supported adapter model:

   ```
   # cat /sys/class/scsi_host/host*/modelname
   ```

   Example output:

   ```
   LPe32002-M2
   LPe32002-M2
   ```

   ```
   # cat /sys/class/scsi_host/host*/modeldesc
   ```

   Example output:

   ```
   Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
   Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
   ```

2. Verify that you are using the recommended Broadcom `lpfc` firmware and inbox driver:

   ```
   # cat /sys/class/scsi_host/host*/fwrev
   14.2.539.16, sli-4:2:c
   14.2.539.16, sli-4:2:c
   ```

   ```
   # cat /sys/module/lpfc/version
   0:14.2.0.5
   ```

   For the current list of supported adapter driver and firmware versions, see the Interoperability Matrix Tool.

3. Verify that `lpfc_enable_fc4_type` is set to `3`:

   ```
   # cat /sys/module/lpfc/parameters/lpfc_enable_fc4_type
   3
   ```

4. Verify that the initiator ports are up and running, and that you can see the target LIFs:

```
# cat /sys/class/fc_host/host*/port_name
0x100000109b3c081f
0x100000109b3c0820
```

```
# cat /sys/class/fc_host/host*/port_state
Online
Online
```

```
# cat /sys/class/scsi_host/host*/nvme_info
NVME Initiator Enabled
XRI Dist lpfc0 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc0 WWPN x100000109b1c1204 WWNN x200000109b1c1204 DID
x011d00 ONLINE
NVME RPORT WWPN x203800a098dfdd91 WWNN x203700a098dfdd91 DID x010c07
TARGET DISCSRVC ONLINE
NVME RPORT WWPN x203900a098dfdd91 WWNN x203700a098dfdd91 DID x011507
TARGET DISCSRVC ONLINE
NVME Statistics
LS: Xmt 0000000f78 Cmpl 0000000f78 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000002fe29bba Issue 000000002fe29bc4 OutIO
000000000000000a
abort 00001bc7 noxri 00000000 nondlp 00000000 qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 00001e15 Err 0000d906
NVME Initiator Enabled
XRI Dist lpfc1 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc1 WWPN x100000109b1c1205 WWNN x200000109b1c1205 DID
x011900 ONLINE
NVME RPORT WWPN x203d00a098dfdd91 WWNN x203700a098dfdd91 DID x010007
TARGET DISCSRVC ONLINE
NVME RPORT WWPN x203a00a098dfdd91 WWNN x203700a098dfdd91 DID x012a07
TARGET DISCSRVC ONLINE
NVME Statistics
LS: Xmt 0000000fa8 Cmpl 0000000fa8 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000002e14f170 Issue 000000002e14f17a OutIO
000000000000000a
abort 000016bb noxri 00000000 nondlp 00000000 qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 00001f50 Err 0000d9f8
```

Marvell/QLogic FC Adapter for NVMe/FC

The native inbox qla2xxx driver included in the OL 8.8 GA kernel has the latest fixes. These fixes are essential for ONTAP support.

Steps

1. Verify that you are running the supported adapter driver and firmware versions:

```
# cat /sys/class/fc_host/host*/symbolic_name
QLE2742 FW:v9.12.00 DVR:v10.02.08.100-k
QLE2742 FW:v9.12.00 DVR:v10.02.08.100-k
```

2. Verify that `ql2xnvmeenable` is set. This enables the Marvell adapter to function as an NVMe/FC initiator:

```
# cat /sys/module/qla2xxx/parameters/ql2xnvmeenable
1
```

Enable 1MB I/O size (Optional)

ONTAP reports an MDTS (Max Data Transfer Size) of 8 in the Identify Controller data. This means the maximum I/O request size can be up to 1MB. To issue I/O requests of size 1MB for a Broadcom NVMe/FC host, you should increase the `lpfc` value of the `lpfc_sg_seg_cnt` parameter to 256 from the default value of 64.

> These steps don’t apply to Qlogic NVMe/FC hosts.

Steps

1. Set the `lpfc_sg_seg_cnt` parameter to 256:

```
cat /etc/modprobe.d/lpfc.conf
```

   You should see an output similar to the following example:

```
options lpfc lpfc_sg_seg_cnt=256
```

2. Run the `dracut -f` command, and reboot the host.
3. Verify that the value for `lpfc_sg_seg_cnt` is 256:

```
cat /sys/module/lpfc/parameters/lpfc_sg_seg_cnt
```

Configure NVMe/TCP

NVMe/TCP does not have auto-connect functionality. Therefore, if a path goes down and is not reinstated within the default time out period of 10 minutes, NVMe/TCP cannot automatically reconnect. To prevent a time out, you should set the retry period for failover events to at least 30 minutes.

Steps

1. Verify that the initiator port can fetch the discovery log page data across the supported NVMe/TCP LIFs:

```
nvme discover -t tcp -w host-traddr -a traddr
```

Example output:

```
#  nvme discover -t tcp -w 192.168.6.13 -a 192.168.6.15
Discovery Log Number of Records 6, Generation counter 8
=====Discovery Log Entry 0======
trtype: tcp
adrfam: ipv4
subtype: unrecognized
treq: not specified
portid: 0
trsvcid: 8009
subnqn: nqn.1992-
08.com.netapp:sn.1c6ac66338e711eda41dd039ea3ad566:discovery
traddr: 192.168.6.17
sectype: none
=====Discovery Log Entry 1======
trtype: tcp
adrfam: ipv4
subtype: unrecognized
treq: not specified
portid: 1
trsvcid: 8009
subnqn: nqn.1992-
08.com.netapp:sn.1c6ac66338e711eda41dd039ea3ad566:discovery
traddr: 192.168.5.17
sectype: none
=====Discovery Log Entry 2======
trtype: tcp
adrfam: ipv4
subtype: unrecognized
treq: not specified
portid: 2
trsvcid: 8009
subnqn: nqn.1992-
08.com.netapp:sn.1c6ac66338e711eda41dd039ea3ad566:discovery
```

```
traddr: 192.168.6.15
sectype: none
=====Discovery Log Entry 3======
trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 0
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.1c6ac66338e711eda41dd039ea3ad566:subsystem.host_95
traddr: 192.168.6.17
sectype: none
..........
```

2. Verify that the other NVMe/TCP initiator-target LIF combinations can successfully fetch discovery log page data:

```
nvme discover -t tcp -w host-traddr -a traddr
```

Example output:

```
# nvme discover -t tcp -w 192.168.5.13 -a 192.168.5.15
# nvme discover -t tcp -w 192.168.5.13 -a 192.168.5.17
# nvme discover -t tcp -w 192.168.6.13 -a 192.168.6.15
# nvme discover -t tcp -w 192.168.6.13 -a 192.168.6.17
```

3. Run the `nvme connect-all` command across all the supported NVMe/TCP initiator-target LIFs across the nodes, and set the controller loss timeout period for at least 30 minutes or 1800 seconds:

```
nvme connect-all -t tcp -w host-traddr -a traddr -l 1800
```

Example output:

```
# nvme connect-all -t tcp -w 192.168.5.13 -a 192.168.5.15 -l 1800
# nvme connect-all -t tcp -w 192.168.5.13 -a 192.168.5.17 -l 1800
# nvme connect-all -t tcp -w 192.168.6.13 -a 192.168.6.15 -l 1800
# nvme connect-all -t tcp -w 192.168.6.13 -a 192.168.6.17 -l 1800
```

Validate NVMe-oF

You can use the following procedure to validate NVMe-oF.

Steps

1. Verify that in-kernel NVMe multipath is enabled:

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

2. Verify that the appropriate NVMe-oF settings (such as `model` set to `NetApp ONTAP Controller` and load balancing `iopolicy` set to `round-robin`) for the respective ONTAP namespaces correctly reflect on the host:

```
# cat /sys/class/nvme-subsystem/nvme-subsys*/model
NetApp ONTAP Controller
NetApp ONTAP Controller
```

```
# cat /sys/class/nvme-subsystem/nvme-subsys*/iopolicy
round-robin
round-robin
```

3. Verify that the namespaces are created and correctly discovered on the host:

```
# nvme list
```

Example output:

```
Node         SN                   Model
---------------------------------------------------------
/dev/nvme0n1 814vWBNRwf9HAAAAAAAB NetApp ONTAP Controller
/dev/nvme0n2 814vWBNRwf9HAAAAAAAB NetApp ONTAP Controller
/dev/nvme0n3 814vWBNRwf9HAAAAAAAB NetApp ONTAP Controller



Namespace Usage    Format             FW             Rev
-----------------------------------------------------------
1                 85.90 GB / 85.90 GB  4 KiB + 0 B   FFFFFFFF
2                 85.90 GB / 85.90 GB  24 KiB + 0 B  FFFFFFFF
3                   85.90 GB / 85.90 GB  4 KiB + 0 B   FFFFFFFF
```

4. Verify that the controller state of each path is live and has the correct ANA status:

NVMe/FC

```
# nvme list-subsys /dev/nvme0n1
```

Example output:

```
nvme-subsys0 - NQN=nqn.1992-
08.com.netapp:sn.5f5f2c4aa73b11e9967e00a098df41bd:subsystem.nvme_ss_
ol_1
\
+- nvme0 fc traddr=nn-0x203700a098dfdd91:pn-0x203800a098dfdd91
host_traddr=nn-0x200000109b1c1204:pn-0x100000109b1c1204 live non-
optimized
+- nvme1 fc traddr=nn-0x203700a098dfdd91:pn-0x203900a098dfdd91
host_traddr=nn-0x200000109b1c1204:pn-0x100000109b1c1204 live non-
optimized
+- nvme2 fc traddr=nn-0x203700a098dfdd91:pn-0x203a00a098dfdd91
host_traddr=nn-0x200000109b1c1205:pn-0x100000109b1c1205 live
optimized
+- nvme3 fc traddr=nn-0x203700a098dfdd91:pn-0x203d00a098dfdd91
host_traddr=nn-0x200000109b1c1205:pn-0x100000109b1c1205 live
optimized
```

NVMe/TCP

```
nvme list-subsys /dev/nvme1n22
```

Example output

```
nvme-subsys1 - NQN=nqn.1992-
08.com.netapp:sn.68c036aaa3cf11edbb95d039ea243511:subsystem.tcp
\
+- nvme2 tcp
traddr=192.168.8.49,trsvcid=4420,host_traddr=192.168.8.1 live non-
optimized
+- nvme3 tcp
traddr=192.168.8.48,trsvcid=4420,host_traddr=192.168.8.1 live non-
optimized
+- nvme6 tcp
traddr=192.168.9.49,trsvcid=4420,host_traddr=192.168.9.1 live
optimized
+- nvme7 tcp
traddr=192.168.9.48,trsvcid=4420,host_traddr=192.168.9.1 live
optimized
```

5. Verify that the NetApp plug-in displays the correct values for each ONTAP namespace device:

Column

```
# nvme netapp ontapdevices -o column
```

Example output:

```
Device        Vserver   Namespace Path
----------------------- ------------------------------
/dev/nvme0n1   vs_ol_nvme  /vol/ol_nvme_vol_1_1_0/ol_nvme_ns
/dev/nvme0n2   vs_ol_nvme  /vol/ol_nvme_vol_1_0_0/ol_nvme_ns
/dev/nvme0n3   vs_ol_nvme  /vol/ol_nvme_vol_1_1_1/ol_nvme_ns





NSID       UUID                                   Size
------------------------------------------------------------
1          72b887b1-5fb6-47b8-be0b-33326e2542e2   85.90GB
2          04bf9f6e-9031-40ea-99c7-a1a61b2d7d08   85.90GB
3          264823b1-8e03-4155-80dd-e904237014a4   85.90GB
```

JSON

```
# nvme netapp ontapdevices -o json
```

Example output

```
{
"ONTAPdevices" : [
    {
        "Device" : "/dev/nvme0n1",
        "Vserver" : "vs_ol_nvme",
        "Namespace_Path" : "/vol/ol_nvme_vol_1_1_0/ol_nvme_ns",
        "NSID" : 1,
        "UUID" : "72b887b1-5fb6-47b8-be0b-33326e2542e2",
        "Size" : "85.90GB",
        "LBA_Data_Size" : 4096,
        "Namespace_Size" : 20971520
    },
    {
        "Device" : "/dev/nvme0n2",
        "Vserver" : "vs_ol_nvme",
        "Namespace_Path" : "/vol/ol_nvme_vol_1_0_0/ol_nvme_ns",
        "NSID" : 2,
        "UUID" : "04bf9f6e-9031-40ea-99c7-a1a61b2d7d08",
        "Size" : "85.90GB",
        "LBA_Data_Size" : 4096,
        "Namespace_Size" : 20971520
      },
      {
         "Device" : "/dev/nvme0n3",
         "Vserver" : "vs_ol_nvme",
         "Namespace_Path" : "/vol/ol_nvme_vol_1_1_1/ol_nvme_ns",
         "NSID" : 3,
         "UUID" : "264823b1-8e03-4155-80dd-e904237014a4",
         "Size" : "85.90GB",
         "LBA_Data_Size" : 4096,
         "Namespace_Size" : 20971520
       },
  ]
}
```

Known issues

The NVMe-oF host configuration for OL 8.8 with ONTAP release has the following known issues:

NetApp Bug ID	Title	Description
1517321	Oracle Linux 8.8 NVMe-oF hosts create duplicate PDCs	On OL 8.8 NVMe-oF hosts, Persistent Discovery Controllers (PDCs) are created by passing the `-p` option to the `nvme discover` command. For a given initiator-target combination, each execution of the `nvme discover` command is expected to create one PDC. However, beginning with OL 8.x, NVMe-oF hosts create duplicate PDCs. This wastes resources on both the host and the target.

NVMe-oF Host Configuration for Oracle Linux 8.7 with ONTAP

NVMe over Fabrics (NVMe-oF), including NVMe over Fibre Channel (NVMe/FC) and other transports, is supported with Oracle Linux (OL) 8.7 with Asymmetric Namespace Access (ANA). In NVMe-oF environments, ANA is the equivalent of ALUA multipathing in iSCSI and FC environments and is implemented with in-kernel NVMe multipath.

The following support is available for the NVMe/FC host configuration for OL 8.7 with ONTAP:

- Support for NVMe over TCP (NVMe/TCP) in addition to NVMe/FC. The NetApp plug-in in the native `nvme-cli` package displays ONTAP details for both NVMe/FC and NVMe/TCP namespaces.
- Use of NVMe and SCSI co-existent traffic on the same host on a given host bus adapter (HBA), without the explicit dm-multipath settings to prevent claiming NVMe namespaces.

For additional details on supported configurations, see the Interoperability Matrix Tool.

Features

- OL 8.7 has in-kernel NVMe multipath enabled for NVMe namespaces by default, therefore, there is no need for explicit settings.

Known limitations

SAN booting using the NVMe-oF protocol is currently not supported.

Validate software versions

You can use the following procedure to validate the minimum supported OL 8.7 software versions.

Steps

1. Install OL 8.7 GA on the server. After the installation is complete, verify that you are running the specified OL 8.7 GA kernel.

   ```
   # uname -r
   ```

   Example output:

```
5.15.0-3.60.5.1.el8uek.x86_64
```

2. Install the `nvme-cli` package:

```
# rpm -qa|grep nvme-cli
```

Example output:

```
nvme-cli-1.16-5.el8.x86_64
```

3. On the Oracle Linux 8.7 host, check the `hostnqn` string at `/etc/nvme/hostnqn`:

```
# cat /etc/nvme/hostnqn
```

Example output:

```
nqn.2014-08.org.nvmexpress:uuid:791c54eb-545d-4ed3-8d41-91a0a53d4b24
```

4. Verify that the `hostnqn` string matches the `hostnqn` string for the corresponding subsystem on the ONTAP array:

```
::> vserver nvme subsystem host show -vserver vs_ol_nvme
```

Example output:

```
Vserver     Subsystem          Host NQN
----------- ---------------
----------------------------------------------------------
vs_ol_nvme  nvme_ss_ol_1    nqn.2014-08.org.nvmexpress:uuid:791c54eb-
545d-4ed3-8d41-91a0a53d4b24
```

If the `hostnqn` strings do not match, you can use the `vserver modify` command to update the `hostnqn` string on your corresponding ONTAP array subsystem to match the `hostnqn` string from `/etc/nvme/hostnqn` on the host.

5. Reboot the host.

If you intend to run both NVMe and SCSI traffic on the same Oracle Linux 8.7 host, NetApp recommends using the in-kernel NVMe multipath for ONTAP namespaces and dm-multipath for ONTAP LUNs respectively. This also means the ONTAP namespaces should be blacklisted in dm-multipath to prevent dm-multipath from claiming these namespace devices. You can do this by adding the `enable_foreign` setting to the `/etc/multipath.conf` file:

```
#cat /etc/multipath.conf
defaults {
    enable_foreign  NONE
}
```

Restart the multipathd daemon by running the `systemctl restart multipathd` command to apply the new settings.

Configure NVMe/FC

You can configure NVMe/FC for Broadcom/Emulex or Marvell/Qlogic adapters.

Broadcom/Emulex

Steps

1. Verify that you are using the supported adapter model.

   ```
   # cat /sys/class/scsi_host/host*/modelname
   ```

   Example output:

   ```
   LPe32002-M2
   LPe32002-M2
   ```

   ```
   # cat /sys/class/scsi_host/host*/modeldesc
   ```

   Example output:

   ```
   Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
   Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
   ```

2. Verify that you are using the recommended Broadcom `lpfc` firmware and inbox driver:

   ```
   # cat /sys/class/scsi_host/host*/fwrev
   12.8.614.23, sli-4:2:c
   12.8.614.23, sli-4:2:c


   # cat /sys/module/lpfc/version
   0:14.0.0.1
   ```

   For the current list of supported adapter driver and firmware versions, see the Interoperability Matrix Tool.

3. Verify that `lpfc_enable_fc4_type` is set to `3`:

   ```
   # cat /sys/module/lpfc/parameters/lpfc_enable_fc4_type
   3
   ```

4. Verify that the initiator ports are up and running, and that you can see the target LIFs:

```
# cat /sys/class/fc_host/host*/port_name
0x100000109b3c081f
0x100000109b3c0820
```

```
# cat /sys/class/fc_host/host*/port_state
Online
Online
```

```
# cat /sys/class/scsi_host/host*/nvme_info
NVME Initiator Enabled
XRI Dist lpfc0 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc0 WWPN x100000109b3c081f WWNN x200000109b3c081f DID
x060300 ONLINE
NVME RPORT WWPN x2010d039ea2c3e2d WWNN x200fd039ea2c3e2d DID x061f0e
TARGET DISCSRVC ONLINE
NVME RPORT WWPN x2011d039ea2c3e2d WWNN x200fd039ea2c3e2d DID x06270f
TARGET DISCSRVC ONLINE
NVME Statistics
LS: Xmt 0000000a71 Cmpl 0000000a71 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 00000000558611c6 Issue 000000005578bb69 OutIO
fffffffffff2a9a3
abort 0000007a noxri 00000000 nondlp 00000447 qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 00000a8e Err 0000e2a8
NVME Initiator Enabled
XRI Dist lpfc1 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc1 WWPN x100000109b3c0820 WWNN x200000109b3c0820 DID
x060200 ONLINE
NVME RPORT WWPN x2015d039ea2c3e2d WWNN x200fd039ea2c3e2d DID x062e0c
TARGET DISCSRVC ONLINE
NVME RPORT WWPN x2014d039ea2c3e2d WWNN x200fd039ea2c3e2d DID x06290f
TARGET DISCSRVC ONLINE
NVME Statistics
LS: Xmt 0000000a69 Cmpl 0000000a69 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 0000000055814701 Issue 0000000055744b1c OutIO
fffffffffff3041b
abort 00000046 noxri 00000000 nondlp 0000043f qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 00000a89 Err 0000e2f3
```

Marvell/Qlogic FC Adapter for NVMe/FC

The native inbox qla2xxx driver included in the OL 8.7 GA kernel has the latest upstream fixes. These fixes are essential for ONTAP support.

Steps

1. Verify that you are running the supported adapter driver and firmware versions:

```
# cat /sys/class/fc_host/host*/symbolic_name
```

Example output

```
QLE2742 FW:v9.10.11 DVR:v10.02.06.200-k
QLE2742 FW:v9.10.11 DVR:v10.02.06.200-k
```

2. Verify that `ql2xnvmeenable` is set. This enables the Marvell adapter to function as an NVMe/FC initiator:

```
# cat /sys/module/qla2xxx/parameters/ql2xnvmeenable
1
```

Enable 1MB I/O (Optional)

ONTAP reports an MDTS (Max Data Transfer Size) of 8 in the Identify Controller data. This means the maximum I/O request size can be up to 1MB. To issue I/O requests of size 1MB for a Broadcom NVMe/FC host, you should increase the `lpfc` value of the `lpfc_sg_seg_cnt` parameter to 256 from the default value of 64.

These steps don’t apply to Qlogic NVMe/FC hosts.

Steps

1. Set the `lpfc_sg_seg_cnt` parameter to 256:

```
cat /etc/modprobe.d/lpfc.conf
```

You should see an output similar to the following example:

```
options lpfc lpfc_sg_seg_cnt=256
```

2. Run the `dracut -f` command, and reboot the host.
3. Verify that the value for `lpfc_sg_seg_cnt` is 256:

```
cat /sys/module/lpfc/parameters/lpfc_sg_seg_cnt
```

Configure NVMe/TCP

NVMe/TCP does not have auto-connect functionality. Therefore, if a path goes down and is not reinstated within the default time out period of 10 minutes, NVMe/TCP cannot automatically reconnect. To prevent a time out, you should set the retry period for failover events to at least 30 minutes.

Steps

1. Verify that the initiator port can fetch the discovery log page data across the supported NVMe/TCP LIFs:

   ```
   nvme discover -t tcp -w host-traddr -a traddr
   ```

   Example output:

   ```
   #  nvme discover -t tcp -w 192.168.6.13 -a 192.168.6.15
   Discovery Log Number of Records 6, Generation counter 8
   =====Discovery Log Entry 0======
   trtype: tcp
   adrfam: ipv4
   subtype: unrecognized
   treq: not specified
   portid: 0
   trsvcid: 8009
   subnqn: nqn.1992-
   08.com.netapp:sn.1c6ac66338e711eda41dd039ea3ad566:discovery
   traddr: 192.168.6.17
   sectype: none
   =====Discovery Log Entry 1======
   trtype: tcp
   adrfam: ipv4
   subtype: unrecognized
   treq: not specified
   portid: 1
   trsvcid: 8009
   subnqn: nqn.1992-
   08.com.netapp:sn.1c6ac66338e711eda41dd039ea3ad566:discovery
   traddr: 192.168.5.17
   sectype: none
   =====Discovery Log Entry 2======
   trtype: tcp
   adrfam: ipv4
   subtype: unrecognized
   treq: not specified
   ```

```
portid: 2
trsvcid: 8009
subnqn: nqn.1992-
08.com.netapp:sn.1c6ac66338e711eda41dd039ea3ad566:discovery
traddr: 192.168.6.15
sectype: none
=====Discovery Log Entry 3======
trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 0
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.1c6ac66338e711eda41dd039ea3ad566:subsystem.host_95
traddr: 192.168.6.17
sectype: none
..........
```

2. Verify that the other NVMe/TCP initiator-target LIF combinations are able to successfully fetch discovery log page data.

```
nvme discover -t tcp -w host-traddr -a traddr
```

Example output:

```
# nvme discover -t tcp -w 192.168.5.13 -a 192.168.5.15
# nvme discover -t tcp -w 192.168.5.13 -a 192.168.5.17
# nvme discover -t tcp -w 192.168.6.13 -a 192.168.6.15
# nvme discover -t tcp -w 192.168.6.13 -a 192.168.6.17
```

3. Run the `nvme connect-all` command across all the supported NVMe/TCP initiator-target LIFs across the nodes, and set the controller loss timeout period for at least 30 minutes or 1800 seconds:

```
nvme connect-all -t tcp -w host-traddr -a traddr -l 1800
```

Example output:

```
# nvme connect-all -t tcp -w 192.168.5.13 -a 192.168.5.15 -l 1800
# nvme connect-all -t tcp -w 192.168.5.13 -a 192.168.5.17 -l 1800
# nvme connect-all -t tcp -w 192.168.6.13 -a 192.168.6.15 -l 1800
# nvme connect-all -t tcp -w 192.168.6.13 -a 192.168.6.17 -l 1800
```

Validate NVMe-oF

You can use the following procedure to validate NVMe-oF.

Steps

1. Verify that in-kernel NVMe multipath is enabled by checking:

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

2. Verify that the appropriate NVMe-oF settings (such as `model` set to `NetApp ONTAP Controller` and load balancing `iopolicy` set to `round-robin`) for the respective ONTAP namespaces correctly reflect on the host:

```
# cat /sys/class/nvme-subsystem/nvme-subsys*/model
NetApp ONTAP Controller
NetApp ONTAP Controller
```

```
# cat /sys/class/nvme-subsystem/nvme-subsys*/iopolicy
round-robin
round-robin
```

3. Verify that the namespaces are created and correctly discovered on the host:

```
# nvme list
```

Example output:

```
Node         SN                   Model
---------------------------------------------------------
/dev/nvme0n1 814vWBNRwf9HAAAAAAAB NetApp ONTAP Controller
/dev/nvme0n2 814vWBNRwf9HAAAAAAAB NetApp ONTAP Controller
/dev/nvme0n3 814vWBNRwf9HAAAAAAAB NetApp ONTAP Controller


Namespace Usage    Format             FW             Rev
-----------------------------------------------------------
1                 85.90 GB / 85.90 GB  4 KiB + 0 B   FFFFFFFF
2                 85.90 GB / 85.90 GB  24 KiB + 0 B  FFFFFFFF
3                 85.90 GB / 85.90 GB  4 KiB + 0 B   FFFFFFFF
```

4. Verify that the controller state of each path is live and has the correct ANA status:

NVMe/FC

```
# nvme list-subsys /dev/nvme0n1
```

Example output:

```
nvme-subsys0 - NQN=nqn.1992-
08.com.netapp:sn.5f5f2c4aa73b11e9967e00a098df41bd:subsystem.nvme_ss_
ol_1
\
+- nvme0 fc traddr=nn-0x203700a098dfdd91:pn-0x203800a098dfdd91
host_traddr=nn-0x200000109b1c1204:pn-0x100000109b1c1204 live non-
optimized
+- nvme1 fc traddr=nn-0x203700a098dfdd91:pn-0x203900a098dfdd91
host_traddr=nn-0x200000109b1c1204:pn-0x100000109b1c1204 live non-
optimized
+- nvme2 fc traddr=nn-0x203700a098dfdd91:pn-0x203a00a098dfdd91
host_traddr=nn-0x200000109b1c1205:pn-0x100000109b1c1205 live
optimized
+- nvme3 fc traddr=nn-0x203700a098dfdd91:pn-0x203d00a098dfdd91
host_traddr=nn-0x200000109b1c1205:pn-0x100000109b1c1205 live
optimized
```

NVMe/TCP

```
# nvme list-subsys /dev/nvme1n40
```

Example output:

```
nvme-subsys1 - NQN=nqn.1992-
08.com.netapp:sn.68c036aaa3cf11edbb95d039ea243511:subsystem.tcp
\
+- nvme2 tcp
traddr=192.168.8.49,trsvcid=4420,host_traddr=192.168.8.1 live non-
optimized
+- nvme3 tcp
traddr=192.168.8.48,trsvcid=4420,host_traddr=192.168.8.1 live non-
optimized
+- nvme6 tcp
traddr=192.168.9.49,trsvcid=4420,host_traddr=192.168.9.1 live
optimized
+- nvme7 tcp
traddr=192.168.9.48,trsvcid=4420,host_traddr=192.168.9.1 live
optimized
```

5. Verify that the NetApp plug-in displays the correct values for each ONTAP namespace device:

Column

```
# nvme netapp ontapdevices -o column
```

Example output:

```
Device          Vserver   Namespace Path
----------------------- ------------------------------
/dev/nvme0n1   vs_ol_nvme  /vol/ol_nvme_vol_1_1_0/ol_nvme_ns
/dev/nvme0n2   vs_ol_nvme  /vol/ol_nvme_vol_1_0_0/ol_nvme_ns
/dev/nvme0n3   vs_ol_nvme  /vol/ol_nvme_vol_1_1_1/ol_nvme_ns


NSID       UUID                                   Size
------------------------------------------------------------
1          72b887b1-5fb6-47b8-be0b-33326e2542e2   85.90GB
2          04bf9f6e-9031-40ea-99c7-a1a61b2d7d08   85.90GB
3          264823b1-8e03-4155-80dd-e904237014a4   85.90GB
```

JSON

```
# nvme netapp ontapdevices -o json
```

Example output

```
{
"ONTAPdevices" : [
    {
        "Device" : "/dev/nvme0n1",
        "Vserver" : "vs_ol_nvme",
        "Namespace_Path" : "/vol/ol_nvme_vol_1_1_0/ol_nvme_ns",
        "NSID" : 1,
        "UUID" : "72b887b1-5fb6-47b8-be0b-33326e2542e2",
        "Size" : "85.90GB",
        "LBA_Data_Size" : 4096,
        "Namespace_Size" : 20971520
    },
    {
        "Device" : "/dev/nvme0n2",
        "Vserver" : "vs_ol_nvme",
        "Namespace_Path" : "/vol/ol_nvme_vol_1_0_0/ol_nvme_ns",
        "NSID" : 2,
        "UUID" : "04bf9f6e-9031-40ea-99c7-a1a61b2d7d08",
        "Size" : "85.90GB",
        "LBA_Data_Size" : 4096,
        "Namespace_Size" : 20971520
      },
      {
         "Device" : "/dev/nvme0n3",
         "Vserver" : "vs_ol_nvme",
         "Namespace_Path" : "/vol/ol_nvme_vol_1_1_1/ol_nvme_ns",
         "NSID" : 3,
         "UUID" : "264823b1-8e03-4155-80dd-e904237014a4",
         "Size" : "85.90GB",
         "LBA_Data_Size" : 4096,
         "Namespace_Size" : 20971520
       },
  ]
}
```

Known issues

The NVMe-oF host configuration for OL 8.7 with ONTAP release has the following known issues:

NetApp Bug ID	Title	Description
1517321	Oracle Linux 8.7 NVMe-oF Hosts create duplicate Persistent Discovery Controllers	On OL 8.7 NVMe-oF hosts, Persistent Discovery Controllers (PDCs) are created by passing the `-p` option to the `nvme discover` command. For a given initiator-target combination, each execution of the `nvme discover` command is expected to create one PDC. However, beginning with OL 8.x, NVMe-oF hosts create duplicate PDCs. This wastes resources on both the host and the target.

NVMe/FC Host Configuration for Oracle Linux 8.6 with ONTAP

You can configure NVMe over Fibre Channel (NVMe/FC) on initiator hosts running Oracle Linux 8.6 and ONTAP as the target.

Supportability

NVMe over Fabrics or NVMe-oF (including NVMe/FC and NVMe/TCP) is supported with Oracle Linux 8.6 with Asymmetric Namespace Access (ANA) that is required for surviving storage failovers (SFOs) on the ONTAP array. ANA is the asymmetric logical unit access (ALUA) equivalent in the NVMe-oF environment, and is currently implemented with in-kernel NVMe Multipath. Using this procedure, you can enable NVMe-oF with in-kernel NVMe Multipath using ANA on Oracle Linux 8.6 and ONTAP as the target.

> You can use the configuration settings provided in this procedure to configure cloud clients connected to Cloud Volumes ONTAP and Amazon FSx for ONTAP.

Features

- Oracle Linux 8.6 has in-kernel NVMe multipath enabled by default for NVMe namepsaces.
- With Oracle Linux 8.6, `nvme-fc auto-connect` scripts are included in the native `nvme-cli` package. You can use these native auto-connect scripts instead of installing external vendor provided outbox auto-connect scripts.
- With Oracle Linux 8.6, a native `udev` rule is provided as part of the `nvme-cli` package which enables round-robin load balancing for NVMe multipath. Therefore, you need not manually create this rule anymore.
- With Oracle Linux 8.6, both NVMe and SCSI traffic can be run on the same host. This is the commonly deployed host configuration. You can configure dm-multipath as usual for SCSI LUNs resulting in mpath devices and also use NVMe multipath to configure NVMe-oF multipath devices (for example, `/dev/nvmeXnY`) on the host.
- With Oracle Linux 8.6, the NetApp plug-in in the native `nvme-cli` package is capable of displaying ONTAP details as well as ONTAP namespaces.

Known limitations

SAN booting using the NVMe-oF protocol is currently not supported.

Configuration requirements

Refer to the Interoperability Matrix Tool for exact details regarding supported configurations.

Enable NVMe/FC with Oracle Linux 8.6

Steps

1. Install Oracle Linux 8.6 GA on the server. After the installation is complete, verify that you are running the specified Oracle Linux 8.6 GA kernel. See the Interoperability Matrix Tool for the current list of supported versions.

```
# uname -r
5.4.17-2136.307.3.1.el8uek.x86_64
```

2. Install the `nvme-cli` package:

```
# rpm -qa|grep nvme-cli
nvme-cli-1.14-3.el8.x86_64
```

3. On the Oracle Linux 8.6 host, check the `hostnqn` string at `/etc/nvme/hostnqn` and verify that it matches the hostnqn string for the corresponding subsystem on the ONTAP array.

```
# cat /etc/nvme/hostnqn
nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0032-3310-8033-b8c04f4c5132
::> vserver nvme subsystem host show -vserver vs_ol_nvme
Vserver     Subsystem          Host NQN
----------- ---------------
----------------------------------------------------------
vs_ol_nvme  nvme_ss_ol_1    nqn.2014-08.org.nvmexpress:uuid:9ed5b327-
b9fc-4cf5-97b3-1b5d986345d1
```

> If the `hostnqn` strings do not match, you should use the `vserver modify` command to update the `hostnqn` string on your corresponding ONTAP array subsystem to match to the `hostnqn` string from `/etc/nvme/hostnqn` on the host:

4. Reboot the host.

If you intend to run both NVMe and SCSI traffic on the same Oracle Linux 8.6 host, NetApp recommends using the in-kernel NVMe multipath for ONTAP namespaces and dm-multipath for ONTAP LUNs respectively. This also means the ONTAP namespaces should be blacklisted in dm-multipath to prevent dm-multipath from claiming these namespace devices. This can be done by adding the `enable_foreign` setting to the `/etc/multipath.conf` file:

```
#cat /etc/multipath.conf
defaults {
    enable_foreign  NONE
}
```

Restart the multipathd daemon by running the `systemctl restart multipathd` command to let the new setting take effect.

Configure Broadcom FC adapter for NVMe/FC

Steps

1. Verify that you are using the supported adapter. For the current list of supported adapters see the Interoperability Matrix Tool:

```
# cat /sys/class/scsi_host/host*/modelname
LPe32002-M2
LPe32002-M2
# cat /sys/class/scsi_host/host*/modeldesc
Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
```

2. Verify that you are using the recommended Broadcom `lpfc` firmware and inbox driver. For the current list of supported adapter driver and firmware versions, see the Interoperability Matrix Tool:

```
# cat /sys/class/scsi_host/host*/fwrev
14.0.505.11, sli-4:2:c
14.0.505.11, sli-4:2:c

# cat /sys/module/lpfc/version
0:12.8.0.11
```

3. Verify that `lpfc_enable_fc4_type` is set to `3`:

```
# cat /sys/module/lpfc/parameters/lpfc_enable_fc4_type
3
```

4. Verify that the initiator ports are up and running, and that you can see the target LIFs:

```
# cat /sys/class/fc_host/host*/port_name
0x100000109b1c1204
0x100000109b1c1205
# cat /sys/class/fc_host/host*/port_state
Online
Online
# cat /sys/class/scsi_host/host*/nvme_info
NVME Initiator Enabled
XRI Dist lpfc0 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc0 WWPN x100000109b1c1204 WWNN x200000109b1c1204 DID
x011d00 ONLINE
NVME RPORT WWPN x203800a098dfdd91 WWNN x203700a098dfdd91 DID x010c07
TARGET DISCSRVC ONLINE
NVME RPORT WWPN x203900a098dfdd91 WWNN x203700a098dfdd91 DID x011507
TARGET DISCSRVC ONLINE
NVME Statistics
LS: Xmt 0000000f78 Cmpl 0000000f78 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000002fe29bba Issue 000000002fe29bc4 OutIO
000000000000000a
abort 00001bc7 noxri 00000000 nondlp 00000000 qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 00001e15 Err 0000d906
NVME Initiator Enabled
XRI Dist lpfc1 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc1 WWPN x100000109b1c1205 WWNN x200000109b1c1205 DID
x011900 ONLINE
NVME RPORT WWPN x203d00a098dfdd91 WWNN x203700a098dfdd91 DID x010007
TARGET DISCSRVC ONLINE
NVME RPORT WWPN x203a00a098dfdd91 WWNN x203700a098dfdd91 DID x012a07
TARGET DISCSRVC ONLINE
NVME Statistics
LS: Xmt 0000000fa8 Cmpl 0000000fa8 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000002e14f170 Issue 000000002e14f17a OutIO
000000000000000a
abort 000016bb noxri 00000000 nondlp 00000000 qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 00001f50 Err 0000d9f8
```

Enable 1MB I/O size

ONTAP reports an MDTS (Max Data Transfer Size) of 8 in the Identify Controller data. This means the maximum I/O request size can be up to 1MB. To issue I/O requests of size 1MB for a Broadcom NVMe/FC host, you should increase the `lpfc` value of the `lpfc_sg_seg_cnt` parameter to 256 from the default value of 64.

> These steps don't apply to Qlogic NVMe/FC hosts.

Steps

1. Set the `lpfc_sg_seg_cnt` parameter to 256:

   ```
   cat /etc/modprobe.d/lpfc.conf
   ```

   You should see an output similar to the following example:

   ```
   options lpfc lpfc_sg_seg_cnt=256
   ```

2. Run the `dracut -f` command, and reboot the host.
3. Verify that the value for `lpfc_sg_seg_cnt` is 256:

   ```
   cat /sys/module/lpfc/parameters/lpfc_sg_seg_cnt
   ```

Configure the Marvell/QLogic FC Adapter for NVMe/FC

The native inbox qla2xxx driver included in the OL 8.6 GA kernel has the latest upstream fixes. These fixes are essential for ONTAP support.

Steps

1. Verify that you are running the supported adapter driver and firmware versions:

   ```
   # cat /sys/class/fc_host/host*/symbolic_name
   QLE2742 FW:v9.08.02 DVR:v10.02.00.106-k
   QLE2742 FW:v9.08.02 DVR:v10.02.00.106-k
   ```

2. Verify that `ql2xnvmeenable` is set which enables the Marvell adapter to function as an NVMe/FC initiator:

   ```
   # cat /sys/module/qla2xxx/parameters/ql2xnvmeenable
   1
   ```

Configure NVMe/TCP

NVMe/TCP does not have auto-connect functionality. Therefore, if a path goes down and is not reinstated within the default time out period of 10 minutes, NVMe/TCP cannot automatically reconnect. To prevent a time out, you should set the retry period for failover events to at least 30 minutes.

Steps

1. Verify that the initiator port can fetch the discovery log page data across the supported NVMe/TCP LIFs:

```
# nvme discover -t tcp -w 192.168.1.8 -a 192.168.1.51
Discovery Log Number of Records 10, Generation counter 119
=====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.56e362e9bb4f11ebbaded039ea165abc:subsystem.nvme_118_tcp
_1
traddr: 192.168.2.56
sectype: none
=====Discovery Log Entry 1======
trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 1
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.56e362e9bb4f11ebbaded039ea165abc:subsystem.nvme_118_tcp
_1
traddr: 192.168.1.51
sectype: none
=====Discovery Log Entry 2======
trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 0
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.56e362e9bb4f11ebbaded039ea165abc:subsystem.nvme_118_tcp
_2
traddr: 192.168.2.56
sectype: none
...
```

2. Similarly, verify that the other NVMe/TCP initiator-target LIF combinations are able to successfully fetch discovery log page data.
   Example,

```
#nvme discover -t tcp -w 192.168.1.8 -a 192.168.1.51
# nvme discover -t tcp -w 192.168.1.8 -a 192.168.1.52
# nvme discover -t tcp -w 192.168.2.9 -a 192.168.2.56
# nvme discover -t tcp -w 192.168.2.9 -a 192.168.2.57
```

3. Now run the `nvme connect-all` command across all the supported NVMe/TCP initiator-target LIFs across the nodes. MAke sure you pass a longer `ctrl_loss_tmo` period (such as, say 30 minutes, which can be set through `-l 1800`) during the `connect-all` so that it would retry for a longer period in the event of a path loss. For example,

```
# nvme connect-all -t tcp -w 192.168.1.8 -a 192.168.1.51 -l 1800
# nvme connect-all -t tcp -w 192.168.1.8 -a 192.168.1.52 -l 1800
# nvme connect-all -t tcp -w 192.168.2.9 -a 192.168.2.56 -l 1800
# nvme connect-all -t tcp -w 192.168.2.9 -a 192.168.2.57 -l 1800
```

Validate NVMe/FC

Steps

1. Verify the following NVMe/FC settings on the Oracle Linux 8.6 host:

```
# cat /sys/module/nvme_core/parameters/multipath
Y
# cat /sys/class/nvme-subsystem/nvme-subsys*/model
NetApp ONTAP Controller
NetApp ONTAP Controller
# cat /sys/class/nvme-subsystem/nvme-subsys*/iopolicy
round-robin
round-robin
```

2. Verify that the namespaces are created and correctly discovered on the host:

```
# nvme list
Node         SN                   Model
---------------------------------------------------------
/dev/nvme0n1 814vWBNRwf9HAAAAAAAB NetApp ONTAP Controller
/dev/nvme0n2 814vWBNRwf9HAAAAAAAB NetApp ONTAP Controller
/dev/nvme0n3 814vWBNRwf9HAAAAAAAB NetApp ONTAP Controller


Namespace Usage    Format             FW             Rev
-----------------------------------------------------------
1                 85.90 GB / 85.90 GB  4 KiB + 0 B   FFFFFFFF
2                 85.90 GB / 85.90 GB  24 KiB + 0 B  FFFFFFFF
3                 85.90 GB / 85.90 GB  4 KiB + 0 B   FFFFFFFF
```

3. Verify that the controller state of each path is live and has the correct ANA status:

```
# nvme list-subsys /dev/nvme0n1
nvme-subsys0 - NQN=nqn.1992-
08.com.netapp:sn.5f5f2c4aa73b11e9967e00a098df41bd:subsystem.nvme_ss_ol_1
\
+- nvme0 fc traddr=nn-0x203700a098dfdd91:pn-0x203800a098dfdd91
host_traddr=nn-0x200000109b1c1204:pn-0x100000109b1c1204 live
inaccessible
+- nvme1 fc traddr=nn-0x203700a098dfdd91:pn-0x203900a098dfdd91
host_traddr=nn-0x200000109b1c1204:pn-0x100000109b1c1204 live
inaccessible
+- nvme2 fc traddr=nn-0x203700a098dfdd91:pn-0x203a00a098dfdd91
host_traddr=nn-0x200000109b1c1205:pn-0x100000109b1c1205 live optimized
+- nvme3 fc traddr=nn-0x203700a098dfdd91:pn-0x203d00a098dfdd91
host_traddr=nn-0x200000109b1c1205:pn-0x100000109b1c1205 live optimized
```

4. Verify that the NetApp plug-in displays the correct values for each ONTAP namespace device:

```
# nvme netapp ontapdevices -o column

Device        Vserver   Namespace Path
----------------------- ------------------------------
/dev/nvme0n1   vs_ol_nvme  /vol/ol_nvme_vol_1_1_0/ol_nvme_ns
/dev/nvme0n2   vs_ol_nvme  /vol/ol_nvme_vol_1_0_0/ol_nvme_ns
/dev/nvme0n3   vs_ol_nvme  /vol/ol_nvme_vol_1_1_1/ol_nvme_ns


NSID       UUID                                   Size
------------------------------------------------------------
1          72b887b1-5fb6-47b8-be0b-33326e2542e2   85.90GB
2          04bf9f6e-9031-40ea-99c7-a1a61b2d7d08   85.90GB
3          264823b1-8e03-4155-80dd-e904237014a4   85.90GB
```

```
# nvme netapp ontapdevices -o json
{
"ONTAPdevices" : [
    {
        "Device" : "/dev/nvme0n1",
        "Vserver" : "vs_ol_nvme",
        "Namespace_Path" : "/vol/ol_nvme_vol_1_1_0/ol_nvme_ns",
        "NSID" : 1,
        "UUID" : "72b887b1-5fb6-47b8-be0b-33326e2542e2",
        "Size" : "85.90GB",
        "LBA_Data_Size" : 4096,
        "Namespace_Size" : 20971520
    },
    {
        "Device" : "/dev/nvme0n2",
        "Vserver" : "vs_ol_nvme",
        "Namespace_Path" : "/vol/ol_nvme_vol_1_0_0/ol_nvme_ns",
        "NSID" : 2,
        "UUID" : "04bf9f6e-9031-40ea-99c7-a1a61b2d7d08",
        "Size" : "85.90GB",
        "LBA_Data_Size" : 4096,
        "Namespace_Size" : 20971520
      },
      {
         "Device" : "/dev/nvme0n3",
         "Vserver" : "vs_ol_nvme",
         "Namespace_Path" : "/vol/ol_nvme_vol_1_1_1/ol_nvme_ns",
         "NSID" : 3,
         "UUID" : "264823b1-8e03-4155-80dd-e904237014a4",
         "Size" : "85.90GB",
         "LBA_Data_Size" : 4096,
         "Namespace_Size" : 20971520
       },
  ]
}
```

Known issues

The NVMe-oF host configuration for OL 8.6 with ONTAP has the following known issues:

NetApp Bug ID	Title	Description
1517321	Oracle Linux 8.6 NVMe-oF Hosts create duplicate Persistent Discovery Controllers	On Oracle Linux 8.6 NVMe over Fabrics (NVMe-oF) hosts, you can use the `nvme discover -p` command to create Persistent Discovery Controllers (PDCs). When this command is used, only one PDC should be created per initiator-target combination. However, if you are running ONTAP 9.10.1 and Oracle Linux 8.6 with an NVMe-oF host, a duplicate PDC is created each time `nvme discover -p` is executed. This leads to unnecessary usage of resources on both the host and the target.

NVMe/FC Host Configuration for Oracle Linux 8.5 with ONTAP

You can configure NVMe over Fibre Channel (NVMe/FC) on initiator hosts running Oracle Linux 8.5 and ONTAP as the target.

Supportability

NVMe over Fabrics or NVMe-oF (including NVMe/FC and NVMe/TCP) is supported with Oracle Linux 8.5 with Asymmetric Namespace Access (ANA) that is required for surviving storage failovers (SFOs) on the ONTAP array. ANA is the asymmetric logical unit access (ALUA) equivalent in the NVMe-oF environment, and is currently implemented with in-kernel NVMe Multipath. Using this procedure, you can enable NVMe-oF with in-kernel NVMe Multipath using ANA on Oracle Linux 8.5 and ONTAP as the target.

> ⓘ You can use the configuration settings provided in this procedure to configure cloud clients connected to Cloud Volumes ONTAP and Amazon FSx for ONTAP.

Features

- Oracle Linux 8.5 has in-kernel NVMe multipath enabled by default for NVMe namepsaces.
- With Oracle Linux 8.5, `nvme-fc auto-connect` scripts are included in the native `nvme-cli` package. You can use these native auto-connect scripts instead of installing external vendor provided outbox auto-connect scripts.
- With Oracle Linux 8.5, a native `udev` rule is provided as part of the `nvme-cli` package which enables round-robin load balancing for NVMe multipath. Therefore, you do not need to manually create this rule anymore.
- With Oracle Linux 8.5, both NVMe and SCSI traffic can be run on the same host. This is the commonly deployed host configuration. You can configure dm-multipath as usual for SCSI LUNs resulting in mpath devices and also use NVMe multipath to configure NVMe-oF multipath devices (for example, `/dev/nvmeXnY`) on the host.
- With Oracle Linux 8.5, the NetApp plugin in the native `nvme-cli` package is capable of displaying ONTAP details as well as ONTAP namespaces.

Known limitations

SAN booting using the NVMe-oF protocol is currently not supported.

Configuration requirements

Refer to the Interoperability Matrix Tool for exact details regarding supported configurations.

Enable NVMe/FC with Oracle Linux 8.5

Steps

1. Install Oracle Linux 8.5 General Availability (GA) on the server. After the installation is complete, verify that you are running the specified Oracle Linux 8.5 GA kernel. See the Interoperability Matrix Tool for the current list of supported versions.

```
# uname -r
5.4.17-2136.309.4.el8uek.x86_64
```

2. Install the `nvme-cli` package.

```
# rpm -qa|grep nvme-cli
nvme-cli-1.14-3.el8.x86_64
```

3. On the Oracle Linux 8.5 host, check the `hostnqn` string at `/etc/nvme/hostnqn` and verify that it matches the `hostnqn` string for the corresponding subsystem on the ONTAP array.

```
# cat /etc/nvme/hostnqn
nqn.2014-08.org.nvmexpress:uuid:9ed5b327-b9fc-4cf5-97b3-1b5d986345d1
::> vserver nvme subsystem host show -vserver vs_ol_nvme

Vserver     Subsystem      Host NQN
---------------------------------------------
vs_ol_nvme nvme_ss_ol_1   nqn.2014-08.org.nvmexpress:uuid:9ed5b327-b9fc-
4cf5-97b3-1b5d986345d1
```

   If the `hostnqn` strings do not match, you should use the `vserver modify` command to update the `hostnqn` string on your corresponding ONTAP array subsystem to match to the `hostnqn` string from `/etc/nvme/hostnqn` on the host.

4. Reboot the host.

If you intend to run both NVMe and SCSI traffic on the same Oracle Linux 8.5 host, NetApp recommends using the in-kernel NVMe multipath for ONTAP namespaces and dm-multipath for ONTAP LUNs respectively. This also means the ONTAP namespaces should be blacklisted in dm-multipath to prevent dm-multipath from claiming these namespace devices. This can be done by adding the `enable_foreign` setting to the `/etc/multipath.conf` file:

```
#cat /etc/multipath.conf
defaults {
    enable_foreign  NONE
}
```

Restart the `multipathd` daemon by running the `systemctl restart multipathd` command to let the new setting take effect.

Configure the Broadcom FC adapter for NVMe/FC

Steps

1. Verify that you are using the supported adapter. For the current list of supported adapters, see the Interoperability Matrix Tool.

```
# cat /sys/class/scsi_host/host*/modelname
LPe32002-M2
LPe32002-M2
# cat /sys/class/scsi_host/host*/modeldesc
Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
```

2. Verify that you are using the recommended Broadcom lpfc firmware and inbox driver. For the current list of supported adapter driver and firmware versions, see the Interoperability Matrix Tool.

```
# cat /sys/class/scsi_host/host*/fwrev
14.0.505.11, sli-4:2:c
14.0.505.11, sli-4:2:c

# cat /sys/module/lpfc/version
0:12.8.0.5
```

3. Verify that `lpfc_enable_fc4_type` is set to 3.

```
# cat /sys/module/lpfc/parameters/lpfc_enable_fc4_type
3
```

4. Verify that the initiator ports are up and running, and you can see the target LIFs.

```
# cat /sys/class/fc_host/host*/port_name
0x100000109b213a00
0x100000109b2139ff
# cat /sys/class/fc_host/host*/port_state
Online
Online
# cat /sys/class/scsi_host/host*/nvme_info

NVME Initiator Enabled
XRI Dist lpfc1 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc1 WWPN x100000109b213a00 WWNN x200000109b213a00 DID
x031700     ONLINE
NVME RPORT WWPN x208cd039ea243510 WWNN x208bd039ea243510 DID x03180a
TARGET DISCSRVC ONLINE
NVME RPORT WWPN x2090d039ea243510 WWNN x208bd039ea243510 DID x03140a
TARGET DISCSRVC ONLINE
NVME Statistics
LS: Xmt 000000000e Cmpl 000000000e Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 0000000000079efc Issue 0000000000079eeb OutIO
ffffffffffffffef
abort 00000002 noxri 00000000 nondlp 00000000 qdepth 00000000 wqerr
00000000 err   00000000
FCP CMPL: xb 00000002 Err 00000004

NVME Initiator Enabled
XRI Dist lpfc0 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc0 WWPN x100000109b2139ff WWNN x200000109b2139ff DID
x031300 ONLINE
NVME RPORT WWPN x208ed039ea243510 WWNN x208bd039ea243510 DID x03230c
TARGET DISCSRVC ONLINE
NVME RPORT WWPN x2092d039ea243510 WWNN x208bd039ea243510 DID x03120c
TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 000000000e Cmpl 000000000e Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 0000000000029ba0 Issue 0000000000029ba2 OutIO
0000000000000002
abort 00000002 noxri 00000000 nondlp 00000000 qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 00000002 Err 00000004
```

Enable 1MB I/O size

ONTAP reports an MDTS (Max Data Transfer Size) of 8 in the Identify Controller data. This means the maximum I/O request size can be up to 1MB. To issue I/O requests of size 1MB for a Broadcom NVMe/FC host, you should increase the `lpfc` value of the `lpfc_sg_seg_cnt` parameter to 256 from the default value of 64.

> These steps don't apply to Qlogic NVMe/FC hosts.

Steps

1. Set the `lpfc_sg_seg_cnt` parameter to 256:

```
cat /etc/modprobe.d/lpfc.conf
```

   You should see an output similar to the following example:

```
options lpfc lpfc_sg_seg_cnt=256
```

2. Run the `dracut -f` command, and reboot the host.
3. Verify that the value for `lpfc_sg_seg_cnt` is 256:

```
cat /sys/module/lpfc/parameters/lpfc_sg_seg_cnt
```

Configure the Marvell/QLogic FC adapter for NVMe/FC

The native inbox qla2xxx driver included in the OL 8.5 GA kernel has the latest upstream fixes. These fixes are essential for ONTAP support.

Steps

1. Verify that you are running the supported adapter driver and firmware versions:

```
# cat /sys/class/fc_host/host*/symbolic_name
QLE2742 FW:v9.06.02 DVR:v10.02.00.106-k
QLE2742 FW:v9.06.02 DVR:v10.02.00.106-k
```

2. Verify `ql2xnvmeenable` is set which enables the Marvell adapter to function as an NVMe/FC initiator.

```
# cat /sys/module/qla2xxx/parameters/ql2xnvmeenable
1
```

Configure NVMe/TCP

NVMe/TCP does not have auto-connect functionality. Therefore, if a path goes down and is not reinstated within the default time out period of 10 minutes, NVMe/TCP cannot automatically reconnect. To prevent a time out, you should set the retry period for failover events to at least 30 minutes.

Steps

1. Verify whether the initiator port is able to fetch discovery log page data across the supported NVMe/TCP LIFs.

```
# nvme discover -t tcp -w 192.168.1.8 -a 192.168.1.51
Discovery Log Number of Records 10, Generation counter 119
=====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.56e362e9bb4f11ebbaded039ea165abc:subsystem.nvme_118_tcp
_1
traddr: 192.168.2.56
sectype: none
=====Discovery Log Entry 1======
trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 1
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.56e362e9bb4f11ebbaded039ea165abc:subsystem.nvme_118_tcp
_1
traddr: 192.168.1.51
sectype: none
=====Discovery Log Entry 2======
trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 0
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.56e362e9bb4f11ebbaded039ea165abc:subsystem.nvme_118_tcp
_2
traddr: 192.168.2.56
sectype: none

...
```

2. Similarly, verify that the other NVMe/TCP initiator-target LIF combinations are able to successfully fetch discovery log page data.
   Example,

```
# nvme discover -t tcp -w 192.168.1.8 -a 192.168.1.51
# nvme discover -t tcp -w 192.168.1.8 -a 192.168.1.52
# nvme discover -t tcp -w 192.168.2.9 -a 192.168.2.56
# nvme discover -t tcp -w 192.168.2.9 -a 192.168.2.57
```

3. Now run the `nvme connect-all` command across all the supported NVMe/TCP initiator-target LIFs across the nodes. Make sure you provide a longer `ctrl_loss_tmo` timer period (such as say 30 minutes, which can be set adding `-l 1800`) during `connect-all` so that it would retry for a longer period in the event of a path loss.
   Example:

```
# nvme connect-all -t tcp -w 192.168.1.8 -a 192.168.1.51 -l 1800
# nvme connect-all -t tcp -w 192.168.1.8 -a 192.168.1.52 -l 1800
# nvme connect-all -t tcp -w 192.168.2.9 -a 192.168.2.56 -l 1800
# nvme connect-all -t tcp -w 192.168.2.9 -a 192.168.2.57 -l 1800
```

Validate NVMe/FC

Steps

1. Verify the following NVMe/FC settings on the Oracle Linux 8.5 host.

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

```
# cat /sys/class/nvme-subsystem/nvme-subsys*/model
NetApp ONTAP Controller
NetApp ONTAP Controller
```

```
# cat /sys/class/nvme-subsystem/nvme-subsys*/iopolicy
round-robin
round-robin
```

2. Verify that the namespaces are created and correctly discovered on the host.

```
# nvme list
Node         SN                    Model
---------------------------------------------------------------
/dev/nvme0n1 814vWBNRwf9HAAAAAAAB  NetApp ONTAP Controller
/dev/nvme0n2 814vWBNRwf9HAAAAAAAB  NetApp ONTAP Controller
/dev/nvme0n3 814vWBNRwf9HAAAAAAAB  NetApp ONTAP Controller

Namespace Usage  Format                 FW            Rev
--------------------------------------------------------------
1                85.90 GB / 85.90 GB     4 KiB + 0 B   FFFFFFFF
2                85.90 GB / 85.90 GB     4 KiB + 0 B   FFFFFFFF
3                85.90 GB / 85.90 GB     4 KiB + 0 B   FFFFFFFF
```

3. Verify that the controller state of each path is live and has the correct ANA status.

```
# nvme list-subsys /dev/nvme0n1
nvme-subsys0 - NQN=nqn.1992-
08.com.netapp:sn.5f5f2c4aa73b11e9967e00a098df41bd:subsystem.nvme_ss_ol_1
\
+- nvme0 fc traddr=nn-0x203700a098dfdd91:pn-0x203800a098dfdd91
host_traddr=nn-0x200000109b1c1204:pn-0x100000109b1c1204 live non-
optimized
+- nvme1 fc traddr=nn-0x203700a098dfdd91:pn-0x203900a098dfdd91
host_traddr=nn-0x200000109b1c1204:pn-0x100000109b1c1204 live non-
optimized
+- nvme2 fc traddr=nn-0x203700a098dfdd91:pn-0x203a00a098dfdd91
host_traddr=nn-0x200000109b1c1205:pn-0x100000109b1c1205 live optimized
+- nvme3 fc traddr=nn-0x203700a098dfdd91:pn-0x203d00a098dfdd91
host_traddr=nn-0x200000109b1c1205:pn-0x100000109b1c1205 live optimized
```

4. Verify the NetApp plug-in displays correct values for each ONTAP namespace device.

```
# nvme netapp ontapdevices -o column
Device       Vserver  Namespace Path
-----------------------------------
/dev/nvme0n1  vs_ol_nvme  /vol/ol_nvme_vol_1_1_0/ol_nvme_ns
/dev/nvme0n2  vs_ol_nvme  /vol/ol_nvme_vol_1_0_0/ol_nvme_ns
/dev/nvme0n3  vs_ol_nvme  /vol/ol_nvme_vol_1_1_1/ol_nvme_ns

NSID    UUID                                   Size
-------------------------------------------------------
1       72b887b1-5fb6-47b8-be0b-33326e2542e2   85.90GB
2       04bf9f6e-9031-40ea-99c7-a1a61b2d7d08   85.90GB
3       264823b1-8e03-4155-80dd-e904237014a4   85.90GB
```

```
# nvme netapp ontapdevices -o json
{
"ONTAPdevices" : [
    {
        "Device" : "/dev/nvme0n1",
        "Vserver" : "vs_ol_nvme",
        "Namespace_Path" : "/vol/ol_nvme_vol_1_1_0/ol_nvme_ns",
        "NSID" : 1,
        "UUID" : "72b887b1-5fb6-47b8-be0b-33326e2542e2",
        "Size" : "85.90GB",
        "LBA_Data_Size" : 4096,
        "Namespace_Size" : 20971520
    },
    {
        "Device" : "/dev/nvme0n2",
        "Vserver" : "vs_ol_nvme",
        "Namespace_Path" : "/vol/ol_nvme_vol_1_0_0/ol_nvme_ns",
        "NSID" : 2,
        "UUID" : "04bf9f6e-9031-40ea-99c7-a1a61b2d7d08",
        "Size" : "85.90GB",
        "LBA_Data_Size" : 4096,
        "Namespace_Size" : 20971520
      },
      {
         "Device" : "/dev/nvme0n3",
         "Vserver" : "vs_ol_nvme",
         "Namespace_Path" : "/vol/ol_nvme_vol_1_1_1/ol_nvme_ns",
         "NSID" : 3,
         "UUID" : "264823b1-8e03-4155-80dd-e904237014a4",
         "Size" : "85.90GB",
         "LBA_Data_Size" : 4096,
         "Namespace_Size" : 20971520
       },
  ]
}
```

Known issues

The NVMe-oF host configuration for OL 8.5 with ONTAP has the following known issues:

NetApp Bug ID	Title	Description
1517321	Oracle Linux 8.5 NVMe-oF Hosts create duplicate Persistent Discovery Controllers	On Oracle Linux 8.5 NVMe over Fabrics (NVMe-oF) hosts, you can use the `nvme discover -p` command to create Persistent Discovery Controllers (PDCs). When this command is used, only one PDC should be created per initiator-target combination. However, if you are running ONTAP 9.10.1 and Oracle Linux 8.5 with an NVMe-oF host, a duplicate PDC is created each time `nvme discover -p` is executed. This leads to unnecessary usage of resources on both the host and the target.

NVMe/FC Host Configuration for Oracle Linux 8.4 with ONTAP

You can configure NVMe over Fibre Channel (NVMe/FC) on initiator hosts running Oracle Linux 8.4 and ONTAP as the target.

Supportability

NVMe over Fabrics or NVMe-oF (including NVMe/FC and NVMe/TCP) is supported with Oracle Linux 8.4 with Asymmetric Namespace Access (ANA), which is required for surviving storage failovers (SFOs) on the ONTAP array. ANA is the asymmetric logical unit access (ALUA) equivalent in the NVMe-oF environment, and is currently implemented with in-kernel NVMe Multipath. This topic details how to enable NVMe-oF with in-kernel NVMe Multipath using ANA on Oracle Linux 8.4 with ONTAP as the target.

> You can use the configuration settings provided in this procedure to configure cloud clients connected to Cloud Volumes ONTAP and Amazon FSx for ONTAP.

Features

- Oracle Linux 8.4 has in-kernel NVMe multipath enabled by default for NVMe namepsaces.
- With Oracle Linux 8.4, `nvme-fc auto-connect` scripts are included in the native `nvme-cli` package. You can use these native auto-connect scripts instead of installing external vendor provided outbox auto-connect scripts.
- With Oracle Linux 8.4, a native `udev` rule is provided as part of the `nvme-cli` package which enables round-robin load balancing for NVMe multipath. Therefore, you do not need to manually create this rule anymore.
- With Oracle Linux 8.4, both NVMe and SCSI traffic can be run on the same host. This is the commonly deployed host configuration. You can configure dm-multipath as usual for SCSI LUNs resulting in mpath devices and also use NVMe multipath to configure NVMe-oF multipath devices (for example, `/dev/nvmeXnY`) on the host.
- With Oracle Linux 8.4, the NetApp plugin in the native `nvme-cli` package is capable of displaying ONTAP details as well as ONTAP namespaces.

Known limitations

SAN booting using the NVMe-oF protocol is currently not supported.

Configuration requirements

Refer to the NetApp Interoperability Matrix (IMT) for exact details on supported configurations.

Enable NVMe/FC

Steps

1. Install Oracle Linux 8.4 GA on the server. After the installation is complete, verify that you are running the specified Oracle Linux 8.4 GA kernel. See the Interoperability Matrix Tool for the current list of supported versions.

```
# uname -r
5.4.17-2102.206.1.el8uek.x86_64
```

2. Install the `nvme-cli` package.

```
# rpm -qa|grep nvme-cli
nvme-cli-1.12-3.el8.x86_64
```

3. On the Oracle Linux 8.4 host, check the hostnqn string at `/etc/nvme/hostnqn` and verify that it matches the hostnqn string for the corresponding subsystem on the ONTAP array.

```
# cat /etc/nvme/hostnqn
nqn.2014-08.org.nvmexpress:uuid:8b43c7c6-e98d-4cc7-a699-d66a69aa714e
::> vserver nvme subsystem host show -vserver vs_coexistance_2

Vserver          Subsystem Host NQN
-------
------------------------------------------------------------------
vs_coexistance_2 nvme_1   nqn.2014-08.org.nvmexpress:uuid:753881b6-3163-
46f9-8145-0d1653d99389
```

> If the hostnqn strings do not match, you should use the `vserver modify` command to update the hostnqn string on your corresponding ONTAP array subsystem to match the hostnqn string from `/etc/nvme/hostnqn` on the host.

4. Reboot the host.

If you intend to run both NVMe and SCSI traffic on the same Oracle Linux 8.4 host, NetApp recommends using the in-kernel NVMe multipath for ONTAP namespaces and dm-multipath for ONTAP LUNs respectively. This also means the ONTAP namespaces should be blacklisted in dm-multipath to prevent dm-multipath from claiming these namespace devices. This can be done by adding the `enable_foreign` setting to the `/etc/multipath.conf` file:

```
#cat /etc/multipath.conf
defaults {
    enable_foreign  NONE
}
```

Restart the multipathd daemon by running the `systemctl restart multipathd` command to let the new setting take effect.

Configuring the Broadcom FC adapter for NVMe/FC

Steps

1. Verify that you are using the supported adapter. For the current list of supported adapters, see the Interoperability Matrix Tool.

```
# cat /sys/class/scsi_host/host*/modelname
LPe32002-M2
LPe32002-M2
```

```
# cat /sys/class/scsi_host/host*/modeldesc
Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
```

2. Verify that you are using the recommended Broadcom lpfc firmware and inbox driver. For the current list of supported adapter driver and firmware versions, see the Interoperability Matrix Tool.

```
# cat /sys/class/scsi_host/host*/fwrev
14.0.505.11, sli-4:2:c
14.0.505.11, sli-4:2:c
```

```
# cat /sys/module/lpfc/version
0:12.8.0.5
```

3. Verify that `lpfc_enable_fc4_type` is set to 3.

```
# cat /sys/module/lpfc/parameters/lpfc_enable_fc4_type
3
```

4. Verify that the initiator ports are up and running, and you can see the target LIFs.

```
# cat /sys/class/fc_host/host*/port_name
0x100000109b213a00
0x100000109b2139ff

# cat /sys/class/fc_host/host*/port_state
Online
Online

# cat /sys/class/scsi_host/host*/nvme_info
NVME Initiator Enabled
XRI Dist lpfc1 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc1 WWPN x100000109b213a00 WWNN x200000109b213a00 DID
x031700     ONLINE
NVME RPORT WWPN x208cd039ea243510 WWNN x208bd039ea243510 DID x03180a
TARGET DISCSRVC ONLINE
NVME RPORT WWPN x2090d039ea243510 WWNN x208bd039ea243510 DID x03140a
TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 000000000e Cmpl 000000000e Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 0000000000079efc Issue 0000000000079eeb OutIO
ffffffffffffffef
abort 00000002 noxri 00000000 nondlp 00000000 qdepth 00000000 wqerr
00000000 err   00000000
FCP CMPL: xb 00000002 Err 00000004

NVME Initiator Enabled
XRI Dist lpfc0 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc0 WWPN x100000109b2139ff WWNN x200000109b2139ff DID
x031300 ONLINE
NVME RPORT WWPN x208ed039ea243510 WWNN x208bd039ea243510 DID x03230c
TARGET DISCSRVC ONLINE
NVME RPORT WWPN x2092d039ea243510 WWNN x208bd039ea243510 DID x03120c
TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 000000000e Cmpl 000000000e Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 0000000000029ba0 Issue 0000000000029ba2 OutIO
0000000000000002
abort 00000002 noxri 00000000 nondlp 00000000 qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 00000002 Err 00000004
```

Enabling 1MB I/O size

ONTAP reports an MDTS (Max Data Transfer Size) of 8 in the Identify Controller data. This means the maximum I/O request size can be up to 1MB. To issue I/O requests of size 1MB for a Broadcom NVMe/FC host, you should increase the `lpfc` value of the `lpfc_sg_seg_cnt` parameter to 256 from the default value of 64.

> ⓘ These steps don't apply to Qlogic NVMe/FC hosts.

Steps

1. Set the `lpfc_sg_seg_cnt` parameter to 256:

   ```
   cat /etc/modprobe.d/lpfc.conf
   ```

   You should see an output similar to the following example:

   ```
   options lpfc lpfc_sg_seg_cnt=256
   ```

2. Run the `dracut -f` command, and reboot the host.
3. Verify that the value for `lpfc_sg_seg_cnt` is 256:

   ```
   cat /sys/module/lpfc/parameters/lpfc_sg_seg_cnt
   ```

Configure the Marvell/QLogic FC adapter for NVMe/FC

The native inbox qla2xxx driver included in the OL 8.4 GA kernel has the latest upstream fixes. These fixes are essential for ONTAP support.

Steps

1. Verify that you are running the supported adapter driver and firmware versions:

   ```
   # cat /sys/class/fc_host/host*/symbolic_name
   QLE2742 FW:v9.08.02 DVR:v10.02.00.103-k
   QLE2742 FW:v9.08.02 DVR:v10.02.00.103-k
   ```

2. Verify that the `ql2xnvmeenable` parameter is set which enables the Marvell adapter to function as an NVMe/FC initiator.

   ```
   # cat /sys/module/qla2xxx/parameters/ql2xnvmeenable
   1
   ```

Configure NVMe/TCP

NVMe/TCP does not have auto-connect functionality. Therefore, if a path goes down and is not reinstated within the default time out period of 10 minutes, NVMe/TCP cannot automatically reconnect. To prevent a time out, you should set the retry period for failover events to at least 30 minutes.

Steps

1. Verify that the initiator port is able to fetch discovery log page data across the supported NVMe/TCP LIFs:

```
# nvme discover -t tcp -w 192.168.1.8 -a 192.168.1.51
Discovery Log Number of Records 10, Generation counter 119
=====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.56e362e9bb4f11ebbaded039ea165abc:subsystem.nvme_118_tcp
_1
traddr: 192.168.2.56
sectype: none
=====Discovery Log Entry 1======
trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 1
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.56e362e9bb4f11ebbaded039ea165abc:subsystem.nvme_118_tcp
_1
traddr: 192.168.1.51
sectype: none
=====Discovery Log Entry 2======
trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 0
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.56e362e9bb4f11ebbaded039ea165abc:subsystem.nvme_118_tcp
_2
traddr: 192.168.2.56
sectype: none
...
```

2. Similarly, verify that other NVMe/TCP initiator-target LIF combinations are able to successfully fetch discovery log page data.
   Example,

```
# nvme discover -t tcp -w 192.168.1.8 -a 192.168.1.51
#nvme discover -t tcp -w 192.168.1.8 -a 192.168.1.52
# nvme discover -t tcp -w 192.168.2.9 -a 192.168.2.56
# nvme discover -t tcp -w 192.168.2.9 -a 192.168.2.57
```

3. Now run the `nvme connect-all` command across all the supported NVMe/TCP initiator-target LIFs across the nodes. Make sure you provide a longer `ctrl_loss_tmo` timer period (30 minutes or more, which can be set adding `-l 1800`) during `connect-all` so that it would retry for a longer period in the event of a path loss.
Example:

```
# nvme connect-all -t tcp -w 192.168.1.8 -a 192.168.1.51 -l 1800
# nvme connect-all -t tcp -w 192.168.1.8 -a 192.168.1.52 -l 1800
# nvme connect-all -t tcp -w 192.168.2.9 -a 192.168.2.56 -l 1800
# nvme connect-all -t tcp -w 192.168.2.9 -a 192.168.2.57 -l 1800
```

Validate NVMe/FC

Steps

1. Verify the following NVMe/FC settings on the Oracle Linux 8.4 host:

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

```
# cat /sys/class/nvme-subsystem/nvme-subsys*/model
NetApp ONTAP Controller
NetApp ONTAP Controller
```

```
# cat /sys/class/nvme-subsystem/nvme-subsys*/iopolicy
round-robin
round-robin
```

2. Verify that the namespaces are created and correctly discoverd on the host:

```
# nvme list
Node                  SN              Model
Namespace
-----------------------------------------------------------------------------------
-----------------
/dev/nvme0n1     814vWBNRwf9HAAAAAAAB  NetApp ONTAP Controller
1
/dev/nvme0n2     814vWBNRwf9HAAAAAAAB  NetApp ONTAP Controller
2
/dev/nvme0n3     814vWBNRwf9HAAAAAAAB  NetApp ONTAP Controller
3

Usage      Format         FW Rev
------------------------------------------------------
85.90 GB / 85.90 GB     4 KiB + 0 B   FFFFFFFF
85.90 GB / 85.90 GB     4 KiB + 0 B   FFFFFFFF
85.90 GB / 85.90 GB     4 KiB + 0 B   FFFFFFFF
```

3. Verify that the controller state of each path is live and has the correct ANA status.

```
# nvme list-subsys /dev/nvme0n1
nvme-subsys0 - NQN=nqn.1992-
08.com.netapp:sn.5f5f2c4aa73b11e9967e00a098df41bd:subsystem.nvme_ss_ol_1
\
+- nvme0 fc traddr=nn-0x203700a098dfdd91:pn-0x203800a098dfdd91
host_traddr=nn-0x200000109b1c1204:pn-0x100000109b1c1204 live non-
optimized
+- nvme1 fc traddr=nn-0x203700a098dfdd91:pn-0x203900a098dfdd91
host_traddr=nn-0x200000109b1c1204:pn-0x100000109b1c1204 live non-
optimized
+- nvme2 fc traddr=nn-0x203700a098dfdd91:pn-0x203a00a098dfdd91
host_traddr=nn-0x200000109b1c1205:pn-0x100000109b1c1205 live optimized
+- nvme3 fc traddr=nn-0x203700a098dfdd91:pn-0x203d00a098dfdd91
host_traddr=nn-0x200000109b1c1205:pn-0x100000109b1c1205 live optimized
```

4. Verify that the NetApp plug-in displays correct values for each ONTAP namespace devices.

```
# nvme netapp ontapdevices -o column
Device                 Vserver          Namespace Path
----------------------- ------------------------------
-------------------------------------------------------------------------
--------- --
/dev/nvme0n1      vs_ol_nvme
/vol/ol_nvme_vol_1_1_0/ol_nvme_ns
/dev/nvme0n2      vs_ol_nvme
/vol/ol_nvme_vol_1_0_0/ol_nvme_ns
/dev/nvme0n3      vs_ol_nvme
/vol/ol_nvme_vol_1_1_1/ol_nvme_ns

NSID        UUID                                  Size
--------------------------------------------------------------
1          72b887b1-5fb6-47b8-be0b-33326e2542e2   85.90GB
2          04bf9f6e-9031-40ea-99c7-a1a61b2d7d08   85.90GB
3          264823b1-8e03-4155-80dd-e904237014a4   85.90GB
```

```
# nvme netapp ontapdevices -o json
{
"ONTAPdevices" : [
    {
        "Device" : "/dev/nvme0n1",
        "Vserver" : "vs_ol_nvme",
        "Namespace_Path" : "/vol/ol_nvme_vol_1_1_0/ol_nvme_ns",
        "NSID" : 1,
        "UUID" : "72b887b1-5fb6-47b8-be0b-33326e2542e2",
        "Size" : "85.90GB",
        "LBA_Data_Size" : 4096,
        "Namespace_Size" : 20971520
    },
    {
        "Device" : "/dev/nvme0n2",
        "Vserver" : "vs_ol_nvme",
        "Namespace_Path" : "/vol/ol_nvme_vol_1_0_0/ol_nvme_ns",
        "NSID" : 2,
        "UUID" : "04bf9f6e-9031-40ea-99c7-a1a61b2d7d08",
        "Size" : "85.90GB",
        "LBA_Data_Size" : 4096,
        "Namespace_Size" : 20971520
      },
      {
         "Device" : "/dev/nvme0n3",
         "Vserver" : "vs_ol_nvme",
         "Namespace_Path" : "/vol/ol_nvme_vol_1_1_1/ol_nvme_ns",
         "NSID" : 3,
         "UUID" : "264823b1-8e03-4155-80dd-e904237014a4",
         "Size" : "85.90GB",
         "LBA_Data_Size" : 4096,
         "Namespace_Size" : 20971520
       },
  ]
}
```

Known issues

The NVMe-oF host configuration for OL 8.4 with ONTAP has the following known issues:

NetApp Bug ID	Title	Description
1517321	Oracle Linux 8.4 NVMe-oF hosts create duplicate Persistent Discovery Controllers	On Oracle Linux 8.4 NVMe over Fabrics (NVMe-oF) hosts, you can use the "nvme discover -p" command to create Persistent Discovery Controllers (PDCs). When this command is used, only one PDC should be created per initiator-target combination. However, if you are running ONTAP 9.10.1 and Oracle Linux 8.4 with an NVMe-oF host, a duplicate PDC is created each time "nvme discover -p" is executed. This leads to unnecessary usage of resources on both the host and the target.

NVMe/FC Host Configuration for Oracle Linux 8.3 with ONTAP

You can configure NVMe over Fibre Channel (NVMe/FC) on initiator hosts running Oracle Linux 8.3 and ONTAP as the target.

Supportability

NVMe over Fabrics or NVMe-oF (including NVMe/FC) is supported with Oracle Linux 8.3 with Asymmetric Namespace Access (ANA) required for surviving storage failovers (SFOs) on the ONTAP array. ANA is the ALUA equivalent in the NVMe-oF environment and is currently implemented with in-kernel NVMe Multipath. Using this procedure, you can enable NVMe-oF with in-kernel NVMe Multipath using ANA on OL 8.3 and ONTAP as the target.

> ⓘ You can use the configuration settings provided in this procedure to configure cloud clients connected to Cloud Volumes ONTAP and Amazon FSx for ONTAP.

Features

- Oracle Linux 8.3 has in-kernel NVMe multipath enabled by default for NVMe namespaces.
- With Oracle Linux 8.3, `nvme-fc auto-connect` scripts are included in the native nvme-cli package. You can use these native auto-connect scripts instead of installing external vendor provided outbox auto-connect scripts.
- With Oracle Linux 8.3, a native `udev` rule is provided as part of the `nvme-cli` package which enables round-robin load balancing for NVMe multipath. Therefore, you do not need to manually create this rule anymore.
- With Oracle Linux 8.3, both NVMe and SCSI traffic can be run on the same host. This is the commonly deployed host configuration. You can configure dm-multipath as usual for SCSI LUNs resulting in mpath devices and also use NVMe multipath to configure NVMe-oF multipath devices (for example, `/dev/nvmeXnY`) on the host.
- With Oracle Linux 8.3, the NetApp plugin in the native `nvme-cli` package is capable of displaying ONTAP details as well as ONTAP namespaces.

Known limitations

SAN booting using the NVMe-oF protocol is currently not supported.

Configuration requirements

Refer to the Interoperability Matrix Tool (IMT) for the current list of supported configurations.

Enable NVMe/FC with Oracle Linux 8.3

Steps

1. Install Oracle Linux 8.3 GA on the server. After the installation is complete, verify that you are running the specified Oracle Linux 8.3 GA kernel. See the Interoperability Matrix Tool for the current list of supported versions.

   ```
   # uname -r
   5.4.17-2011.7.4.el8uek.x86_64
   ```

2. Install the `nvme-cli` package.

   ```
   # rpm -qa|grep nvme-cli

   nvme-cli-1.12-2.el8.x86_64_
   ```

3. On the Oracle Linux 8.3 host, check the hostnqn string at `/etc/nvme/hostnqn` and verify that it matches the `hostnqn` string for the corresponding subsystem on the ONTAP array.

   ```
   #cat /etc/nvme/hostnqn
   nqn.2014-08.org.nvmexpress:uuid:446c21ab-f4c1-47ed-9a8f-1def96f3fed2

   ::> vserver nvme subsystem host show -vserver vs_coexistance_2
   Vserver   Subsystem      Host NQN
   ------- ---------
   ----------------------------------------------------------
   vs_coexistance_2 nvme_1 nqn.2014-08.org.nvmexpress:uuid:446c21ab-f4c1-
   47ed-9a8f-1def96f3fed2
   ```

   If the hostnqn strings do not match, you should use the `vserver modify` command to update the hostnqn string on your corresponding ONTAP array subsystem to match to the hostnqn string from `/etc/nvme/hostnqn` on the host.

4. Reboot the host.

If you intend to run both NVMe and SCSI traffic on the same Oracle Linux 8.3 host, NetApp recommends using the in-kernel NVMe multipath for ONTAP namespaces and `dm-multipath` for ONTAP LUNs respectively. This also means the ONTAP namespaces should be blacklisted in `dm-multipath` to prevent `dm-multipath` from claiming these namespace devices. This can be done by adding the *enable_foreign* setting to the `/etc/multipath.conf` file:

```
#cat /etc/multipath.conf
defaults {
   enable_foreign  NONE
}
```

Restart the multipathd daemon by running the *systemctl restart multipathd* command to let the new setting take effect.

Configure the Broadcom FC adapter for NVMe/FC

Steps

1. Verify that you are using the supported adapter. For the current list of supported adapters, see the Interoperability Matrix Tool.

```
#cat /sys/class/scsi_host/host*/modelname
LPe36002-M2
LPe36002-M2
```

```
#cat /sys/class/scsi_host/host*/modeldesc
Emulex LPe36002-M64 2-Port 64Gb Fibre Channel Adapter
Emulex LPe36002-M64 2-Port 64Gb Fibre Channel Adapter
```

2. Verify that you are using the recommended Broadcom lpfc firmware and inbox driver. For the current list of supported adapter drivers and firmware versions, see the Interoperability Matrix Tool.

```
#cat /sys/class/scsi_host/host*/fwrev
12.8.351.49, sli-4:6:d
12.8.351.49, sli-4:6:d
```

```
#cat /sys/module/lpfc/version
0:12.6.0.3
```

3. Verify that the `lpfc_enable_fc4_type` parameter is set to 3.

```
#cat /sys/module/lpfc/parameters/lpfc_enable_fc4_type
3
```

4. Verify that the initiator ports are up and running, and that you can see the target LIFs.

```
#cat /sys/class/fc_host/host*/port_name
0x100000109bf0447b
0x100000109bf0447c
```

```
#cat /sys/class/fc_host/host*/port_state
Online
Online
```

```
#cat /sys/class/scsi_host/host*/nvme_info

NVME Initiator Enabled
XRI Dist lpfc0 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc0 WWPN x100000109bf0447b WWNN x200000109bf0447b DID
x022400 ONLINE
NVME RPORT WWPN x20e1d039ea243510 WWNN x20e0d039ea243510 DID x0a0314
TARGET DISCSRVC ONLINE
NVME RPORT WWPN x20e4d039ea243510 WWNN x20e0d039ea243510 DID x0a0713
TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 00000003b6 Cmpl 00000003b6 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 00000000be1425e8 Issue 00000000be1425f2 OutIO
000000000000000a
abort 00000251 noxri 00000000 nondlp 00000000 qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 00000c5b Err 0000d176

NVME Initiator Enabled
XRI Dist lpfc1 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc1 WWPN x100000109bf0447c WWNN x200000109bf0447c DID
x021600 ONLINE
NVME RPORT WWPN x20e2d039ea243510 WWNN x20e0d039ea243510 DID x0a0213
TARGET DISCSRVC ONLINE
NVME RPORT WWPN x20e3d039ea243510 WWNN x20e0d039ea243510 DID x0a0614
TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 0000000419 Cmpl 0000000419 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 00000000be37ff65 Issue 00000000be37ff84 OutIO
000000000000001f
abort 0000025a noxri 00000000 nondlp 00000000 qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 00000c89 Err 0000cd87
```

Enable 1MB I/O size

ONTAP reports an MDTS (Max Data Transfer Size) of 8 in the Identify Controller data. This means the maximum I/O request size can be up to 1MB. To issue I/O requests of size 1MB for a Broadcom NVMe/FC host, you should increase the `lpfc` value of the `lpfc_sg_seg_cnt` parameter to 256 from the default value of 64.

These steps don’t apply to Qlogic NVMe/FC hosts.

Steps

1. Set the `lpfc_sg_seg_cnt` parameter to 256:

   ```
   cat /etc/modprobe.d/lpfc.conf
   ```

   You should see an output similar to the following example:

   ```
   options lpfc lpfc_sg_seg_cnt=256
   ```

2. Run the `dracut -f` command, and reboot the host.
3. Verify that the value for `lpfc_sg_seg_cnt` is 256:

   ```
   cat /sys/module/lpfc/parameters/lpfc_sg_seg_cnt
   ```

Configure the Marvell/QLogic FC adapter for NVMe/FC

The native inbox qla2xxx driver included in the OL 8.3 GA kernel has the latest upstream fixes. These fixes are essential for ONTAP support.

Steps

1. Verify that you are running the supported adapter driver and firmware versions:

   ```
   #cat /sys/class/fc_host/host*/symbolic_name
   QLE2742 FW:v9.10.11 DVR:v10.01.00.25-k
   QLE2742 FW:v9.10.11 DVR:v10.01.00.25-k
   ```

2. Verify that the `ql2xnvmeenable` parameter is set which enables the Marvell adapter to function as an NVMe/FC initiator.

   ```
   #cat /sys/module/qla2xxx/parameters/ql2xnvmeenable
   1
   ```

Validate NVMe/FC

Steps

1. Verify the following NVMe/FC settings on the Oracle Linux 8.3 host.

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

#cat /sys/class/nvme-subsystem/nvme-subsys*/model
NetApp ONTAP Controller
NetApp ONTAP Controller

#cat /sys/class/nvme-subsystem/nvme-subsys*/iopolicy
round-robin
round-robin
```

2. Verify that the namespaces are created and correctly discoverd on the host.

```
# nvme list
Node         SN                   Model                   Namespace Usage
Format FW Rev
---------------- --------------------
---------------------------------------- --------- ------------------
/dev/nvme0n1 81Ec-JRMlkL9AAAAAAAB NetApp ONTAP Controller 1         37.58
GB / 37.58 GB 4 KiB + 0 B FFFFFFFF
/dev/nvme0n10 81Ec-JRMlkL9AAAAAAAB NetApp ONTAP Controller 10      37.58
GB / 37.58 GB 4 KiB + 0 B FFFFFFFF
/dev/nvme0n11 81Ec-JRMlkL9AAAAAAAB NetApp ONTAP Controller 11      37.58
GB / 37.58 GB 4 KiB + 0 B FFFFFFFF
/dev/nvme0n12 81Ec-JRMlkL9AAAAAAAB NetApp ONTAP Controller 12      37.58
GB / 37.58 GB 4 KiB + 0 B FFFFFFFF
/dev/nvme0n13 81Ec-JRMlkL9AAAAAAAB NetApp ONTAP Controller 13      37.58
GB / 37.58 GB 4 KiB + 0 B FFFFFFFF
/dev/nvme0n14 81Ec-JRMlkL9AAAAAAAB NetApp ONTAP Controller 14      37.58
GB / 37.58 GB 4 KiB + 0 B FFFFFFFF
/dev/nvme0n15 81Ec-JRMlkL9AAAAAAAB NetApp ONTAP Controller 15      37.58
GB / 37.58 GB 4 KiB + 0 B FFFFFFFF
/dev/nvme0n16 81Ec-JRMlkL9AAAAAAAB NetApp ONTAP Controller 16      37.58
GB / 37.58 GB 4 KiB + 0 B FFFFFFFF
/dev/nvme0n17 81Ec-JRMlkL9AAAAAAAB NetApp ONTAP Controller 17      37.58
GB / 37.58 GB 4 KiB + 0 B FFFFFFFF
/dev/nvme0n18 81Ec-JRMlkL9AAAAAAAB NetApp ONTAP Controller 18      37.58
GB / 37.58 GB 4 KiB + 0 B FFFFFFFF
/dev/nvme0n19 81Ec-JRMlkL9AAAAAAAB NetApp ONTAP Controller 19      37.58
GB / 37.58 GB 4 KiB + 0 B FFFFFFFF
/dev/nvme0n2 81Ec-JRMlkL9AAAAAAAB NetApp ONTAP Controller 2        37.58
GB / 37.58 GB 4 KiB + 0 B FFFFFFFF
/dev/nvme0n20 81Ec-JRMlkL9AAAAAAAB NetApp ONTAP Controller 20      37.58
GB / 37.58 GB 4 KiB + 0 B FFFFFFFF
/dev/nvme0n3 81Ec-JRMlkL9AAAAAAAB NetApp ONTAP Controller 3        37.58
GB / 37.58 GB 4 KiB + 0 B FFFFFFFF
/dev/nvme0n4 81Ec-JRMlkL9AAAAAAAB NetApp ONTAP Controller 4        37.58
GB / 37.58 GB 4 KiB + 0 B FFFFFFFF
/dev/nvme0n5 81Ec-JRMlkL9AAAAAAAB NetApp ONTAP Controller 5        37.58
GB / 37.58 GB 4 KiB + 0 B FFFFFFFF
/dev/nvme0n6 81Ec-JRMlkL9AAAAAAAB NetApp ONTAP Controller 6        37.58
GB / 37.58 GB 4 KiB + 0 B FFFFFFFF
/dev/nvme0n7 81Ec-JRMlkL9AAAAAAAB NetApp ONTAP Controller 7        37.58
GB / 37.58 GB 4 KiB + 0 B FFFFFFFF
/dev/nvme0n8 81Ec-JRMlkL9AAAAAAAB NetApp ONTAP Controller 8        37.58
GB / 37.58 GB 4 KiB + 0 B FFFFFFFF
/dev/nvme0n9 81Ec-JRMlkL9AAAAAAAB NetApp ONTAP Controller 9        37.58
GB / 37.58 GB 4 KiB + 0 B FFFFFFFF
```

3. Verify that the controller state of each path is live and has correct ANA status.

```
# nvme list-subsys /dev/nvme0n1
nvme-subsys0 - NQN=nqn.1992-
08.com.netapp:sn.b79f5c6e4d0911edb3a0d039ea243511:subsystem.nvme_1
\ +
+- nvme214 fc traddr=nn-0x20e0d039ea243510:pn-0x20e4d039ea243510
host_traddr=nn-0x200000109bf0447b:pn-0x100000109bf0447b live non-
optimized
+- nvme219 fc traddr=nn-0x20e0d039ea243510:pn-0x20e2d039ea243510
host_traddr=nn-0x200000109bf0447c:pn-0x100000109bf0447c live optimized
+- nvme223 fc traddr=nn-0x20e0d039ea243510:pn-0x20e1d039ea243510
host_traddr=nn-0x200000109bf0447b:pn-0x100000109bf0447b live optimized
+- nvme228 fc traddr=nn-0x20e0d039ea243510:pn-0x20e3d039ea243510
host_traddr=nn-0x200000109bf0447c:pn-0x100000109bf0447c live non-
optimized
```

4. Verify the NetApp plug-in displays correct values for each ONTAP namespace devices.

```
#nvme netapp ontapdevices -o column
Device      Vserver         Namespace Path              NSID UUID
Size
---------------- -------------------------
-------------------------------------------------- ---- ---------
/dev/nvme0n1 LPE36002_ASA_BL /vol/fcnvme_1_0_0/fcnvme_ns 1 ae10e16d-
1fa4-49c2-8594-02bf6f3b1af1 37.58GB
/dev/nvme0n10 LPE36002_ASA_BL /vol/fcnvme_1_0_9/fcnvme_ns 10 2cf00782-
e2bf-40fe-8495-63e4501727cd 37.58GB
/dev/nvme0n11 LPE36002_ASA_BL /vol/fcnvme_1_1_9/fcnvme_ns 11 fbefbe6c-
90fe-46a2-8a51-47bad9e2eb95 37.58GB
/dev/nvme0n12 LPE36002_ASA_BL /vol/fcnvme_1_1_0/fcnvme_ns 12 0e9cc8fa-
d821-4f1c-8944-3003dcded864 37.58GB
/dev/nvme0n13 LPE36002_ASA_BL /vol/fcnvme_1_1_1/fcnvme_ns 13 31f03b13-
aaf9-4a3f-826b-d126ef007991 37.58GB
/dev/nvme0n14 LPE36002_ASA_BL /vol/fcnvme_1_1_8/fcnvme_ns 14 bcf4627c-
5bf9-4a51-a920-5da174ec9876 37.58GB
/dev/nvme0n15 LPE36002_ASA_BL /vol/fcnvme_1_1_7/fcnvme_ns 15 239fd09d-
11db-46a3-8e94-b5ebe6eb2421 37.58GB
/dev/nvme0n16 LPE36002_ASA_BL /vol/fcnvme_1_1_2/fcnvme_ns 16 1d8004df-
f2e8-48c8-8ccb-ce45f18a15ae 37.58GB
/dev/nvme0n17 LPE36002_ASA_BL /vol/fcnvme_1_1_3/fcnvme_ns 17 4f7afbcf-
3ace-4e6c-9245-cbf5bd155ef4 37.58GB
/dev/nvme0n18 LPE36002_ASA_BL /vol/fcnvme_1_1_4/fcnvme_ns 18 b022c944-
6ebf-4986-a28c-8d9e8ec130c9 37.58GB
/dev/nvme0n19 LPE36002_ASA_BL /vol/fcnvme_1_1_5/fcnvme_ns 19 c457d0c7-
bfea-43aa-97ef-c749d8612a72 37.58GB
/dev/nvme0n2 LPE36002_ASA_BL /vol/fcnvme_1_0_1/fcnvme_ns 2 d2413d8b-
e82e-4412-89d3-c9a751ed7716 37.58GB
/dev/nvme0n20 LPE36002_ASA_BL /vol/fcnvme_1_1_6/fcnvme_ns 20 650e0d93-
967d-4415-874a-36bf9c93c952 37.58GB
/dev/nvme0n3 LPE36002_ASA_BL /vol/fcnvme_1_0_2/fcnvme_ns 3 09d89d9a-
7835-423f-93e7-f6f3ece1dcbc 37.58GB
/dev/nvme0n4 LPE36002_ASA_BL /vol/fcnvme_1_0_3/fcnvme_ns 4 d8e99326-
a67c-469f-b3e9-e0e4a38c8a76 37.58GB
/dev/nvme0n5 LPE36002_ASA_BL /vol/fcnvme_1_0_4/fcnvme_ns 5 c91c71f9-
3e04-4844-b376-30acab6311f1 37.58GB
/dev/nvme0n6 LPE36002_ASA_BL /vol/fcnvme_1_0_5/fcnvme_ns 6 4e8b4345-
e5b1-4aa4-ae1a-adf0de2879ea 37.58GB
/dev/nvme0n7 LPE36002_ASA_BL /vol/fcnvme_1_0_6/fcnvme_ns 7 ef715a16-
a946-4bb8-8735-74f214785874 37.58GB
/dev/nvme0n8 LPE36002_ASA_BL /vol/fcnvme_1_0_7/fcnvme_ns 8 4b038502-
966c-49fd-9631-a17f23478ae0 37.58GB
/dev/nvme0n9 LPE36002_ASA_BL /vol/fcnvme_1_0_8/fcnvme_ns 9 f565724c-
992f-41f6-83b5-da1fe741c09b 37.58GB
```

```
#nvme netapp ontapdevices -o json
{
"ONTAPdevices" : [
{
"Device" : "/dev/nvme0n1",
"Vserver" : "LPE36002_ASA_BL",
"Namespace_Path" : "/vol/fcnvme_1_0_0/fcnvme_ns",
"NSID" : 1,
"UUID" : "ae10e16d-1fa4-49c2-8594-02bf6f3b1af1",
"Size" : "37.58GB",
"LBA_Data_Size" : 4096,
"Namespace_Size" : 9175040
},
{
"Device" : "/dev/nvme0n10",
"Vserver" : "LPE36002_ASA_BL",
"Namespace_Path" : "/vol/fcnvme_1_0_9/fcnvme_ns",
"NSID" : 10,
"UUID" : "2cf00782-e2bf-40fe-8495-63e4501727cd",
"Size" : "37.58GB",
"LBA_Data_Size" : 4096,
"Namespace_Size" : 9175040
},
{
"Device" : "/dev/nvme0n11",
"Vserver" : "LPE36002_ASA_BL",
"Namespace_Path" : "/vol/fcnvme_1_1_9/fcnvme_ns",
"NSID" : 11,
"UUID" : "fbefbe6c-90fe-46a2-8a51-47bad9e2eb95",
"Size" : "37.58GB",
"LBA_Data_Size" : 4096,
"Namespace_Size" : 9175040
},
{
"Device" : "/dev/nvme0n12",
"Vserver" : "LPE36002_ASA_BL",
"Namespace_Path" : "/vol/fcnvme_1_1_0/fcnvme_ns",
"NSID" : 12,
"UUID" : "0e9cc8fa-d821-4f1c-8944-3003dcded864",
"Size" : "37.58GB",
"LBA_Data_Size" : 4096,
"Namespace_Size" : 9175040
},
{
"Device" : "/dev/nvme0n13",
```

```
"Vserver" : "LPE36002_ASA_BL",
"Namespace_Path" : "/vol/fcnvme_1_1_1/fcnvme_ns",
"NSID" : 13,
"UUID" : "31f03b13-aaf9-4a3f-826b-d126ef007991",
"Size" : "37.58GB",
"LBA_Data_Size" : 4096,
"Namespace_Size" : 9175040
},
```

Known issues

The NVMe-oF host configuration for OL 8.3 with ONTAP has the following known issues:

NetApp Bug ID	Title	Description
1517321	Oracle Linux 8.3 NVMe-oF Hosts create duplicate Persistent Discovery Controllers	On Oracle Linux 8.3 NVMe over Fabrics (NVMe-oF) hosts, you can use the `nvme discover -p` command to create Persistent Discovery Controllers (PDCs). When this command is used, only one PDC should be created per initiator-target combination. However, if you are running ONTAP 9.10.1 and Oracle Linux 8.3 with an NVMe-oF host, a duplicate PDC is created each time `nvme discover -p` is executed. This leads to unnecessary usage of resources on both the host and the target.

NVMe/FC Host Configuration for Oracle Linux 8.2 with ONTAP

You can configure NVMe over Fibre Channel (NVMe/FC) on initiator hosts running Oracle Linux 8.2 and ONTAP as the target.

Supportability

Beginning with ONTAP 9.6, NVMe/FC is supported for Oracle Linux 8.2. The Oracle Linux 8.2 host can run both NVMe/FC and FCP traffic through the same fibre channel (FC) initiator adapter ports. See the Hardware Universe for a list of supported FC adapters and controllers. For the current list of supported configurations see the Interoperability Matrix Tool.

> You can use the configuration settings provided in this procedure to configure cloud clients connected to Cloud Volumes ONTAP and Amazon FSx for ONTAP.

Known limitations

SAN booting using the NVMe-oF protocol is currently not supported.

Enable NVMe/FC

1. Install Oracle Linux 8.2 on the server.
2. After the installation is complete, verify that you are running the supported Unbreakable Enterprise kernel. See the Interoperability Matrix Tool.

```
# uname -r
5.4.17-2011.1.2.el8uek.x86_64
```

3. Upgrade the nvme-cli package. The native nvme-cli package contains the NVMe/FC auto-connect scripts, ONTAP udev rule which enables round-robin load balancing for NVMe Multipath as well as the NetApp plug-in for ONTAP namespaces.

```
# rpm -qa|grep nvme-cli
nvme-cli-1.9-5.el8.x86_64
```

4. On the Oracle Linux 8.2 host, check the host NQN string at /etc/nvme/hostnqn and verify that it matches the host NQN string for the corresponding subsystem on the ONTAP array.

```
# cat /etc/nvme/hostnqn
nqn.2014-08.org.nvmexpress:uuid:9ed5b327-b9fc-4cf5-97b3-1b5d986345d1
```

```
::> vserver nvme subsystem host show -vserver vs_ol_nvme
Vserver  Subsystem Host NQN
----------- ---------------
----------------------------------------------------------
vs_ol_nvme
              nvme_ss_ol_1
                                 nqn.2014-
08.org.nvmexpress:uuid:9ed5b327-b9fc-4cf5-97b3-1b5d986345d1
```

If the hostnqn strings do not match, you should use the vserver modify command to update the host NQN string on your corresponding ONTAP array subsystem to match to host NQN string from etc/nvme/hostnqn on the host.

Configure the Broadcom FC Adapter for NVMe/FC

1. Verify that you are using the supported adapter. For the current list of supported adapters, see the Interoperability Matrix Tool.

```
# cat /sys/class/scsi_host/host*/modelname
LPe32002-M2
LPe32002-M2
```

```
# cat /sys/class/scsi_host/host*/modeldesc
Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
```

2. NVMe support in lpfc is already enabled by default:

```
# cat /sys/module/lpfc/parameters/lpfc_enable_fc4_type
3
```

   Newer lpfc drivers (both inbox and outbox) have lpfc_enable_fc4_type default set to 3. Therefore, you do not need to set this explicitly in the /etc/modprobe.d/lpfc.conf.

3. Verify that the NVMe/FC initiator ports are enabled and able to see the target ports, and all are up and running.

   In the example below, only a single initiator port has been enabled and connected with two target LIFs as seen in the below output:

```
# cat /sys/class/scsi_host/host*/nvme_info
NVME Initiator Enabled
XRI Dist lpfc0 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc0 WWPN x100000109b1c1204 WWNN x200000109b1c1204 DID
x011d00 ONLINE
NVME RPORT WWPN x203800a098dfdd91 WWNN x203700a098dfdd91 DID x010c07
TARGET DISCSRVC ONLINE
NVME RPORT WWPN x203900a098dfdd91 WWNN x203700a098dfdd91 DID x011507
TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 0000000f78 Cmpl 0000000f78 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000002fe29bba Issue 000000002fe29bc4 OutIO
000000000000000a
abort 00001bc7 noxri 00000000 nondlp 00000000 qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 00001e15 Err 0000d906

NVME Initiator Enabled
XRI Dist lpfc1 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc1 WWPN x100000109b1c1205 WWNN x200000109b1c1205 DID
x011900 ONLINE
NVME RPORT WWPN x203d00a098dfdd91 WWNN x203700a098dfdd91 DID x010007
TARGET DISCSRVC ONLINE
NVME RPORT WWPN x203a00a098dfdd91 WWNN x203700a098dfdd91 DID x012a07
TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 0000000fa8 Cmpl 0000000fa8 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000002e14f170 Issue 000000002e14f17a OutIO
000000000000000a
abort 000016bb noxri 00000000 nondlp 00000000 qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 00001f50 Err 0000d9f8
```

Validate NVMe/FC

1. Verify the following NVMe/FC settings.

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

```
# cat /sys/class/nvme-subsystem/nvme-subsys*/model
NetApp ONTAP Controller
NetApp ONTAP Controller
```

```
# cat /sys/class/nvme-subsystem/nvme-subsys*/iopolicy
round-robin
round-robin
```

In the above example, two namespaces are mapped to the Oracle Linux 8.2 ANA host. These are visible through four target LIFs: two local node LIFs and two other partner/remote node LIFs. This setup shows as two ANA Optimized and two ANA Inaccessible paths for each namespace on the host.

2. Verify that the namespaces are created.

```
# nvme list
Node                  SN
Model                                               Namespace Usage
Format         FW Rev
--------------------   ---------------------------------------
----------------------------------------------- -----------------
------------------------------- ---------------- ---------------
/dev/nvme0n1     814vWBNRwf9HAAAAAAAB  NetApp ONTAP Controller
1                  85.90 GB / 85.90 GB     4 KiB + 0 B   FFFFFFFF
/dev/nvme0n2     814vWBNRwf9HAAAAAAAB  NetApp ONTAP Controller
2                  85.90 GB / 85.90 GB     4 KiB + 0 B   FFFFFFFF
/dev/nvme0n3     814vWBNRwf9HAAAAAAAB  NetApp ONTAP Controller
3                  85.90 GB / 85.90 GB     4 KiB + 0 B   FFFFFFFF
```

3. Verify the status of the ANA paths.

```
# nvme list-subsys /dev/nvme0n1
nvme-subsys0 - NQN=nqn.1992-
08.com.netapp:sn.5f5f2c4aa73b11e9967e00a098df41bd:subsystem.nvme_ss_ol_1
\
+- nvme0 fc traddr=nn-0x203700a098dfdd91:pn-0x203800a098dfdd91
host_traddr=nn-0x200000109b1c1204:pn-0x100000109b1c1204 live
inaccessible
+- nvme1 fc traddr=nn-0x203700a098dfdd91:pn-0x203900a098dfdd91
host_traddr=nn-0x200000109b1c1204:pn-0x100000109b1c1204 live
inaccessible
+- nvme2 fc traddr=nn-0x203700a098dfdd91:pn-0x203a00a098dfdd91
host_traddr=nn-0x200000109b1c1205:pn-0x100000109b1c1205 live optimized
+- nvme3 fc traddr=nn-0x203700a098dfdd91:pn-0x203d00a098dfdd91
host_traddr=nn-0x200000109b1c1205:pn-0x100000109b1c1205 live optimized
```

4. Verify the NetApp plug-in for ONTAP devices.

```
# nvme netapp ontapdevices -o column
Device                 Vserver                         Namespace Path
NSID    UUID
Size
----------------------- ------------------------------
-----------------------------------------------------------------------
--------- -----------------------------------------------------------
---------
/dev/nvme0n1      vs_ol_nvme
/vol/ol_nvme_vol_1_1_0/ol_nvme_ns              1          72b887b1-5fb6-
47b8-be0b-33326e2542e2   85.90GB
/dev/nvme0n2      vs_ol_nvme
/vol/ol_nvme_vol_1_0_0/ol_nvme_ns              2          04bf9f6e-9031-
40ea-99c7-a1a61b2d7d08    85.90GB
/dev/nvme0n3      vs_ol_nvme
/vol/ol_nvme_vol_1_1_1/ol_nvme_ns              3          264823b1-8e03-
4155-80dd-e904237014a4  85.90GB

# nvme netapp ontapdevices -o json
{
"ONTAPdevices" : [
    {
        "Device" : "/dev/nvme0n1",
        "Vserver" : "vs_ol_nvme",
        "Namespace_Path" : "/vol/ol_nvme_vol_1_1_0/ol_nvme_ns",
        "NSID" : 1,
        "UUID" : "72b887b1-5fb6-47b8-be0b-33326e2542e2",
        "Size" : "85.90GB",
```

```
            "LBA_Data_Size" : 4096,
            "Namespace_Size" : 20971520
        },
        {
            "Device" : "/dev/nvme0n2",
            "Vserver" : "vs_ol_nvme",
            "Namespace_Path" : "/vol/ol_nvme_vol_1_0_0/ol_nvme_ns",
            "NSID" : 2,
            "UUID" : "04bf9f6e-9031-40ea-99c7-a1a61b2d7d08",
            "Size" : "85.90GB",
            "LBA_Data_Size" : 4096,
            "Namespace_Size" : 20971520
          },
          {
             "Device" : "/dev/nvme0n3",
             "Vserver" : "vs_ol_nvme",
             "Namespace_Path" : "/vol/ol_nvme_vol_1_1_1/ol_nvme_ns",
             "NSID" : 3,
             "UUID" : "264823b1-8e03-4155-80dd-e904237014a4",
             "Size" : "85.90GB",
             "LBA_Data_Size" : 4096,
             "Namespace_Size" : 20971520
           },
    ]
}
```

Enable 1MB I/O size for Broadcom NVMe/FC

ONTAP reports an MDTS (Max Data Transfer Size) of 8 in the Identify Controller data. This means the maximum I/O request size can be up to 1MB. To issue I/O requests of size 1MB for a Broadcom NVMe/FC host, you should increase the `lpfc` value of the `lpfc_sg_seg_cnt` parameter to 256 from the default value of 64.

ⓘ These steps don't apply to Qlogic NVMe/FC hosts.

Steps

1. Set the `lpfc_sg_seg_cnt` parameter to 256:

```
cat /etc/modprobe.d/lpfc.conf
```

You should see an output similar to the following example:

```
options lpfc lpfc_sg_seg_cnt=256
```

2. Run the `dracut -f` command, and reboot the host.
3. Verify that the value for `lpfc_sg_seg_cnt` is 256:

```
cat /sys/module/lpfc/parameters/lpfc_sg_seg_cnt
```

NVMe/FC Host Configuration for Oracle Linux 8.1 with ONTAP

You can configure NVMe over Fibre Channel (NVMe/FC) on initiator hosts running Oracle Linux 8.1 and ONTAP as the target.

Supportability

Beginning with ONTAP 9.6, NVMe/FC is supported for Oracle Linux 8.1. The Oracle Linux 8.1 host can run both NVMe and SCSI traffic through the same fibre channel (FC) initiator adapter ports. Note that the Broadcom initiator can serve both NVMe/FC and FCP traffic through the same FC adapter ports. See the Hardware Universe for a list of supported FC adapters and controllers. For the current list of supported configurations see the Interoperability Matrix Tool.

> You can use the configuration settings provided in this procedure to configure cloud clients connected to Cloud Volumes ONTAP and Amazon FSx for ONTAP.

Known limitations

- Native NVMe/FC auto-connect scripts are not available in the nvme-cli package. Use the HBA vendor provided external auto-connect scripts.
- By default, round-robin load balancing is not enabled in NVMe Multipath. You must write a udev rule to enable this functionality. Steps are provided in the section on Enabling NVMe/FC on Oracle Linux 8.1.
- There is no sanlun support for NVMe/FC and, as a consequence, no Linux Host Utilities support for NVMe/FC on Oracle Linux 8.1. Use the ONTAP command output available as part of the NetApp plug-in included in the native nvme-cli.
- SAN booting using the NVMe-oF protocol is currently not supported.

Enable NVMe/FC

1. Install Oracle Linux 8.1 on the server.
2. After the installation is complete, verify that you are running the supported Unbreakable Enterprise kernel. See the Interoperability Matrix Tool.

```
# uname -r
5.4.17-2011.0.7.el8uek.x86_64
```

3. Upgrade the nvme-cli package.

```
# rpm -qa | grep nvmefc
nvmefc-connect-12.6.61.0-1.noarch
```

4. Add the string below as a separate udev rule at /lib/udev/rules.d/71-nvme-iopolicy-netapp-ONTAP.rules. This enables round-robin load balancing for NVMe multipath.

```
# cat /lib/udev/rules.d/71-nvme-iopolicy-netapp-ONTAP.rules
# Enable round-robin for NetApp ONTAP
ACTION=="add", SUBSYSTEM=="nvme-subsystem", ATTR{model}=="NetApp ONTAP
Controller", ATTR{iopolicy}="round-robin"
```

5. On the Oracle Linux 8.1 host, check the host NQN string at /etc/nvme/hostnqn and verify that it matches the host NQN string for the corresponding subsystem on the ONTAP array.

```
# cat /etc/nvme/hostnqn
nqn.2014-08.org.nvmexpress:uuid:75953f3b-77fe-4e03-bf3c-09d5a156fbcd
```

```
*> vserver nvme subsystem host show -vserver vs_nvme_10
Vserver Subsystem Host NQN
------- ---------
----------------------------------------------------------
Oracle Linux_141_nvme_ss_10_0
nqn.2014-08.org.nvmexpress:uuid:75953f3b-77fe-4e03-bf3c-09d5a156fbcd
```

If the hostnqn strings do not match, you should use the vserver modify command to update the host NQN string on your corresponding ONTAP array subsystem to match to host NQN string from etc/nvme/hostnqn on the host.

6. Reboot the host.

Configure the Broadcom FC Adapter for NVMe/FC

1. Verify that you are using the supported adapter. For the current list of supported adapters, see the Interoperability Matrix Tool.

```
# cat /sys/class/scsi_host/host*/modelname
LPe32002-M2
LPe32002-M2
```

```
# cat /sys/class/scsi_host/host*/modeldesc
Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
```

2. NVMe support in lpfc is already enabled by default:

```
# cat /sys/module/lpfc/parameters/lpfc_enable_fc4_type
3
```

Newer lpfc drivers (both inbox and outbox) have lpfc_enable_fc4_type default set to 3. Therefore, you do not need to set this explicitly in the /etc/modprobe.d/lpfc.conf.

3. Next, install the recommended lpfc auto-connect scripts:

```
# rpm -ivh nvmefc-connect-12.6.61.0-1.noarch.rpm
```

4. Verify that the auto-connect scripts are installed.

```
# rpm -qa | grep nvmefc
nvmefc-connect-12.6.61.0-1.noarch
```

5. Verify that the initiator ports are up and running.

```
# cat /sys/class/fc_host/host*/port_name
0x10000090fae0ec61
0x10000090fae0ec62

# cat /sys/class/fc_host/host*/port_state
Online
Online
```

6. Verify that the NVMe/FC initiator ports are enabled and able to see the target ports, and all are up and running.

   In the example below, only a single initiator port has been enabled and connected with two target LIFs as seen in the below output:

```
# cat /sys/class/scsi_host/host*/nvme_info

NVME Initiator Enabled
XRI Dist lpfc0 Total 6144 NVME 2947 SCSI 2947 ELS 250
NVME LPORT lpfc0 WWPN x10000090fae0ec61 WWNN x20000090fae0ec61 DID
x012000 ONLINE
NVME RPORT WWPN x202d00a098c80f09 WWNN x202c00a098c80f09 DID x010201
TARGET DISCSRVC ONLINE
NVME RPORT WWPN x203100a098c80f09 WWNN x202c00a098c80f09 DID x010601
TARGET DISCSRVC ONLINE
```

Validate NVMe/FC

1. Verify the following NVMe/FC settings.

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

```
# cat /sys/class/nvme-subsystem/nvme-subsys*/model
NetApp ONTAP Controller
NetApp ONTAP Controller
```

```
# cat /sys/class/nvme-subsystem/nvme-subsys*/iopolicy
round-robin
round-robin
```

In the above example, two namespaces are mapped to the Oracle Linux 8.1 ANA host. These are visible through four target LIFs: two local node LIFs and two other partner/remote node LIFs. This setup shows as two ANA Optimized and two ANA Inaccessible paths for each namespace on the host.

2. Verify that the namespaces are created.

```
# nvme list
Node                SN                                          Model
Namespace Usage                              Format          FW Rev
-------------------- --------------------------------------
---------------------------------------- ----------------
-------------------------------  ----------------  -------------
/dev/nvme0n1  814vWBNRwfBCAAAAAAAB NetApp ONTAP Controller        2
107.37 GB / 107.37 GB  4 KiB + 0 B   FFFFFFFF
/dev/nvme0n2  814vWBNRwfBCAAAAAAAB NetApp ONTAP Controller        3
107.37 GB / 107.37 GB  4 KiB + 0 B   FFFFFFFF
```

3. Verify the status of the ANA paths.

```
# nvme list-subsys /dev/nvme0n1
nvme-subsys0 - NQN=nqn.1992-
08.com.netapp:sn.5a32407351c711eaaa4800a098df41bd:subsystem.test
\
+- nvme0 fc traddr=nn-0x207300a098dfdd91:pn-0x207400a098dfdd91
host_traddr=nn-0x200000109b1c1204:pn-0x100000109b1c1204 live optimized
+- nvme1 fc traddr=nn-0x207300a098dfdd91:pn-0x207600a098dfdd91
host_traddr=nn-0x200000109b1c1204:pn-0x100000109b1c1204 live
inaccessible
+- nvme2 fc traddr=nn-0x207300a098dfdd91:pn-0x207500a098dfdd91
host_traddr=nn-0x200000109b1c1205:pn-0x100000109b1c1205 live optimized
+- nvme3 fc traddr=nn-0x207300a098dfdd91:pn-0x207700a098dfdd91
host_traddr=nn-0x200000109b1c1205:pn-0x100000109b1c1205 live
inaccessible
```

4. Verify the NetApp plug-in for ONTAP devices.

```
# nvme netapp ontapdevices -o column
Device   Vserver  Namespace Path             NSID   UUID   Size
-------  --------- ------------------------  ------ ----- -----
/dev/nvme0n1   vs_nvme_10       /vol/rhel_141_vol_10_0/ol_157_ns_10_0
1        55baf453-f629-4a18-9364-b6aee3f50dad   53.69GB

# nvme netapp ontapdevices -o json
{
   "ONTAPdevices" : [
   {
        Device" : "/dev/nvme0n1",
        "Vserver" : "vs_nvme_10",
        "Namespace_Path" : "/vol/rhel_141_vol_10_0/ol_157_ns_10_0",
         "NSID" : 1,
         "UUID" : "55baf453-f629-4a18-9364-b6aee3f50dad",
         "Size" : "53.69GB",
         "LBA_Data_Size" : 4096,
         "Namespace_Size" : 13107200
    }
]
```

Enable 1MB I/O size for Broadcom NVMe/FC

ONTAP reports an MDTS (Max Data Transfer Size) of 8 in the Identify Controller data. This means the maximum I/O request size can be up to 1MB. To issue I/O requests of size 1MB for a Broadcom NVMe/FC host, you should increase the `lpfc` value of the `lpfc_sg_seg_cnt` parameter to 256 from the default value of 64.

These steps don't apply to Qlogic NVMe/FC hosts.

Steps

1. Set the `lpfc_sg_seg_cnt` parameter to 256:

```
cat /etc/modprobe.d/lpfc.conf
```

   You should see an output similar to the following example:

```
options lpfc lpfc_sg_seg_cnt=256
```

2. Run the `dracut -f` command, and reboot the host.
3. Verify that the value for `lpfc_sg_seg_cnt` is 256:

```
cat /sys/module/lpfc/parameters/lpfc_sg_seg_cnt
```

Oracle Linux 7

NVMe/FC Host Configuration for Oracle Linux 7.9 with ONTAP

You can configure NVMe over Fibre Channel (NVMe/FC) on initiator hosts running Oracle Linux 7.9 and ONTAP as the target.

Supportability

Beginning with ONTAP 9.6, NVMe/FC is supported for Oracle Linux 7.9. The Oracle Linux 7.9 host can run both NVMe and SCSI traffic through the same fibre channel (FC) initiator adapter ports. See the Hardware Universe for a list of supported FC adapters and controllers. For the current list of supported configurations see the Interoperability Matrix Tool.

You can use the configuration settings provided in this procedure to configure cloud clients connected to Cloud Volumes ONTAP and Amazon FSx for ONTAP.

Known limitations

- Native NVMe/FC auto-connect scripts are not available in the `nvme-cli` package. Use the HBA vendor provided external auto-connect scripts.
- By default, round-robin load balancing is not enabled in NVMe Multipath. You must write a udev rule to enable this functionality. Steps are provided in the section on Enabling NVMe/FC on Oracle Linux 7.9.
- There is no sanlun support for NVMe/FC and, as a consequence, no Linux Host Utilities support for NVMe/FC on Oracle Linux 7.9. Use the ONTAP command output available as part of the NetApp plug-in included in the native nvme-cli.
- SAN booting using the NVMe-oF protocol is currently not supported.

Enable NVMe/FC

1. Install Oracle Linux 7.9 on the server.
2. After the installation is complete, verify that you are running the supported Unbreakable Enterprise kernel. See the Interoperability Matrix Tool.

```
# uname -r
5.4.17-2011.6.2.el7uek.x86_64
```

3. Upgrade the `nvme-cli` package.

```
# rpm -qa | grep nvme-cli
nvme-cli-1.8.1-3.el7.x86_64
```

4. Add the string below as a separate udev rule at `/lib/udev/rules.d/71-nvme-iopolicy-netapp-ONTAP.rules`. This enables round-robin load balancing for NVMe multipath.

```
# cat /lib/udev/rules.d/71-nvme-iopolicy-netapp-ONTAP.rules
# Enable round-robin for NetApp ONTAP
ACTION=="add", SUBSYSTEMS=="nvme-subsystem", ATTRS{model}=="NetApp ONTAP
Controller          ", ATTR{iopolicy}="round-robin"
```

5. On the Oracle Linux L 7.9 host, check the host NQN string at `/etc/nvme/hostnqn` and verify that it matches the host NQN string for the corresponding subsystem on the ONTAP array.

```
# cat /etc/nvme/hostnqn
nqn.2014-08.org.nvmexpress:uuid:497ad959-e6d0-4987-8dc2-a89267400874
```

```
*> vserver nvme subsystem host show -vserver vs_nvme_10
Vserver Subsystem Host NQN
------- --------- -------------------------------------- -----------
ol_157_nvme_ss_10_0
nqn.2014-08.org.nvmexpress:uuid:497ad959-e6d0-4987-8dc2-a89267400874
```

If the `hostnqn` strings do not match, you should use the `vserver modify` command to update the host NQN string on your corresponding ONTAP array subsystem to match to host NQN string from `etc/nvme/hostnqn` on the host.

6. Reboot the host.

Configure the Broadcom FC Adapter for NVMe/FC

1. Verify that you are using the supported adapter. For the current list of supported adapters, see the

Interoperability Matrix Tool.

```
# cat /sys/class/scsi_host/host*/modelname
LPe32002-M2
LPe32002-M2
```

```
# cat /sys/class/scsi_host/host*/modeldesc
Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
```

2. NVMe support in lpfc is already enabled by default:

```
# cat /sys/module/lpfc/parameters/lpfc_enable_fc4_type
3
```

Newer lpfc drivers (both inbox and outbox) have lpfc_enable_fc4_type default set to 3. Therefore, you do not need to set this explicitly in the `/etc/modprobe.d/lpfc.conf`.

3. Next, install the recommended lpfc auto-connect scripts:

```
# rpm -ivh nvmefc-connect-12.8.264.0-1.noarch.rpm
```

4. Verify that the auto-connect scripts are installed.

```
# rpm -qa | grep nvmefc
nvmefc-connect-12.8.264.0-1.noarch
```

5. Verify that the initiator ports are up and running.

```
# cat /sys/class/fc_host/host*/port_name
0x10000090fae0ec61
0x10000090fae0ec62

# cat /sys/class/fc_host/host*/port_state
Online
Online
```

6. Verify that the NVMe/FC initiator ports are enabled and able to see the target ports, and all are up and running.

In the example below, only a single initiator port has been enabled and connected with two target LIFs as seen in the below output:

```
# cat /sys/class/scsi_host/host*/nvme_info

NVME Initiator Enabled
XRI Dist lpfc0 Total 6144 NVME 2947 SCSI 2947 ELS 250
NVME LPORT lpfc0 WWPN x10000090fae0ec61 WWNN x20000090fae0ec61 DID
x012000 ONLINE
NVME RPORT WWPN x202d00a098c80f09 WWNN x202c00a098c80f09 DID x010201
TARGET DISCSRVC ONLINE
NVME RPORT WWPN x203100a098c80f09 WWNN x202c00a098c80f09 DID x010601
TARGET DISCSRVC ONLINE
```

Validate NVMe/FC

1. Verify the following NVMe/FC settings.

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

```
# cat /sys/class/nvme-subsystem/nvme-subsys*/model
NetApp ONTAP Controller
NetApp ONTAP Controller
```

```
# cat /sys/class/nvme-subsystem/nvme-subsys*/iopolicy
round-robin
round-robin
```

In the above example, two namespaces are mapped to the Oracle Linux 7.9 ANA host. These are visible through four target LIFs: two local node LIFs and two other partner/remote node LIFs. This setup shows as two ANA Optimized and two ANA Inaccessible paths for each namespace on the host.

2. Verify that the namespaces are created.

```
# nvme list
Node SN Model Namespace Usage Format FW Rev
---------------- -------------------- -----------------------
/dev/nvme0n1 80BADBKnB/JvAAAAAAAC NetApp ONTAP Controller 1 53.69 GB /
53.69 GB 4 KiB + 0 B FFFFFFFF
```

3. Verify the status of the ANA paths.

```
# nvme list-subsys/dev/nvme0n1
Nvme-subsysf0 – NQN=nqn.1992-
08.com.netapp:sn.341541339b9511e8a9b500a098c80f09:subsystem.ol_157_nvme_
ss_10_0
\
+- nvme0 fc traddr=nn-0x202c00a098c80f09:pn-0x202d00a098c80f09
host_traddr=nn-0x20000090fae0ec61:pn-0x10000090fae0ec61 live optimized
+- nvme1 fc traddr=nn-0x207300a098dfdd91:pn-0x207600a098dfdd91
host_traddr=nn-0x200000109b1c1204:pn-0x100000109b1c1204 live
inaccessible
+- nvme2 fc traddr=nn-0x207300a098dfdd91:pn-0x207500a098dfdd91
host_traddr=nn-0x200000109b1c1205:pn-0x100000109b1c1205 live optimized
+- nvme3 fc traddr=nn-0x207300a098dfdd91:pn-0x207700a098dfdd91 host
traddr=nn-0x200000109b1c1205:pn-0x100000109b1c1205 live inaccessible
```

4. Verify the NetApp plug-in for ONTAP devices.

```
# nvme netapp ontapdevices -o column
Device   Vserver  Namespace Path             NSID   UUID   Size
-------  -------- ------------------------  ------ ----- -----
/dev/nvme0n1   vs_nvme_10       /vol/rhel_141_vol_10_0/ol_157_ns_10_0
1        55baf453-f629-4a18-9364-b6aee3f50dad   53.69GB

# nvme netapp ontapdevices -o json
{
   "ONTAPdevices" : [
   {
        Device" : "/dev/nvme0n1",
        "Vserver" : "vs_nvme_10",
        "Namespace_Path" : "/vol/rhel_141_vol_10_0/ol_157_ns_10_0",
         "NSID" : 1,
         "UUID" : "55baf453-f629-4a18-9364-b6aee3f50dad",
         "Size" : "53.69GB",
         "LBA_Data_Size" : 4096,
         "Namespace_Size" : 13107200
    }
]
```

Enable 1MB I/O size for Broadcom NVMe/FC

ONTAP reports an MDTS (Max Data Transfer Size) of 8 in the Identify Controller data. This means the maximum I/O request size can be up to 1MB. To issue I/O requests of size 1MB for a Broadcom NVMe/FC host, you should increase the `lpfc` value of the `lpfc_sg_seg_cnt` parameter to 256 from the default value of 64.

These steps don’t apply to Qlogic NVMe/FC hosts.

Steps

1. Set the `lpfc_sg_seg_cnt` parameter to 256:

```
cat /etc/modprobe.d/lpfc.conf
```

   You should see an output similar to the following example:

```
options lpfc lpfc_sg_seg_cnt=256
```

2. Run the `dracut -f` command, and reboot the host.
3. Verify that the value for `lpfc_sg_seg_cnt` is 256:

```
cat /sys/module/lpfc/parameters/lpfc_sg_seg_cnt
```

NVMe/FC Host Configuration for Oracle Linux 7.8 with ONTAP

You can configure NVMe over Fibre Channel (NVMe/FC) on initiator hosts running Oracle Linux 7.8 and ONTAP as the target.

Supportability

Beginning with ONTAP 9.6, NVMe/FC is supported for Oracle Linux 7.8. The Oracle Linux 7.8 host can run both NVMe and SCSI traffic through the same fibre channel (FC) initiator adapter ports. Note that the Broadcom initiator can serve both NVMe/FC and FCP traffic through the same FC adapter ports. See the Hardware Universe for a list of supported FC adapters and controllers. For the current list of supported configurations see the Interoperability Matrix Tool.

You can use the configuration settings provided in this procedure to configure cloud clients connected to Cloud Volumes ONTAP and Amazon FSx for ONTAP.

Known limitations

- Native NVMe/FC auto-connect scripts are not available in the nvme-cli package. Use the HBA vendor provided external auto-connect scripts.
- By default, round-robin load balancing is not enabled in NVMe Multipath. You must write a udev rule to enable this functionality. Steps are provided in the section on Enabling NVMe/FC on Oracle Linux 7.8.
- There is no sanlun support for NVMe/FC and, as a consequence, no Linux Host Utilities support for NVMe/FC on Oracle Linux 7.8. Use the ONTAP command output available as part of the NetApp plug-in included in the native nvme-cli.
- SAN booting using the NVMe-oF protocol is currently not supported.

Enabling NVMe/FC

1. Install Oracle Linux 7.8 on the server.
2. After the installation is complete, verify that you are running the supported Unbreakable Enterprise kernel. See the Interoperability Matrix Tool.

```
# uname -r
4.14.35-1902.9.2.el7uek
```

3. Upgrade the nvme-cli package.

```
# rpm -qa | grep nvme-cli
nvme-cli-1.8.1-3.el7.x86_64
```

4. Add the string below as a separate udev rule at /lib/udev/rules.d/71-nvme-iopolicy-netapp-ONTAP.rules. This enables round-robin load balancing for NVMe multipath.

```
# cat /lib/udev/rules.d/71-nvme-iopolicy-netapp-ONTAP.rules
# Enable round-robin for NetApp ONTAP
ACTION=="add", SUBSYSTEM=="nvme-subsystem", ATTR{model}=="NetApp ONTAP
Controller", ATTR{iopolicy}="round-robin"
```

5. On the Oracle Linux L 7.8 host, check the host NQN string at /etc/nvme/hostnqn and verify that it matches the host NQN string for the corresponding subsystem on the ONTAP array.

```
# cat /etc/nvme/hostnqn
nqn.2014-08.org.nvmexpress:uuid:75953f3b-77fe-4e03-bf3c-09d5a156fbcd
```

```
*> vserver nvme subsystem host show -vserver vs_nvme_10
Vserver Subsystem Host NQN
------- --------- -------------------------------------- -----------
ol_157_nvme_ss_10_0
nqn.2014-08.org.nvmexpress:uuid:75953f3b-77fe-4e03-bf3c-09d5a156fbcd
```

   If the hostnqn strings do not match, you should use the vserver modify command to update the host NQN string on your corresponding ONTAP array subsystem to match to host NQN string from etc/nvme/hostnqn on the host.

6. Reboot the host.

Configuring the Broadcom FC Adapter for NVMe/FC

1. Verify that you are using the supported adapter. For the current list of supported adapters, see the Interoperability Matrix Tool.

```
# cat /sys/class/scsi_host/host*/modelname
LPe32002-M2
LPe32002-M2
```

```
# cat /sys/class/scsi_host/host*/modeldesc
Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
```

2. NVMe support in lpfc is already enabled by default:

```
# cat /sys/module/lpfc/parameters/lpfc_enable_fc4_type
3
```

Newer lpfc drivers (both inbox and outbox) have lpfc_enable_fc4_type default set to 3. Therefore, you do not need to set this explicitly in the /etc/modprobe.d/lpfc.conf.

3. Next, install the recommended lpfc auto-connect scripts:

```
# rpm -ivh nvmefc-connect-12.4.65.0-1.noarch.rpm
```

4. Verify that the auto-connect scripts are installed.

```
# rpm -qa | grep nvmefc
nvmefc-connect-12.4.65.0-1.noarch
```

5. Verify that the initiator ports are up and running.

```
# cat /sys/class/fc_host/host*/port_name
0x10000090fae0ec61
0x10000090fae0ec62

# cat /sys/class/fc_host/host*/port_state
Online
Online
```

6. Verify that the NVMe/FC initiator ports are enabled and able to see the target ports, and all are up and running.

   In the example below, only a single initiator port has been enabled and connected with two target LIFs as seen in the below output:

```
# cat /sys/class/scsi_host/host*/nvme_info

NVME Initiator Enabled
XRI Dist lpfc0 Total 6144 NVME 2947 SCSI 2947 ELS 250
NVME LPORT lpfc0 WWPN x10000090fae0ec61 WWNN x20000090fae0ec61 DID
x012000 ONLINE
NVME RPORT WWPN x202d00a098c80f09 WWNN x202c00a098c80f09 DID x010201
TARGET DISCSRVC ONLINE
NVME RPORT WWPN x203100a098c80f09 WWNN x202c00a098c80f09 DID x010601
TARGET DISCSRVC ONLINE
```

Validating NVMe/FC

1. Verify the following NVMe/FC settings.

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

```
# cat /sys/class/nvme-subsystem/nvme-subsys*/model
NetApp ONTAP Controller
NetApp ONTAP Controller
```

```
# cat /sys/class/nvme-subsystem/nvme-subsys*/iopolicy
round-robin
round-robin
```

In the above example, two namespaces are mapped to the Oracle Linux 7.8 ANA host. These are visible through four target LIFs: two local node LIFs and two other partner/remote node LIFs. This setup shows as two ANA Optimized and two ANA Inaccessible paths for each namespace on the host.

2. Verify that the namespaces are created.

```
# nvme list
Node SN Model Namespace Usage Format FW Rev
---------------- -------------------- -----------------------
/dev/nvme0n1 80BADBKnB/JvAAAAAAAC NetApp ONTAP Controller 1 53.69 GB /
53.69 GB 4 KiB + 0 B FFFFFFFF
```

3. Verify the status of the ANA paths.

```
# nvme list-subsys/dev/nvme0n1
Nvme-subsysf0 – NQN=nqn.1992-
08.com.netapp:sn.341541339b9511e8a9b500a098c80f09:subsystem.ol_157_nvme_
ss_10_0
\
+- nvme0 fc traddr=nn-0x202c00a098c80f09:pn-0x202d00a098c80f09
host_traddr=nn-0x20000090fae0ec61:pn-0x10000090fae0ec61 live optimized
+- nvme1 fc traddr=nn-0x207300a098dfdd91:pn-0x207600a098dfdd91
host_traddr=nn-0x200000109b1c1204:pn-0x100000109b1c1204 live
inaccessible
+- nvme2 fc traddr=nn-0x207300a098dfdd91:pn-0x207500a098dfdd91
host_traddr=nn-0x200000109b1c1205:pn-0x100000109b1c1205 live optimized
+- nvme3 fc traddr=nn-0x207300a098dfdd91:pn-0x207700a098dfdd91 host
traddr=nn-0x200000109b1c1205:pn-0x100000109b1c1205 live inaccessible
```

4. Verify the NetApp plug-in for ONTAP devices.

```
# nvme netapp ontapdevices -o column
Device   Vserver  Namespace Path             NSID   UUID   Size
-------  --------- ------------------------  ------ ----- -----
/dev/nvme0n1   vs_nvme_10       /vol/rhel_141_vol_10_0/ol_157_ns_10_0
1        55baf453-f629-4a18-9364-b6aee3f50dad   53.69GB

# nvme netapp ontapdevices -o json
{
   "ONTAPdevices" : [
   {
        Device" : "/dev/nvme0n1",
        "Vserver" : "vs_nvme_10",
        "Namespace_Path" : "/vol/rhel_141_vol_10_0/ol_157_ns_10_0",
         "NSID" : 1,
         "UUID" : "55baf453-f629-4a18-9364-b6aee3f50dad",
         "Size" : "53.69GB",
         "LBA_Data_Size" : 4096,
         "Namespace_Size" : 13107200
    }
]
```

Enabling 1MB I/O size for Broadcom NVMe/FC

ONTAP reports an MDTS (Max Data Transfer Size) of 8 in the Identify Controller data. This means the maximum I/O request size can be up to 1MB. To issue I/O requests of size 1MB for a Broadcom NVMe/FC host, you should increase the `lpfc` value of the `lpfc_sg_seg_cnt` parameter to 256 from the default value of 64.

These steps don’t apply to Qlogic NVMe/FC hosts.

Steps

1. Set the `lpfc_sg_seg_cnt` parameter to 256:

```
cat /etc/modprobe.d/lpfc.conf
```

   You should see an output similar to the following example:

```
options lpfc lpfc_sg_seg_cnt=256
```

2. Run the `dracut -f` command, and reboot the host.
3. Verify that the value for `lpfc_sg_seg_cnt` is 256:

```
cat /sys/module/lpfc/parameters/lpfc_sg_seg_cnt
```

NVMe/FC Host Configuration for Oracle Linux 7.7 with ONTAP

You can configure NVMe over Fibre Channel (NVMe/FC) on initiator hosts running Oracle Linux 7.7 and ONTAP as the target.

Supportability

Beginning with ONTAP 9.6, NVMe/FC is supported for Oracle Linux 7.7. The Oracle Linux 7.7 host can run both NVMe and SCSI traffic through the same fibre channel initiator adapter ports. See the Hardware Universe for a list of supported FC adapters and controllers. For the current list of supported configurations see the Interoperability Matrix Tool.

You can use the configuration settings provided in this procedure to configure cloud clients connected to Cloud Volumes ONTAP and Amazon FSx for ONTAP.

Known limitations

- Native NVMe/FC auto-connect scripts are not available in the nvme-cli package.
  You can use the HBA vendor provided external auto-connect scripts.
- By default, round-robin load balancing is not enabled.
  You must write a udev rule to enable this functionality. Steps are provided in the section on Enabling NVMe/FC on OL 7.7.
- SAN booting using the NVMe-oF protocol is currently not supported.

Enabling NVMe on OL 7.7

1. Ensure the default Oracle Linux 7.7 kernel is installed.
2. Reboot the host and verify that it boots into specified OL 7.7 kernel.

```
# uname -r
4.14.35-1902.9.2.el7uek
```

3. Upgrade to the nvme-cli-1.8.1-3.el7 package.

```
# rpm -qa|grep nvme-cli
nvme-cli-1.8.1-3.el7.x86_64
```

4. Add the string below as a separate udev rule at `/lib/udev/rules.d/71-nvme-iopolicy-netapp-ONTAP.rules`. This enables round-robin load balancing for NVMe multipath.

```
# Enable round-robin for NetApp ONTAP
ACTION==”add”, SUBSYSTEM==”nvme-subsystem”, ATTR{model}==”NetApp ONTAP
Controller”, ATTR{iopolicy}=”round-robin
```

5. On the OL 7.7 host, check the host NQN string at `/etc/nvme/hostnqn` and verify that it matches the host NQN string for the corresponding subsystem on the ONTAP array.

```
# cat /etc/nvme/hostnqn
nqn.2014-08.org.nvmexpress:uuid:75953f3b-77fe-4e03-bf3c-09d5a156fbcd
```

```
*> vserver nvme subsystem host show -vserver vs_nvme_10
Vserver Subsystem Host NQN
------- --------- -------------------------------------- -----------
ol_157_nvme_ss_10_0
nqn.2014-08.org.nvmexpress:uuid:75953f3b-77fe-4e03-bf3c-09d5a156fbcd
```

> If the host NQN strings do not match, you should use the vserver modify command to update the host NQN string on your corresponding ONTAP array subsystem to match to host NQN string from `/etc/nvme/hostnqn` on the host.

1. Reboot the host.

Configuring the Broadcom FC Adapter for NVMe/FC

1. Verify that you are using a supported adapter. For the current list of supported adapters see the Interoperability Matrix Tool.

```
# cat /sys/class/scsi_host/host*/modelname
LPe32002-M2
LPe32002-M2
```

```
# cat /sys/class/scsi_host/host*/modeldesc
Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
```

2. Copy and install the Broadcom outbox auto-connect scripts package.

```
# rpm -ivh nvmefc-connect-12.4.65.0-1.noarch.rpm
```

3. Reboot the host.
4. Verify that you are using the recommended Broadcom lpfc firmware, native inbox driver and outbox auto-connect package versions. For a list of supported versions, see the Interoperability Matrix Tool.

```
# cat /sys/class/scsi_host/host*/fwrev
12.4.243.17, sil-4.2.c
12.4.243.17, sil-4.2.c

# cat /sys/module/lpfc/version
0:12.0.0.10

# rpm -qa | grep nvmefc
nvmefc-connect-12.4.65.0-1.noarch
```

5. Verify that lpfc_enable_fc4_type is set to 3.

```
# cat /sys/module/lpfc/parameters/lpfc_enable_fc4_type
3
```

6. Verify that the initiator ports are up and running.

```
# cat /sys/class/fc_host/host*/port_name
0x10000090fae0ec61
0x10000090fae0ec62
```

```
# cat /sys/class/fc_host/host*/port_state
Online
Online
```

7. Verify that the NVMe/FC initiator ports are enabled, running and able to see the target LIFs.

```
# cat /sys/class/scsi_host/host*/nvme_info
NVME Initiator Enabled
XRI Dist lpfc0 Total 6144 NVME 2947 SCSI 2977 ELS 250
NVME LPORT lpfc0 WWPN x10000090fae0ec61 WWNN x20000090fae0ec61 DID
x012000 ONLINE
NVME RPORT WWPN x202d00a098c80f09 WWNN x202c00a098c80f09 DID x010201
TARGET DISCSRVC ONLINE
NVME RPORT WWPN x203100a098c80f09 WWNN x202c00a098c80f09 DID x010601
TARGET DISCSRVC ONLINE
NVME Statistics
…
```

Validating NVMe/FC

1. Verify the following NVMe/FC settings.

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

# cat /sys/class/nvme-subsystem/nvme-subsys*/model
NetApp ONTAP Controller
NetApp ONTAP Controller

# cat /sys/class/nvme-subsystem/nvme-subsys*/iopolicy
round-robin
round-robin
```

2. Verify that the namespaces are created.

```
# nvme list
Node SN Model Namespace Usage Format FW Rev
---------------- -------------------- -----------------------
/dev/nvme0n1 80BADBKnB/JvAAAAAAAC NetApp ONTAP Controller 1 53.69 GB /
53.69 GB 4 KiB + 0 B FFFFFFFF
```

3. Verify the status of the ANA paths.

```
# nvme list-subsys/dev/nvme0n1
Nvme-subsysf0 – NQN=nqn.1992-
08.com.netapp:sn.341541339b9511e8a9b500a098c80f09:subsystem.ol_157_nvme_
ss_10_0
\
+- nvme0 fc traddr=nn-0x202c00a098c80f09:pn-0x202d00a098c80f09
host_traddr=nn-0x20000090fae0ec61:pn-0x10000090fae0ec61 live optimized
+- nvme1 fc traddr=nn-0x207300a098dfdd91:pn-0x207600a098dfdd91
host_traddr=nn-0x200000109b1c1204:pn-0x100000109b1c1204 live
inaccessible
+- nvme2 fc traddr=nn-0x207300a098dfdd91:pn-0x207500a098dfdd91
host_traddr=nn-0x200000109b1c1205:pn-0x100000109b1c1205 live optimized
+- nvme3 fc traddr=nn-0x207300a098dfdd91:pn-0x207700a098dfdd91 host
traddr=nn-0x200000109b1c1205:pn-0x100000109b1c1205 live inaccessible
```

4. Verify the NetApp plug-in for ONTAP devices.

```
# nvme netapp ontapdevices -o column
Device   Vserver  Namespace Path             NSID   UUID   Size
-------  -------- ------------------------  ------ ----- -----
/dev/nvme0n1   vs_nvme_10       /vol/rhel_141_vol_10_0/ol_157_ns_10_0
1        55baf453-f629-4a18-9364-b6aee3f50dad   53.69GB

# nvme netapp ontapdevices -o json
{
   "ONTAPdevices" : [
   {
        Device" : "/dev/nvme0n1",
        "Vserver" : "vs_nvme_10",
        "Namespace_Path" : "/vol/rhel_141_vol_10_0/ol_157_ns_10_0",
         "NSID" : 1,
         "UUID" : "55baf453-f629-4a18-9364-b6aee3f50dad",
         "Size" : "53.69GB",
         "LBA_Data_Size" : 4096,
         "Namespace_Size" : 13107200
    }
]
```

Enabling 1MB I/O size for Broadcom NVMe/FC

ONTAP reports an MDTS (Max Data Transfer Size) of 8 in the Identify Controller data. This means the maximum I/O request size can be up to 1MB. To issue I/O requests of size 1MB for a Broadcom NVMe/FC host, you should increase the `lpfc` value of the `lpfc_sg_seg_cnt` parameter to 256 from the default value of 64.

These steps don't apply to Qlogic NVMe/FC hosts.

Steps

1. Set the `lpfc_sg_seg_cnt` parameter to 256:

```
cat /etc/modprobe.d/lpfc.conf
```

You should see an output similar to the following example:

```
options lpfc lpfc_sg_seg_cnt=256
```

2. Run the `dracut -f` command, and reboot the host.
3. Verify that the value for `lpfc_sg_seg_cnt` is 256:

```
cat /sys/module/lpfc/parameters/lpfc_sg_seg_cnt
```

LPFC verbose logging

Set the lpfc driver for NVMe/FC.

Steps

1. Set the `lpfc_log_verbose` driver setting to any of the following values to log NVMe/FC events.

```
#define LOG_NVME 0x00100000 /* NVME general events. */
#define LOG_NVME_DISC 0x00200000 /* NVME Discovery/Connect events. */
#define LOG_NVME_ABTS 0x00400000 /* NVME ABTS events. */
#define LOG_NVME_IOERR 0x00800000 /* NVME IO Error events. */
```

2. After setting the values, run the `dracut-f` command and reboot the host.
3. Verify the settings.

```
# cat /etc/modprobe.d/lpfc.conf options lpfc lpfc_log_verbose=0xf00083

# cat /sys/module/lpfc/parameters/lpfc_log_verbose 15728771
```

RHEL

RHEL 10

NVMe-oF host configuration for RHEL 10 with ONTAP

NetApp SAN host configurations support the NVMe over Fabrics (NVMe-oF) protocol with Asymmetric Namespace Access (ANA). In NVMe-oF environments, ANA is equivalent to asymmetric logical unit access (ALUA) multipathing in iSCSI and FCP environments. ANA is implemented using the in-kernel NVMe multipath feature.

About this task

You can use the following support and features with the NVMe-oF host configuration for Red Hat Enterprise Linux (RHEL) 10. You should also review the known limitations before starting the configuration process.

- Support available:
  - Support for NVMe over TCP (NVMe/TCP) in addition to NVMe over Fibre Channel (NVMe/FC). The NetApp plug-in in the native `nvme-cli` package displays ONTAP details for both NVMe/FC and NVMe/TCP namespaces.
  - Running both NVMe and SCSI traffic on the same host. For example, you can configure dm-multipath for SCSI mpath devices on SCSI LUNs and use NVMe multipath to configure NVMe-oF namespace devices on the host.
  - Beginning with ONTAP 9.12.1, support for secure in-band authentication is introduced for NVMe/TCP. You can use secure in-band authentication for NVMe/TCP with RHEL 10.

  For additional details on supported configurations, see the Interoperability Matrix Tool.

- Features available:
  - Beginning with RHEL 10, native NVMe multipathing is always enabled, and DM multipath support for NVMe-oF is not supported.
- Known limitations:
  - Avoid issuing the `nvme disconnect-all` command on systems booting from SAN over NVMe-TCP or NVMe-FC namespaces because it disconnects both root and data filesystems and might lead to system instability.

Step 1: Optionally, enable SAN booting

You can configure your host to use SAN booting to simplify deployment and improve scalability.

Before you begin

Use the Interoperability Matrix Tool to verify that your Linux OS, host bus adapter (HBA), HBA firmware, HBA boot BIOS, and ONTAP version support SAN booting.

Steps

1. Create a SAN boot namespace and map it to the host.
2. Enable SAN booting in the server BIOS for the ports to which the SAN boot namespace is mapped.

   For information on how to enable the HBA BIOS, see your vendor-specific documentation.

3. Verify that the configuration was successful by rebooting the host and verifying that the OS is up and running.

Step 2: Validate software versions

Use the following procedure to validate the minimum supported RHEL 10 software versions.

Steps

1. Install RHEL 10 on the server. After the installation is complete, verify that you are running the specified RHEL 10 kernel:

   ```
   uname -r
   ```

   The following example shows a RHEL kernel version:

   ```
   6.12.0-55.9.1.el10_0.x86_64
   ```

2. Install the `nvme-cli` package:

   ```
   rpm -qa|grep nvme-cli
   ```

   The following example shows an `nvme-cli` package version:

   ```
   nvme-cli-2.11-5.el10.x86_64
   ```

3. Install the `libnvme` package:

   ```
   rpm -qa|grep libnvme
   ```

   The following example shows an `libnvme` package version:

   ```
   libnvme-1.11.1-1.el10.x86_64
   ```

4. On the host, check the hostnqn string at `/etc/nvme/hostnqn`:

   ```
   cat /etc/nvme/hostnqn
   ```

   The following example shows an `hostnqn` version:

   ```
   nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
   ```

5. Verify that the `hostnqn` string matches the `hostnqn` string for the corresponding subsystem on the ONTAP array:

```
::> vserver nvme subsystem host show -vserver vs_nvme_194_rhel10
```

Show example

```
Vserver Subsystem Priority  Host NQN
------- --------- --------
------------------------------------------------
vs_ nvme_194_rhel10
        nvme4
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048- c7c04f425633
        nvme_1
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048- c7c04f425633
        nvme_2
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048- c7c04f425633
        nvme_3
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048- c7c04f425633
4 entries were displayed.
```

If the `hostnqn` strings do not match, use the `vserver modify` command to update the `hostnqn` string on your corresponding ONTAP array subsystem to match the `hostnqn` string from `/etc/nvme/hostnqn` on the host.

Step 3: Configure NVMe/FC

You can configure NVMe/FC with Broadcom/Emulex FC or Marvell/Qlogic FC adapters. For NVMe/FC configured with a Broadcom adapter, you can enable I/O requests of size 1MB.

Broadcom/Emulex

Configure NVMe/FC for a Broadcom/Emulex adapter.

Steps

1. Verify that you are using the supported adapter model:

   a. Display the model names:

   ```
   cat /sys/class/scsi_host/host*/modelname
   ```

   You should see the following output:

   ```
   LPe36002-M64
   LPe36002-M64
   ```

   b. Display the model descriptions:

   ```
   cat /sys/class/scsi_host/host*/modeldesc
   ```

   You should see an output similar to the following example:

   ```
   Emulex LightPulse LPe36002-M64 2-Port 64Gb Fibre Channel Adapter
   Emulex LightPulse LPe36002-M64 2-Port 64Gb Fibre Channel Adapter
   ```

2. Verify that you are using the recommended Broadcom `lpfc` firmware and inbox driver:

   a. Display the firmware version:

   ```
   cat /sys/class/scsi_host/host*/fwrev
   ```

   The following example shows firmware versions:

   ```
   14.0.539.16, sli-4:6:d
   14.0.539.16, sli-4:6:d
   ```

   b. Display the inbox driver version:

   ```
   cat /sys/module/lpfc/version
   ```

   The following example shows a driver version:

```
0:14.4.0.6
```

For the current list of supported adapter driver and firmware versions, see the Interoperability Matrix Tool.

3. Verify that the expected output of `lpfc_enable_fc4_type` is set to `3`:

```
cat /sys/module/lpfc/parameters/lpfc_enable_fc4_type
```

4. Verify that you can view your initiator ports:

```
cat /sys/class/fc_host/host*/port_name
```

The following example shows port identities:

```
0x2100f4c7aa0cd7c2
0x2100f4c7aa0cd7c3
```

5. Verify that your initiator ports are online:

```
cat /sys/class/fc_host/host*/port_state
```

You should see the following output:

```
Online
Online
```

6. Verify that the NVMe/FC initiator ports are enabled and that the target ports are visible:

```
cat /sys/class/scsi_host/host*/nvme_info
```

Show example

```
NVME Initiator Enabled
XRI Dist lpfc2 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc2 WWPN x100000109bf044b1 WWNN x200000109bf044b1
DID x022a00 ONLINE
NVME RPORT       WWPN x202fd039eaa7dfc8 WWNN x202cd039eaa7dfc8
DID x021310 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x202dd039eaa7dfc8 WWNN x202cd039eaa7dfc8
DID x020b10 TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 0000000810 Cmpl 0000000810 Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000007b098f07 Issue 000000007aee27c4 OutIO
fffffffffe498bd
        abort 000013b4 noxri 00000000 nondlp 00000058 qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 000013b4 Err 00021443

NVME Initiator Enabled
XRI Dist lpfc3 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc3 WWPN x100000109bf044b2 WWNN x200000109bf044b2
DID x021b00 ONLINE
NVME RPORT       WWPN x2033d039eaa7dfc8 WWNN x202cd039eaa7dfc8
DID x020110 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2032d039eaa7dfc8 WWNN x202cd039eaa7dfc8
DID x022910 TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 0000000840 Cmpl 0000000840 Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000007afd4434 Issue 000000007ae31b83 OutIO
fffffffffe5d74f
        abort 000014a5 noxri 00000000 nondlp 0000006a qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 000014a5 Err 0002149a
```

Marvell/QLogic

Configure NVMe/FC for a Marvell/QLogic adapter.

Steps

1. Verify that you are running the supported adapter driver and firmware versions:

```
cat /sys/class/fc_host/host*/symbolic_name
```

The follow example shows driver and firmware versions:

```
QLE2872 FW:v9.15.00 DVR:v10.02.09.300-k
QLE2872 FW:v9.15.00 DVR:v10.02.09.300-k
```

2. Verify that `ql2xnvmeenable` is set. This enables the Marvell adapter to function as an NVMe/FC initiator:

```
cat /sys/module/qla2xxx/parameters/ql2xnvmeenable
```

The expected ouptut is 1.

Step 4: Optionally, enable 1MB I/O

ONTAP reports an MDTS (Max Data Transfer Size) of 8 in the Identify Controller data. This means the maximum I/O request size can be up to 1MB. To issue I/O requests of size 1MB for a Broadcom NVMe/FC host, you should increase the `lpfc` value of the `lpfc_sg_seg_cnt` parameter to 256 from the default value of 64.

> These steps don't apply to Qlogic NVMe/FC hosts.

Steps

1. Set the `lpfc_sg_seg_cnt` parameter to 256:

```
cat /etc/modprobe.d/lpfc.conf
```

You should see an output similar to the following example:

```
options lpfc lpfc_sg_seg_cnt=256
```

2. Run the `dracut -f` command, and reboot the host.
3. Verify that the value for `lpfc_sg_seg_cnt` is 256:

```
cat /sys/module/lpfc/parameters/lpfc_sg_seg_cnt
```

Step 5: Verify NVMe boot services

With RHEL 10, the `nvmefc-boot-connections.service` and `nvmf-autoconnect.service` boot services included in the NVMe/FC `nvme-cli` package are automatically enabled when the system boots.

After booting completes, verify that the `nvmefc-boot-connections.service` and `nvmf-autoconnect.service` boot services are enabled.

Steps

1. Verify that `nvmf-autoconnect.service` is enabled:

   ```
   systemctl status nvmf-autoconnect.service
   ```

   Show example output

   ```
   nvmf-autoconnect.service - Connect NVMe-oF subsystems automatically
   during boot
        Loaded: loaded (/usr/lib/systemd/system/nvmf-
   autoconnect.service; enabled; preset: disabled)
        Active: inactive (dead)

   Jun 10 04:06:26 SR630-13-201.lab.eng.btc.netapp.in systemd[1]:
   Starting Connect NVMe-oF subsystems automatically during boot...
   Jun 10 04:06:26 SR630-13-201.lab.eng.btc.netapp.in systemd[1]: nvmf-
   autoconnect.service: Deactivated successfully.
   Jun 10 04:06:26 SR630-13-201.lab.eng.btc.netapp.in systemd[1]:
   Finished Connect NVMe-oF subsystems automatically during boot.
   ```

2. Verify that `nvmefc-boot-connections.service` is enabled:

   ```
   systemctl status nvmefc-boot-connections.service
   ```

Show example output

```
nvmefc-boot-connections.service - Auto-connect to subsystems on FC-
NVME devices found during boot
     Loaded: loaded (/usr/lib/systemd/system/nvmefc-boot-
connections.service; enabled; preset: enabled)
     Active: inactive (dead) since Tue 2025-06-10 01:08:36 EDT; 2h
59min ago
   Main PID: 7090 (code=exited, status=0/SUCCESS)
        CPU: 30ms

Jun 10 01:08:36 localhost systemd[1]: Starting Auto-connect to
subsystems on FC-NVME devices found during boot...
Jun 10 01:08:36 localhost systemd[1]: nvmefc-boot-
connections.service: Deactivated successfully.
Jun 10 01:08:36 localhost systemd[1]: Finished Auto-connect to
subsystems on FC-NVME devices found during boot.
```

Step 6: Configure NVMe/TCP

The NVMe/TCP protocol doesn't support the `auto-connect` operation. Instead, you can discover the NVMe/TCP subsystems and namespaces by performing the NVMe/TCP `connect` or `connect-all` operations manually.

Steps

1. Verify that the initiator port can fetch the discovery log page data across the supported NVMe/TCP LIFs:

```
nvme discover -t tcp -w host-traddr -a traddr
```

Show example

```
nvme discover -t tcp -w 192.168.20.1 -a 192.168.20.20

Discovery Log Number of Records 8, Generation counter 18
=====Discovery Log Entry 0======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  4
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.64e65e6caae711ef9668d039ea951c46:discovery
traddr:  192.168.21.21
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 1======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  2
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.64e65e6caae711ef9668d039ea951c46:discovery
traddr:  192.168.20.21
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 2======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  3
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.64e65e6caae711ef9668d039ea951c46:discovery
traddr:  192.168.21.20
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 3======
trtype:  tcp
```

```
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  1
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.64e65e6caae711ef9668d039ea951c46:discovery
traddr:  192.168.20.20
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 4======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.64e65e6caae711ef9668d039ea951c46:subsystem.rhel10_t
cp_subsystem
traddr:  192.168.21.21
eflags:  none
sectype: none
=====Discovery Log Entry 5======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.64e65e6caae711ef9668d039ea951c46:subsystem.rhel10_t
cp_subsystem
traddr:  192.168.20.21
eflags:  none
sectype: none
=====Discovery Log Entry 6======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  3
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.64e65e6caae711ef9668d039ea951c46:subsystem.rhel10_t
```

```
cp_subsystem
traddr:  192.168.21.20
eflags:  none
sectype: none
=====Discovery Log Entry 7======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.64e65e6caae711ef9668d039ea951c46:subsystem.rhel10_t
cp_subsystem
traddr:  192.168.20.20
eflags:  none
sectype: none
```

2. Verify that the other NVMe/TCP initiator-target LIF combinations can successfully fetch discovery log page data:

```
nvme discover -t tcp -w host-traddr -a traddr
```

Show example

```
nvme discover -t tcp -w 192.168.20.1 -a 192.168.20.20
nvme discover -t tcp -w 192.168.21.1 -a 192.168.21.20
nvme discover -t tcp -w 192.168.20.1 -a 192.168.20.21
nvme discover -t tcp -w 192.168.21.1 -a 192.168.21.21
```

3. Run the `nvme connect-all` command across all the supported NVMe/TCP initiator-target LIFs across the nodes:

```
nvme connect-all -t tcp -w host-traddr -a traddr
```

Show example

```
nvme    connect-all -t  tcp -w  192.168.20.1    -a  192.168.20.20
nvme    connect-all -t  tcp -w  192.168.21.1    -a  192.168.21.20
nvme    connect-all -t  tcp -w  192.168.20.1    -a  192.168.20.21
nvme    connect-all -t  tcp -w  192.168.21.1    -a  192.168.21.21
```

> Beginning with RHEL 9.4, the setting for the NVMe/TCP `ctrl_loss_tmo timeout` is automatically set to "off". As a result:
>
> - There are no limits on the number of retries (indefinite retry).
> - You don't need to manually configure a specific `ctrl_loss_tmo timeout` duration when using the `nvme connect` or `nvme connect-all` commands (option -l).
> - The NVMe/TCP controllers don't experience timeouts in the event of a path failure and remain connected indefinitely.

Step 7: Validate NVMe-oF

Verify that the in-kernel NVMe multipath status, ANA status, and ONTAP namespaces are correct for the NVMe-oF configuration.

Steps

1. Verify that the in-kernel NVMe multipath is enabled:

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

   You should see the following output:

   ```
   Y
   ```

2. Verify that the appropriate NVMe-oF settings (such as, model set to NetApp ONTAP Controller and load balancing iopolicy set to round-robin) for the respective ONTAP namespaces correctly reflect on the host:

   a. Display the subsystems:

      ```
      cat /sys/class/nvme-subsystem/nvme-subsys*/model
      ```

      You should see the following output:

      ```
      NetApp ONTAP Controller
      NetApp ONTAP Controller
      ```

b. Display the policy:

```
cat /sys/class/nvme-subsystem/nvme-subsys*/iopolicy
```

You should see the following output:

```
round-robin
round-robin
```

3. Verify that the namespaces are created and correctly discovered on the host:

```
nvme list
```

Show example

```
Node         SN                   Model
---------------------------------------------------------
/dev/nvme4n1 81Ix2BVuekWcAAAAAAB   NetApp ONTAP Controller


Namespace Usage    Format             FW             Rev
-----------------------------------------------------------
1                 21.47 GB / 21.47 GB   4 KiB + 0 B   FFFFFFFF
```

4. Verify that the controller state of each path is live and has the correct ANA status:

NVMe/FC

```
nvme list-subsys /dev/nvme5n1
```

Show example

```
nvme-subsys5 - NQN=nqn.1992-
08.com.netapp:sn.f7565b15a66911ef9668d039ea951c46:subsystem.nvme
1
               hostnqn=nqn.2014-08.org.nvmexpress:uuid:4c4c4544-
0056-5410-8048-c7c04f425633
\
 +- nvme126 fc traddr=nn-0x2036d039ea951c45:pn-
0x2038d039ea951c45,host_traddr=nn-0x2000f4c7aa0cd7c3:pn-
0x2100f4c7aa0cd7c3 live optimized
 +- nvme176 fc traddr=nn-0x2036d039ea951c45:pn-
0x2037d039ea951c45,host_traddr=nn-0x2000f4c7aa0cd7c2:pn-
0x2100f4c7aa0cd7c2 live optimized
 +- nvme5 fc traddr=nn-0x2036d039ea951c45:pn-
0x2039d039ea951c45,host_traddr=nn-0x2000f4c7aa0cd7c2:pn-
0x2100f4c7aa0cd7c2 live non-optimized
 +- nvme71 fc traddr=nn-0x2036d039ea951c45:pn-
0x203ad039ea951c45,host_traddr=nn-0x2000f4c7aa0cd7c3:pn-
0x2100f4c7aa0cd7c3 live non-optimized
```

NVMe/TCP

```
nvme list-subsys /dev/nvme4n2
```

Show example

```
nvme-subsys4 - NQN=nqn.1992-
08.com.netapp:sn.64e65e6caae711ef9668d039ea951c46:subsystem.nvme
4
               hostnqn=nqn.2014-08.org.nvmexpress:uuid:4c4c4544-
0035-5910-804b-c2c04f444d33
\
+- nvme102 tcp
traddr=192.168.21.20,trsvcid=4420,host_traddr=192.168.21.1,src_a
ddr=192.168.21.1 live non-optimized
+- nvme151 tcp
traddr=192.168.21.21,trsvcid=4420,host_traddr=192.168.21.1,src_a
ddr=192.168.21.1 live optimized
+- nvme4 tcp
traddr=192.168.20.20,trsvcid=4420,host_traddr=192.168.20.1,src_a
ddr=192.168.20.1 live non-optimized
+- nvme53 tcp
traddr=192.168.20.21,trsvcid=4420,host_traddr=192.168.20.1,src_a
ddr=192.168.20.1 live optimized
```

5. Verify that the NetApp plug-in displays the correct values for each ONTAP namespace device:

Column

```
nvme netapp ontapdevices -o column
```

Show example

```
Device         Vserver   Namespace Path
----------------------- ------------------------------
/dev/nvme10n1     vs_tcp_rhel10       /vol/vol10/ns10

NSID       UUID                                   Size
----------------------- ------------------------------
1    bbf51146-fc64-4197-b8cf-8a24f6f359b3   21.47GB
```

JSON

```
nvme netapp ontapdevices -o json
```

Show example

```
{
  "ONTAPdevices":[
    {
      "Device":"/dev/nvme10n1",
      "Vserver":"vs_tcp_rhel10",
      "Namespace_Path":"/vol/vol10/ns10",
      "NSID":1,
      "UUID":"bbf51146-fc64-4197-b8cf-8a24f6f359b3",
      "Size":"21.47GB",
      "LBA_Data_Size":4096,
      "Namespace_Size":5242880
}
]
    }
```

Step 8: Set up secure in-band authentication

Beginning with ONTAP 9.12.1, secure in-band authentication is supported over NVMe/TCP between a RHEL 10 host and an ONTAP controller.

Each host or controller must be associated with a `DH-HMAC-CHAP` key to set up secure authentication. A `DH-`

`HMAC-CHAP` key is a combination of the NQN of the NVMe host or controller and an authentication secret configured by the administrator. To authenticate its peer, an NVMe host or controller must recognize the key associated with the peer.

Set up secure in-band authentication using the CLI or a config JSON file. If you need to specify different dhchap keys for different subsystems, you must use a config JSON file.

CLI

Set up secure in-band authentication using the CLI.

Steps

1. Obtain the host NQN:

   ```
   cat /etc/nvme/hostnqn
   ```

2. Generate the dhchap key for the RHEL 10 host.

   The following output describes the `gen-dhchap-key` command paramters:

   ```
   nvme gen-dhchap-key -s optional_secret -l key_length {32|48|64} -m
   HMAC_function {0|1|2|3} -n host_nqn
   •   -s secret key in hexadecimal characters to be used to initialize
   the host key
   •   -l length of the resulting key in bytes
   •   -m HMAC function to use for key transformation
   0 = none, 1- SHA-256, 2 = SHA-384, 3=SHA-512
   •   -n host NQN to use for key transformation
   ```

   In the following example, a random dhchap key with HMAC set to 3 (SHA-512) is generated.

   ```
   nvme gen-dhchap-key -m 3 -n nqn.2014-
   08.org.nvmexpress:uuid:4c4c4544-0035-5910-804b-c2c04f444d33
   DHHC-
   1:03:7zf8I9gaRcDWH3tCH5vLGaoyjzPIvwNWusBfKdpJa+hia1aKDKJQ2o53pX3wYM9
   xdv5DtKNNhJInZ7X8wU2RQpQIngc=:
   ```

3. On the ONTAP controller, add the host and specify both dhchap keys:

   ```
   vserver nvme subsystem host add -vserver <svm_name> -subsystem
   <subsystem> -host-nqn <host_nqn> -dhchap-host-secret
   <authentication_host_secret> -dhchap-controller-secret
   <authentication_controller_secret> -dhchap-hash-function {sha-
   256|sha-512} -dhchap-group {none|2048-bit|3072-bit|4096-bit|6144-
   bit|8192-bit}
   ```

4. A host supports two types of authentication methods, unidirectional and bidirectional. On the host, connect to the ONTAP controller and specify dhchap keys based on the chosen authentication method:

```
nvme connect -t tcp -w <host-traddr> -a <tr-addr> -n <host_nqn> -S
<authentication_host_secret> -C <authentication_controller_secret>
```

5. Validate the `nvme connect authentication` command by verifying the host and controller dhchap keys:

   a. Verify the host dhchap keys:

   ```
   cat /sys/class/nvme-subsystem/<nvme-subsysX>/nvme*/dhchap_secret
   ```

   Show example output for a unidirectional configuration

   ```
   cat /sys/class/nvme-subsystem/nvme-subsys1/nvme*/dhchap_secret
   DHHC-
   1:03:fMCrJharXUOqRoIsOEaG6m2PH1yYvu5+z3jTmzEKUbcWu26I33b93b
   il2WR09XDho/ld3L45J+0FeCsStBEAfhYgkQU=:
   DHHC-
   1:03:fMCrJharXUOqRoIsOEaG6m2PH1yYvu5+z3jTmzEKUbcWu26I33b93b
   il2WR09XDho/ld3L45J+0FeCsStBEAfhYgkQU=:
   DHHC-
   1:03:fMCrJharXUOqRoIsOEaG6m2PH1yYvu5+z3jTmzEKUbcWu26I33b93b
   il2WR09XDho/ld3L45J+0FeCsStBEAfhYgkQU=:
   DHHC-
   1:03:fMCrJharXUOqRoIsOEaG6m2PH1yYvu5+z3jTmzEKUbcWu26I33b93b
   il2WR09XDho/ld3L45J+0FeCsStBEAfhYgkQU=:
   ```

   b. Verify the controller dhchap keys:

   ```
   cat /sys/class/nvme-subsystem/<nvme-
   subsysX>/nvme*/dhchap_ctrl_secret
   ```

Show example output for a bidirectional configuration

```
cat /sys/class/nvme-subsystem/nvme-
subsys6/nvme*/dhchap_ctrl_secret
DHHC- 1:03:7zf8I9gaRcDWH3tCH5vLGaoyjzPIvwNWusBfKdpJa+hia
1aKDKJQ2o53pX3wYM9xdv5DtKNNhJInZ7X8wU2RQpQIngc=:

DHHC- 1:03:7zf8I9gaRcDWH3tCH5vLGaoyjzPIvwNWusBfKdpJa+hia
1aKDKJQ2o53pX3wYM9xdv5DtKNNhJInZ7X8wU2RQpQIngc=:

DHHC- 1:03:7zf8I9gaRcDWH3tCH5vLGaoyjzPIvwNWusBfKdpJa+hia
1aKDKJQ2o53pX3wYM9xdv5DtKNNhJInZ7X8wU2RQpQIngc=:

DHHC- 1:03:7zf8I9gaRcDWH3tCH5vLGaoyjzPIvwNWusBfKdpJa+hia
1aKDKJQ2o53pX3wYM9xdv5DtKNNhJInZ7X8wU2RQpQIngc=:
```

JSON file

When multiple NVMe subsystems are available on the ONTAP controller configuration, you can use the `/etc/nvme/config.json` file with the `nvme connect-all` command.

Use the `-o` option to generate the JSON file. See the NVMe connect-all manual pages for more syntax options.

Steps

1. Configure the JSON file.

> In the following example, `dhchap_key` corresponds to `dhchap_secret` and `dhchap_ctrl_key` corresponds to `dhchap_ctrl_secret`.

Show example

```
cat /etc/nvme/config.json
[
{
"hostnqn":"nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0035-5910-
804b-c2c04f444d33",
"hostid":"4c4c4544-0035-5910-804b-c2c04f444d33",
"dhchap_key":"DHHC-
1:03:7zf8I9gaRcDWH3tCH5vLGaoyjzPIvwNWusBfKdpJa+hia1aKDKJQ2o53pX3
wYM9xdv5DtKNNhJInZ7X8wU2RQpQIngc=:",
"subsystems":[
{
"nqn":"nqn.1992-
08.com.netapp:sn.127ade26168811f0a50ed039eab69ad3:subsystem.inba
nd_unidirectional",
"ports":[
{
"transport":"tcp",
"traddr":"192.168.20.17",
"host_traddr":"192.168.20.1",
"trsvcid":"4420"
},
{
"transport":"tcp",
"traddr":"192.168.20.18",
"host_traddr":"192.168.20.1",
"trsvcid":"4420"
},
{
"transport":"tcp",
"traddr":"192.168.21.18",
"host_traddr":"192.168.21.1",
"trsvcid":"4420"
},
{
"transport":"tcp",
"traddr":"192.168.21.17",
"host_traddr":"192.168.21.1",
"trsvcid":"4420"
}]
```

2. Connect to the ONTAP controller using the config JSON file:

```
nvme connect-all -J /etc/nvme/config.json
```

Show example

```
traddr=192.168.20.20 is already connected
traddr=192.168.20.20 is already connected
traddr=192.168.20.20 is already connected
traddr=192.168.20.20 is already connected
traddr=192.168.20.20 is already connected
traddr=192.168.20.20 is already connected
traddr=192.168.20.20 is already connected
traddr=192.168.20.20 is already connected
traddr=192.168.20.21 is already connected
traddr=192.168.20.21 is already connected
traddr=192.168.20.21 is already connected
traddr=192.168.20.21 is already connected
traddr=192.168.20.21 is already connected
traddr=192.168.20.21 is already connected
traddr=192.168.20.21 is already connected
traddr=192.168.20.21 is already connected
```

3. Verify that the dhchap secrets have been enabled for the respective controllers for each subsystem.
   a. Verify the host dhchap keys:

      ```
      cat /sys/class/nvme-subsystem/nvme-subsys0/nvme0/dhchap_secret
      ```

      The following example shows a dhchap key:

      ```
      DHHC-1:03:7zf8I9gaRcDWH3tCH5vLGaoyjzPIvwNWusBfKdpJa+hia1
      aKDKJQ2o53pX3wYM9xdv5DtKNNhJInZ7X8wU2RQpQIngc=:
      ```

   b. Verify the controller dhchap keys:

      ```
      cat /sys/class/nvme-subsystem/nvme-
      subsys0/nvme0/dhchap_ctrl_secret
      ```

      You should see an output similar to the following example:

```
DHHC-1:03:fMCrJharXUOqRoIsOEaG6m2PH1yYvu5+z3jT
mzEKUbcWu26I33b93bil2WR09XDho/ld3L45J+0FeCsStBEAfhYgkQU=:
```

Step 9: Review the known issues

There are no known issues.

RHEL 9

Configure RHEL 9.6 for NVMe-oF with ONTAP storage

NetApp SAN host configurations support the NVMe over Fabrics (NVMe-oF) protocol with Asymmetric Namespace Access (ANA). In NVMe-oF environments, ANA is equivalent to asymmetric logical unit access (ALUA) multipathing in iSCSI and FCP environments. ANA is implemented using the in-kernel NVMe multipath feature.

About this task

You can use the following support and features with the NVMe-oF host configuration for Red Hat Enterprise Linux (RHEL) 9.6. You should also review the known limitations before starting the configuration process.

- Support available:
  - Support for NVMe over TCP (NVMe/TCP) in addition to NVMe over Fibre Channel (NVMe/FC). The NetApp plug-in in the native `nvme-cli` package displays ONTAP details for both NVMe/FC and NVMe/TCP namespaces.
  - Running both NVMe and SCSI traffic on the same host. For example, you can configure dm-multipath for SCSI mpath devices on SCSI LUNs and use NVMe multipath to configure NVMe-oF namespace devices on the host.
  - RHEL 9.6 enables in-kernel NVMe multipath for NVMe namespaces by default, removing the need for explicit settings.
  - Beginning with ONTAP 9.12.1, support for secure in-band authentication is introduced for NVMe/TCP. You can use secure in-band authentication for NVMe/TCP with RHEL 9.6.

  For additional details on supported configurations, see the Interoperability Matrix Tool.

- Features available:
  - There are no new features in this release..
- Known limitations:
  - Avoid issuing the `nvme disconnect-all` command on systems booting from SAN over NVMe-TCP or NVMe-FC namespaces because it disconnects both root and data filesystems and might lead to system instability.

Step 1: Optionally, enable SAN booting

You can configure your host to use SAN booting to simplify deployment and improve scalability.

Before you begin

Use the Interoperability Matrix Tool to verify that your Linux OS, host bus adapter (HBA), HBA firmware, HBA boot BIOS, and ONTAP version support SAN booting.

Steps

1. Create a SAN boot namespace and map it to the host.
2. Enable SAN booting in the server BIOS for the ports to which the SAN boot namespace is mapped.

   For information on how to enable the HBA BIOS, see your vendor-specific documentation.

3. Verify that the configuration was successful by rebooting the host and verifying that the OS is up and running.

Step 2: Validate software versions

Use the following procedure to validate the minimum supported RHEL 9.6 software versions.

Steps

1. Install RHEL 9.6 on the server. After the installation is complete, verify that you are running the specified RHEL 9.6 kernel:

   ```
   uname -r
   ```

   The following example shows a Rocky Linux kernel version:

   ```
   5.14.0-570.12.1.el9_6.x86_64
   ```

2. Install the `nvme-cli` package:

   ```
   rpm -qa|grep nvme-cli
   ```

   The following example shows an nvme-cli package version:

   ```
   nvme-cli-2.11-5.el9.x86_64
   ```

3. Install the `libnvme` package:

   ```
   rpm -qa|grep libnvme
   ```

   The following example shows an `libnvme` package version:

   ```
   libnvme-1.11.1-1.el9.x86_64
   ```

4. On the RHEL host, check the hostnqn string at `/etc/nvme/hostnqn`:

```
cat /etc/nvme/hostnqn
```

The following example shows an `hostnqn` version:

```
nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
```

5. Verify that the `hostnqn` string matches the `hostnqn` string for the corresponding subsystem on the ONTAP array:

```
::> vserver nvme subsystem host show -vserver vs_coexistence_LPE36002
```

Show example

```
Vserver Subsystem Priority  Host NQN
------- --------- --------
------------------------------------------------
vs_coexistence_LPE36002
        nvme
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
        nvme_1
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
        nvme_2
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
        nvme_3
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
4 entries were displayed.
```

If the `hostnqn` strings do not match, use the `vserver modify` command to update the `hostnqn` string on your corresponding ONTAP array subsystem to match the `hostnqn` string from `/etc/nvme/hostnqn` on the host.

Step 3: Configure NVMe/FC

You can configure NVMe/FC with Broadcom/Emulex FC or Marvell/Qlogic FC adapters. For NVMe/FC configured with a Broadcom adapter, you can enable I/O requests of size 1MB.

Broadcom/Emulex

Configure NVMe/FC for a Broadcom/Emulex adapter.

Steps

1. Verify that you are using the supported adapter model:

   a. Display the model names:

   ```
   cat /sys/class/scsi_host/host*/modelname
   ```

   You should see the following output:

   ```
   LPe36002-M64
   LPe36002-M64
   ```

   b. Display the model descriptions:

   ```
   cat /sys/class/scsi_host/host*/modeldesc
   ```

   You should see an output similar to the following example:

   ```
   Emulex LightPulse LPe36002-M64 2-Port 64Gb Fibre Channel Adapter
   Emulex LightPulse LPe36002-M64 2-Port 64Gb Fibre Channel Adapter
   ```

2. Verify that you are using the recommended Broadcom `lpfc` firmware and inbox driver:

   a. Display the firmware version:

   ```
   cat /sys/class/scsi_host/host*/fwrev
   ```

   The following example shows firmware versions:

   ```
   14.0.539.16, sli-4:6:d
   14.0.539.16, sli-4:6:d
   ```

   b. Display the inbox driver version:

   ```
   cat /sys/module/lpfc/version
   ```

   The following example shows a driver version:

```
0:14.4.0.6
```

For the current list of supported adapter driver and firmware versions, see the Interoperability Matrix Tool.

3. Verify that the expected output of `lpfc_enable_fc4_type` is set to `3`:

```
cat /sys/module/lpfc/parameters/lpfc_enable_fc4_type
```

4. Verify that you can view your initiator ports:

```
cat /sys/class/fc_host/host*/port_name
```

The following example shows port identities:

```
0x2100f4c7aa0cd7c2
0x2100f4c7aa0cd7c3
```

5. Verify that your initiator ports are online:

```
cat /sys/class/fc_host/host*/port_state
```

You should see the following output:

```
Online
Online
```

6. Verify that the NVMe/FC initiator ports are enabled and that the target ports are visible:

```
cat /sys/class/scsi_host/host*/nvme_info
```

Show example

```
NVME Initiator Enabled
XRI Dist lpfc0 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc0 WWPN x100000109b954518 WWNN x200000109b954518
DID x000000 ONLINE

NVME Statistics
LS: Xmt 0000000000 Cmpl 0000000000 Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 0000000000000000 Issue 0000000000000000 OutIO
0000000000000000
          abort 00000000 noxri 00000000 nondlp 00000000 qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 00000000 Err 00000000

NVME Initiator Enabled
XRI Dist lpfc1 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc1 WWPN x100000109b954519 WWNN x200000109b954519
DID x020500 ONLINE

NVME Statistics
LS: Xmt 0000000000 Cmpl 0000000000 Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 0000000000000000 Issue 0000000000000000 OutIO
0000000000000000
         abort 00000000 noxri 00000000 nondlp 00000000 qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 00000000 Err 00000000

NVME Initiator Enabled
XRI Dist lpfc2 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc2 WWPN x100000109bf044b1 WWNN x200000109bf044b1
DID x022a00 ONLINE
NVME RPORT       WWPN x200bd039eaa7dfc8 WWNN x2008d039eaa7dfc8
DID x021319 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2155d039eaa7dfc8 WWNN x2154d039eaa7dfc8
DID x02130f TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2001d039eaa7dfc8 WWNN x2000d039eaa7dfc8
DID x021310 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x200dd039eaa7dfc8 WWNN x2008d039eaa7dfc8
DID x020b15 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2156d039eaa7dfc8 WWNN x2154d039eaa7dfc8
DID x020b0d TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2003d039eaa7dfc8 WWNN x2000d039eaa7dfc8
DID x020b10 TARGET DISCSRVC ONLINE
```

```
NVME Statistics
LS: Xmt 0000003049 Cmpl 0000003049 Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 0000000018f9450b Issue 0000000018f5de57 OutIO
fffffffffffc994c
          abort 000036d3 noxri 00000313 nondlp 00000c8d qdepth
00000000 wqerr 00000064 err 00000000
FCP CMPL: xb 000036d1 Err 000fef0f

NVME Initiator Enabled
XRI Dist lpfc3 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc3 WWPN x100000109bf044b2 WWNN x200000109bf044b2
DID x021b00 ONLINE
NVME RPORT       WWPN x2062d039eaa7dfc8 WWNN x2008d039eaa7dfc8
DID x022915 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2157d039eaa7dfc8 WWNN x2154d039eaa7dfc8
DID x02290f TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2002d039eaa7dfc8 WWNN x2000d039eaa7dfc8
DID x022910 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2065d039eaa7dfc8 WWNN x2008d039eaa7dfc8
DID x020119 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2158d039eaa7dfc8 WWNN x2154d039eaa7dfc8
DID x02010d TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2004d039eaa7dfc8 WWNN x2000d039eaa7dfc8
DID x020110 TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 0000002f2c Cmpl 0000002f2c Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000001aaf3eb5 Issue 000000001aab4373 OutIO
fffffffffffc04be
          abort 000035cc noxri 0000038c nondlp 000009e3 qdepth
00000000 wqerr 00000082 err 00000000
FCP CMPL: xb 000035cc Err 000fcfc0
```

Marvell/QLogic

Configure NVMe/FC for a Marvell/QLogic adapter.

Steps

1. Verify that you are running the supported adapter driver and firmware versions:

```
cat /sys/class/fc_host/host*/symbolic_name
```

The follow example shows driver and firmware versions:

```
QLE2872 FW:v9.15.00 DVR:v10.02.09.300-k
QLE2872 FW:v9.15.00 DVR:v10.02.09.300-k
```

2. Verify that `ql2xnvmeenable` is set. This enables the Marvell adapter to function as an NVMe/FC initiator:

```
cat /sys/module/qla2xxx/parameters/ql2xnvmeenable
```

The expected ouptut is 1.

Step 4: Optionally, enable 1MB I/O

ONTAP reports an MDTS (Max Data Transfer Size) of 8 in the Identify Controller data. This means the maximum I/O request size can be up to 1MB. To issue I/O requests of size 1MB for a Broadcom NVMe/FC host, you should increase the `lpfc` value of the `lpfc_sg_seg_cnt` parameter to 256 from the default value of 64.

> These steps don't apply to Qlogic NVMe/FC hosts.

Steps

1. Set the `lpfc_sg_seg_cnt` parameter to 256:

```
cat /etc/modprobe.d/lpfc.conf
```

You should see an output similar to the following example:

```
options lpfc lpfc_sg_seg_cnt=256
```

2. Run the `dracut -f` command, and reboot the host.
3. Verify that the value for `lpfc_sg_seg_cnt` is 256:

```
cat /sys/module/lpfc/parameters/lpfc_sg_seg_cnt
```

Step 5: Verify NVMe boot services

With RHEL 9.6, the `nvmefc-boot-connections.service` and `nvmf-autoconnect.service` boot services included in the NVMe/FC `nvme-cli` package are automatically enabled when the system boots.

After booting completes, verify that the `nvmefc-boot-connections.service` and `nvmf-autoconnect.service` boot services are enabled.

Steps

1. Verify that `nvmf-autoconnect.service` is enabled:

```
systemctl status nvmf-autoconnect.service
```

Show example output

```
nvmf-autoconnect.service - Connect NVMe-oF subsystems automatically
during boot
     Loaded: loaded (/usr/lib/systemd/system/nvmf-
autoconnect.service; enabled; preset: disabled)
     Active: inactive (dead)

Jun 10 04:06:26 SR630-13-201.lab.eng.btc.netapp.in systemd[1]:
Starting Connect NVMe-oF subsystems automatically during boot...
Jun 10 04:06:26 SR630-13-201.lab.eng.btc.netapp.in systemd[1]: nvmf-
autoconnect.service: Deactivated successfully.
Jun 10 04:06:26 SR630-13-201.lab.eng.btc.netapp.in systemd[1]:
Finished Connect NVMe-oF subsystems automatically during boot.
```

2. Verify that `nvmefc-boot-connections.service` is enabled:

```
systemctl status nvmefc-boot-connections.service
```

Show example output

```
nvmefc-boot-connections.service - Auto-connect to subsystems on FC-
NVME devices found during boot
     Loaded: loaded (/usr/lib/systemd/system/nvmefc-boot-
connections.service; enabled; preset: enabled)
     Active: inactive (dead) since Tue 2025-06-10 01:08:36 EDT; 2h
59min ago
   Main PID: 7090 (code=exited, status=0/SUCCESS)
        CPU: 30ms

Jun 10 01:08:36 localhost systemd[1]: Starting Auto-connect to
subsystems on FC-NVME devices found during boot...
Jun 10 01:08:36 localhost systemd[1]: nvmefc-boot-
connections.service: Deactivated successfully.
Jun 10 01:08:36 localhost systemd[1]: Finished Auto-connect to
subsystems on FC-NVME devices found during boot.
```

Step 6: Configure NVMe/TCP

The NVMe/TCP protocol doesn't support the `auto-connect` operation. Instead, you can discover the NVMe/TCP subsystems and namespaces by performing the NVMe/TCP `connect` or `connect-all` operations manually.

Steps

1. Verify that the initiator port can fetch the discovery log page data across the supported NVMe/TCP LIFs:

```
nvme discover -t tcp -w host-traddr -a traddr
```

Show example

```
nvme discover -t tcp -w 192.168.1.31 -a 192.168.1.24
Discovery Log Number of Records 20, Generation counter 25
=====Discovery Log Entry 0======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  4
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr:  192.168.2.25
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 1======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  2
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr:  192.168.1.25
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 2======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  5
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr:  192.168.2.24
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 3======
trtype:  tcp
adrfam:  ipv4
```

```
subtype: current discovery subsystem
treq:    not specified
portid:  1
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr:  192.168.1.24
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 4======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_1
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 5======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_1
traddr:  192.168.1.25
eflags:  none
sectype: none
=====Discovery Log Entry 6======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  5
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_1
```

```
traddr:  192.168.2.24
eflags:  none
sectype: none
=====Discovery Log Entry 7======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_1
traddr:  192.168.1.24
eflags:  none
sectype: none
=====Discovery Log Entry 8======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_4
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 9======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_4
traddr:  192.168.1.25
eflags:  none
sectype: none
=====Discovery Log Entry 10======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
```

```
treq:    not specified
portid:  5
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_4
traddr:  192.168.2.24
eflags:  none
sectype: none
=====Discovery Log Entry 11======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_4
traddr:  192.168.1.24
eflags:  none
sectype: none
=====Discovery Log Entry 12======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_3
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 13======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_3
traddr:  192.168.1.25
```

```
eflags:  none
sectype: none
=====Discovery Log Entry 14======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  5
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_3
traddr:  192.168.2.24
eflags:  none
sectype: none
=====Discovery Log Entry 15======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_3
traddr:  192.168.1.24
eflags:  none
sectype: none
=====Discovery Log Entry 16======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_2
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 17======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
```

```
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_2
traddr:  192.168.1.25
eflags:  none
sectype: none
=====Discovery Log Entry 18======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  5
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_2
traddr:  192.168.2.24
eflags:  none
sectype: none
=====Discovery Log Entry 19======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_2
traddr:  192.168.1.24
eflags:  none
sectype: none
```

2. Verify that the other NVMe/TCP initiator-target LIF combinations are able to successfully fetch discovery log page data:

```
nvme discover -t tcp -w host-traddr -a traddr
```

Show example

```
nvme discover -t tcp -w 192.168.1.31 -a 192.168.1.24
nvme discover -t tcp -w 192.168.2.31 -a 192.168.2.24
nvme discover -t tcp -w 192.168.1.31 -a 192.168.1.25
nvme discover -t tcp -w 192.168.2.31 -a 192.168.2.25
```

3. Run the `nvme connect-all` command across all the supported NVMe/TCP initiator-target LIFs across the nodes:

```
nvme connect-all -t tcp -w host-traddr -a traddr
```

Show example

```
nvme    connect-all -t  tcp -w  192.168.1.31    -a  192.168.1.24
nvme    connect-all -t  tcp -w  192.168.2.31    -a  192.168.2.24
nvme    connect-all -t  tcp -w  192.168.1.31    -a  192.168.1.25
nvme    connect-all -t  tcp -w  192.168.2.31    -a  192.168.2.25
```

Beginning with RHEL 9.4, the setting for the NVMe/TCP `ctrl_loss_tmo timeout` is automatically set to "off". As a result:

- There are no limits on the number of retries (indefinite retry).
- You don't need to manually configure a specific `ctrl_loss_tmo timeout` duration when using the `nvme connect` or `nvme connect-all` commands (option -l).
- The NVMe/TCP controllers don't experience timeouts in the event of a path failure and remain connected indefinitely.

Step 7: Validate NVMe-oF

Verify that the in-kernel NVMe multipath status, ANA status, and ONTAP namespaces are correct for the NVMe-oF configuration.

Steps

1. Verify that the in-kernel NVMe multipath is enabled:

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

You should see the following output:

```
Y
```

2. Verify that the appropriate NVMe-oF settings (such as, model set to NetApp ONTAP Controller and load balancing iopolicy set to round-robin) for the respective ONTAP namespaces correctly reflect on the host:
   a. Display the subsystems:

      ```
      cat /sys/class/nvme-subsystem/nvme-subsys*/model
      ```

      You should see the following output:

      ```
      NetApp ONTAP Controller
      NetApp ONTAP Controller
      ```

   b. Display the policy:

      ```
      cat /sys/class/nvme-subsystem/nvme-subsys*/iopolicy
      ```

      You should see the following output:

      ```
      round-robin
      round-robin
      ```

3. Verify that the namespaces are created and correctly discovered on the host:

   ```
   nvme list
   ```

   Show example

   ```
   Node          SN                   Model
   ---------------------------------------------------------
   /dev/nvme4n1 81Ix2BVuekWcAAAAAAB   NetApp ONTAP Controller


   Namespace Usage    Format             FW             Rev
   -----------------------------------------------------------
   1                 21.47 GB / 21.47 GB   4 KiB + 0 B   FFFFFFFF
   ```

4. Verify that the controller state of each path is live and has the correct ANA status:

NVMe/FC

```
nvme list-subsys /dev/nvme4n5
```

Show example

```
nvme-subsys4 - NQN=nqn.1992-
08.com.netapp:sn.3a5d31f5502c11ef9f50d039eab6cb6d:subsystem.nvme
_1
               hostnqn=nqn.2014-08.org.nvmexpress:uuid:e6dade64-
216d-
11ec-b7bb-7ed30a5482c3
iopolicy=round-robin\
+- nvme1 fc traddr=nn-0x2082d039eaa7dfc8:pn-
0x2088d039eaa7dfc8,host_traddr=nn-0x20000024ff752e6d:pn-
0x21000024ff752e6d live optimized
+- nvme12 fc traddr=nn-0x2082d039eaa7dfc8:pn-
0x208ad039eaa7dfc8,host_traddr=nn-0x20000024ff752e6d:pn-
0x21000024ff752e6d live non-optimized
+- nvme10 fc traddr=nn-0x2082d039eaa7dfc8:pn-
0x2087d039eaa7dfc8,host_traddr=nn-0x20000024ff752e6c:pn-
0x21000024ff752e6c live non-optimized
+- nvme3 fc traddr=nn-0x2082d039eaa7dfc8:pn-
0x2083d039eaa7dfc8,host_traddr=nn-0x20000024ff752e6c:pn-
0x21000024ff752e6c live optimized
```

NVMe/TCP

```
nvme list-subsys /dev/nvme1n1
```

Show example

```
nvme-subsys5 - NQN=nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme
_tcp_3
hostnqn=nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0035-5910-804b-
b5c04f444d33
iopolicy=round-robin
\
+- nvme13 tcp
traddr=192.168.2.25,trsvcid=4420,host_traddr=192.168.2.31,
src_addr=192.168.2.31 live optimized
+- nvme14 tcp
traddr=192.168.2.24,trsvcid=4420,host_traddr=192.168.2.31,
src_addr=192.168.2.31 live non-optimized
+- nvme5 tcp
traddr=192.168.1.25,trsvcid=4420,host_traddr=192.168.1.31,
src_addr=192.168.1.31 live optimized
+- nvme6 tcp
traddr=192.168.1.24,trsvcid=4420,host_traddr=192.168.1.31,
src_addr=192.168.1.31 live non-optimized
```

5. Verify that the NetApp plug-in displays the correct values for each ONTAP namespace device:

Column

```
nvme netapp ontapdevices -o column
```

Show example

```
Device        Vserver   Namespace Path
----------------------- ------------------------------
/dev/nvme1n1     linux_tcnvme_iscsi
/vol/tcpnvme_1_0_0/tcpnvme_ns

NSID       UUID                                   Size
------------------------------------------------------------
1    5f7f630d-8ea5-407f-a490-484b95b15dd6   21.47GB
```

JSON

```
nvme netapp ontapdevices -o json
```

Show example

```
{
  "ONTAPdevices":[
    {
      "Device":"/dev/nvme1n1",
      "Vserver":"linux_tcnvme_iscsi",
      "Namespace_Path":"/vol/tcpnvme_1_0_0/tcpnvme_ns",
      "NSID":1,
      "UUID":"5f7f630d-8ea5-407f-a490-484b95b15dd6",
      "Size":"21.47GB",
      "LBA_Data_Size":4096,
      "Namespace_Size":5242880
    },
]
}
```

Step 8: Set up secure in-band authentication

Beginning with ONTAP 9.12.1, secure in-band authentication is supported over NVMe/TCP between a RHEL 9.6 host and an ONTAP controller.

Each host or controller must be associated with a `DH-HMAC-CHAP` key to set up secure authentication. A `DH-HMAC-CHAP` key is a combination of the NQN of the NVMe host or controller and an authentication secret configured by the administrator. To authenticate its peer, an NVMe host or controller must recognize the key associated with the peer.

Set up secure in-band authentication using the CLI or a config JSON file. If you need to specify different dhchap keys for different subsystems, you must use a config JSON file.

CLI

Set up secure in-band authentication using the CLI.

Steps

1. Obtain the host NQN:

```
cat /etc/nvme/hostnqn
```

2. Generate the dhchap key for the RHEL 9.6 host.

   The following output describes the `gen-dhchap-key` command paramters:

```
nvme gen-dhchap-key -s optional_secret -l key_length {32|48|64} -m
HMAC_function {0|1|2|3} -n host_nqn
•   -s secret key in hexadecimal characters to be used to initialize
the host key
•   -l length of the resulting key in bytes
•   -m HMAC function to use for key transformation
0 = none, 1- SHA-256, 2 = SHA-384, 3=SHA-512
•   -n host NQN to use for key transformation
```

   In the following example, a random dhchap key with HMAC set to 3 (SHA-512) is generated.

```
nvme gen-dhchap-key -m 3 -n nqn.2014-
08.org.nvmexpress:uuid:e6dade64-216d-11ec-b7bb-7ed30a5482c3
DHHC-
1:03:wSpuuKbBHTzC0W9JZxMBsYd9JFV8Si9aDh22k2BR/4m852vH7KGlrJeMpzhmyjD
WOo0PJJM6yZsTeEpGkDHMHQ255+g=:
```

3. On the ONTAP controller, add the host and specify both dhchap keys:

```
vserver nvme subsystem host add -vserver <svm_name> -subsystem
<subsystem> -host-nqn <host_nqn> -dhchap-host-secret
<authentication_host_secret> -dhchap-controller-secret
<authentication_controller_secret> -dhchap-hash-function {sha-
256|sha-512} -dhchap-group {none|2048-bit|3072-bit|4096-bit|6144-
bit|8192-bit}
```

4. A host supports two types of authentication methods, unidirectional and bidirectional. On the host, connect to the ONTAP controller and specify dhchap keys based on the chosen authentication method:

```
nvme connect -t tcp -w <host-traddr> -a <tr-addr> -n <host_nqn> -S
<authentication_host_secret> -C <authentication_controller_secret>
```

5. Validate the `nvme connect authentication` command by verifying the host and controller dhchap keys:

   a. Verify the host dhchap keys:

   ```
   cat /sys/class/nvme-subsystem/<nvme-subsysX>/nvme*/dhchap_secret
   ```

   Show example output for a unidirectional configuration

   ```
   cat /sys/class/nvme-subsystem/nvme-subsys1/nvme*/dhchap_secret
   DHHC-1:01:CNxTYq73T9vJk0JpOfDBZrhDCqpWBN4XVZI5WxwPgDUieHAi:
   DHHC-1:01:CNxTYq73T9vJk0JpOfDBZrhDCqpWBN4XVZI5WxwPgDUieHAi:
   DHHC-1:01:CNxTYq73T9vJk0JpOfDBZrhDCqpWBN4XVZI5WxwPgDUieHAi:
   DHHC-1:01:CNxTYq73T9vJk0JpOfDBZrhDCqpWBN4XVZI5WxwPgDUieHAi:
   ```

   b. Verify the controller dhchap keys:

   ```
   cat /sys/class/nvme-subsystem/<nvme-
   subsysX>/nvme*/dhchap_ctrl_secret
   ```

   Show example output for a bidirectional configuration

   ```
   cat /sys/class/nvme-subsystem/nvme-
   subsys6/nvme*/dhchap_ctrl_secret
   DHHC-
   1:03:wSpuuKbBHTzC0W9JZxMBsYd9JFV8Si9aDh22k2BR/4m852vH7KGlrJeMp
   zhmyjDWOo0PJJM6yZsTeEpGkDHMHQ255+g=:
   DHHC-
   1:03:wSpuuKbBHTzC0W9JZxMBsYd9JFV8Si9aDh22k2BR/4m852vH7KGlrJeMp
   zhmyjDWOo0PJJM6yZsTeEpGkDHMHQ255+g=:
   DHHC-
   1:03:wSpuuKbBHTzC0W9JZxMBsYd9JFV8Si9aDh22k2BR/4m852vH7KGlrJeMp
   zhmyjDWOo0PJJM6yZsTeEpGkDHMHQ255+g=:
   DHHC-
   1:03:wSpuuKbBHTzC0W9JZxMBsYd9JFV8Si9aDh22k2BR/4m852vH7KGlrJeMp
   zhmyjDWOo0PJJM6yZsTeEpGkDHMHQ255+g=:
   ```

JSON file

When multiple NVMe subsystems are available on the ONTAP controller configuration, you can use the `/etc/nvme/config.json` file with the `nvme connect-all` command.

Use the `-o` option to generate the JSON file. See the NVMe connect-all manual pages for more syntax options.

Steps

1. Configure the JSON file.

   > In the following example, `dhchap_key` corresponds to `dhchap_secret` and `dhchap_ctrl_key` corresponds to `dhchap_ctrl_secret`.

Show example

```
cat /etc/nvme/config.json
[
{
  "hostnqn":"nqn.2014-08.org.nvmexpress:uuid:9796c1ec-0d34-11eb-
  b6b2-3a68dd3bab57",
  "hostid":"b033cd4fd6db4724adb48655bfb55448",
  "dhchap_key":" DHHC-
1:01:CNxTYq73T9vJk0JpOfDBZrhDCqpWBN4XVZI5WxwPgDUieHAi:"
},
{
  "hostnqn":"nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0035-5910-
  804b-b5c04f444d33",
  "subsystems":[
        {
          "nqn":"nqn.1992-

08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.bidi
          r_DHCP",
          "ports":[
              {
                  "transport":"tcp",
                    "traddr":" 192.168.1.24 ",
                  "host_traddr":" 192.168.1.31 ",
                  "trsvcid":"4420",
                  "dhchap_ctrl_key":"DHHC-
                  1:03:
wSpuuKbBHTzC0W9JZxMBsYd9JFV8Si9aDh22k2BR/4m852vH7KGlrJeMpzhmyjDW
Oo0PJJM6yZsTeEpGkDHMHQ255+g=:"
              },
              {
                  "transport":"tcp",
                  "traddr":" 192.168.1.25 ",
                  "host_traddr":" 192.168.1.31",
                  "trsvcid":"4420",
                  "dhchap_ctrl_key":"DHHC-
                  1:03:
wSpuuKbBHTzC0W9JZxMBsYd9JFV8Si9aDh22k2BR/4m852vH7KGlrJeMpzhmyjDW
Oo0PJJM6yZsTeEpGkDHMHQ255+g=:"
              },
              {
                  "transport":"tcp",
                 "traddr":" 192.168.2.24 ",
                  "host_traddr":" 192.168.2.31",
                  "trsvcid":"4420",
```

```
                    "dhchap_ctrl_key":"DHHC-
                    1:03:
wSpuuKbBHTzC0W9JZxMBsYd9JFV8Si9aDh22k2BR/4m852vH7KGlrJeMpzhmyjDW
Oo0PJJM6yZsTeEpGkDHMHQ255+g=:"
                },
                {
                    "transport":"tcp",
                    "traddr":" 192.168.2.25 ",
                      "host_traddr":" 192.168.2.31",
                    "trsvcid":"4420",
                    "dhchap_ctrl_key":"DHHC-
                    1:03:
wSpuuKbBHTzC0W9JZxMBsYd9JFV8Si9aDh22k2BR/4m852vH7KGlrJeMpzhmyjDW
Oo0PJJM6yZsTeEpGkDHMHQ255+g=:"
                }
            ]
        }
    ]
}
]
```

2. Connect to the ONTAP controller using the config JSON file:

```
nvme connect-all -J /etc/nvme/config.json
```

Show example

```
already connected to hostnqn=nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0035-5910-804b-
b5c04f444d33,nqn=nqn.1992-
08.com.netapp:sn.8dde3be2cc7c11efb777d039eab6cb6d:subsystem.
bidi
r_DHCP,transport=tcp,traddr=192.168.1.25,trsvcid=4420
already connected to hostnqn=nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0035-5910-804b-
b5c04f444d33,nqn=nqn.1992-
08.com.netapp:sn.8dde3be2cc7c11efb777d039eab6cb6d:subsystem.
bidi
r_DHCP,transport=tcp,traddr=192.168.2.25,trsvcid=4420
already connected to hostnqn=nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0035-5910-804b-
b5c04f444d33,nqn=nqn.1992-
08.com.netapp:sn.8dde3be2cc7c11efb777d039eab6cb6d:subsystem.
bidi
r_DHCP,transport=tcp,traddr=192.168.1.24,trsvcid=4420
already connected to hostnqn=nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0035-5910-804b-
b5c04f444d33,nqn=nqn.1992-
08.com.netapp:sn.8dde3be2cc7c11efb777d039eab6cb6d:subsystem.
bidi
r_DHCP,transport=tcp,traddr=192.168.2.24,trsvcid=4420
```

3. Verify that the dhchap secrets have been enabled for the respective controllers for each subsystem:
   a. Verify the host dhchap keys:

   ```
   cat /sys/class/nvme-subsystem/nvme-subsys0/nvme0/dhchap_secret
   ```

   The following example shows a dhchap key:

   ```
   DHHC-1:01:CNxTYq73T9vJk0JpOfDBZrhDCqpWBN4XVZI5WxwPgDUieHAi:
   ```

   b. Verify the controller dhchap keys:

   ```
   cat /sys/class/nvme-subsystem/nvme-
   subsys0/nvme0/dhchap_ctrl_secret
   ```

   You should see an output similar to the following example:

```
DHHC-
1:03:wSpuuKbBHTzC0W9JZxMBsYd9JFV8Si9aDh22k2BR/4m852vH7KGlrJeMpzhm
yjDWOo0PJJM6yZsTeEpGkDHMHQ255+g=:
```

Step 9: Review the known issues

There are no known issues.

Configure RHEL 9.5 for NVMe-oF with ONTAP storage

NetApp SAN host configurations support the NVMe over Fabrics (NVMe-oF) protocol with Asymmetric Namespace Access (ANA). In NVMe-oF environments, ANA is equivalent to asymmetric logical unit access (ALUA) multipathing in iSCSI and FCP environments. ANA is implemented using the in-kernel NVMe multipath feature.

About this task

You can use the following support and features with the NVMe-oF host configuration for Red Hat Enterprise Linux (RHEL) 9.5. You should also review the known limitations before starting the configuration process.

- Support available:
  - Support for NVMe over TCP (NVMe/TCP) in addition to NVMe over Fibre Channel (NVMe/FC). The NetApp plug-in in the native `nvme-cli` package displays ONTAP details for both NVMe/FC and NVMe/TCP namespaces.
  - Running both NVMe and SCSI traffic on the same host. For example, you can configure dm-multipath for SCSI mpath devices on SCSI LUNs and use NVMe multipath to configure NVMe-oF namespace devices on the host.

    For additional details on supported configurations, see the Interoperability Matrix Tool.
- Features available:
  - Beginning with ONTAP 9.12.1, support for secure in-band authentication is introduced for NVMe-oF. You can use secure in-band authentication for NVMe-oF with RHEL 9.5.
  - RHEL 9.5 enables in-kernel NVMe multipath for NVMe namespaces by default, removing the need for explicit settings.
- Known limitations:
  - There are no known limitations.

Enable SAN booting

You can configure your host to use SAN booting to simplify deployment and improve scalability.

Before you begin

Use the Interoperability Matrix Tool to verify that your Linux OS, host bus adapter (HBA), HBA firmware, HBA boot BIOS, and ONTAP version support SAN booting.

Steps

1. Create a SAN boot namespace and map it to the host.

2. Enable SAN booting in the server BIOS for the ports to which the SAN boot LUN is mapped.

   For information on how to enable the HBA BIOS, see your vendor-specific documentation.

3. Verify that the configuration was successful by rebooting the host and verifying that the OS is up and running.

Validate software versions

You can use the following procedure to validate the minimum supported RHEL 9.5 software versions.

Steps

1. Install RHEL 9.5 on the server. After the installation is complete, verify that you are running the specified RHEL 9.5 kernel:

```
uname -r
```

```
5.14.0-503.11.1.el9_5.x86_64
```

2. Install the `nvme-cli` package:

```
rpm -qa|grep nvme-cli
```

```
nvme-cli-2.9.1-6.el9.x86_64
```

3. Install the `libnvme` package:

```
rpm -qa|grep libnvme
```

```
libnvme-1.9-3.el9.x86_64
```

4. On the RHEL 9.5 host, check the hostnqn string at `/etc/nvme/hostnqn`:

```
cat /etc/nvme/hostnqn
```

```
nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
```

5. Verify that the `hostnqn` string matches the `hostnqn` string for the corresponding subsystem on the ONTAP array:

```
::> vserver nvme subsystem host show -vserver vs_coexistence_LPE36002
```

Show example

```
Vserver Subsystem Priority  Host NQN
------- --------- --------
-------------------------------------------------
vs_coexistence_LPE36002
        nvme
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
        nvme_1
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
        nvme_2
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
        nvme_3
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
4 entries were displayed.
```

If the `hostnqn` strings do not match, use the `vserver modify` command to update the `hostnqn` string on your corresponding ONTAP array subsystem to match the `hostnqn` string from `/etc/nvme/hostnqn` on the host.

Configure NVMe/FC

You can configure NVMe/FC with Broadcom/Emulex FC or Marvell/Qlogic FC adapters. For NVMe/FC configured with a Broadcom adapter, you can enable I/O requests of size 1MB.

Broadcom/Emulex

Configure NVMe/FC for a Broadcom/Emulex adapter.

Steps

1. Verify that you are using the supported adapter model:
   a. `cat /sys/class/scsi_host/host*/modelname`

      ```
      LPe36002-M64
      LPe36002-M64
      ```

   b. `cat /sys/class/scsi_host/host*/modeldesc`

      ```
      Emulex LightPulse LPe36002-M64 2-Port 64Gb Fibre Channel Adapter
      Emulex LightPulse LPe36002-M64 2-Port 64Gb Fibre Channel Adapter
      ```

2. Verify that you are using the recommended Broadcom `lpfc` firmware and inbox driver:
   a. `cat /sys/class/scsi_host/host*/fwrev`

      ```
      14.4.317.10, sli-4:6:d
      14.4.317.10, sli-4:6:d
      ```

   b. `cat /sys/module/lpfc/version`

      ```
      0:14.4.0.2
      ```

   For the current list of supported adapter driver and firmware versions, see the Interoperability Matrix Tool.

3. Verify that the expected output of `lpfc_enable_fc4_type` is set to `3`:

   `cat /sys/module/lpfc/parameters/lpfc_enable_fc4_type`

   ```
   3
   ```

4. Verify that you can view your initiator ports:

   `cat /sys/class/fc_host/host*/port_name`

```
0x100000109bf044b1
0x100000109bf044b2
```

5. Verify that your initiator ports are online:

   `cat /sys/class/fc_host/host*/port_state`

   ```
   Online
   Online
   ```

6. Verify that the NVMe/FC initiator ports are enabled and that the target ports are visible:

   `cat /sys/class/scsi_host/host*/nvme_info`

Show example

```
NVME Initiator Enabled
XRI Dist lpfc2 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc2 WWPN x100000109bf044b1 WWNN x200000109bf044b1
DID x022a00 ONLINE
NVME RPORT       WWPN x202fd039eaa7dfc8 WWNN x202cd039eaa7dfc8
DID x021310 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x202dd039eaa7dfc8 WWNN x202cd039eaa7dfc8
DID x020b10 TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 0000000810 Cmpl 0000000810 Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000007b098f07 Issue 000000007aee27c4 OutIO
fffffffffe498bd
        abort 000013b4 noxri 00000000 nondlp 00000058 qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 000013b4 Err 00021443

NVME Initiator Enabled
XRI Dist lpfc3 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc3 WWPN x100000109bf044b2 WWNN x200000109bf044b2
DID x021b00 ONLINE
NVME RPORT       WWPN x2033d039eaa7dfc8 WWNN x202cd039eaa7dfc8
DID x020110 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2032d039eaa7dfc8 WWNN x202cd039eaa7dfc8
DID x022910 TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 0000000840 Cmpl 0000000840 Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000007afd4434 Issue 000000007ae31b83 OutIO
fffffffffe5d74f
        abort 000014a5 noxri 00000000 nondlp 0000006a qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 000014a5 Err 0002149a
```

Marvell/QLogic

Configure NVMe/FC for a Marvell/QLogic adapter.

> The native inbox qla2xxx driver included in the RHEL 9.5 GA kernel has the latest fixes. These fixes are essential for ONTAP support.

Steps

1. Verify that you are running the supported adapter driver and firmware versions:

```
cat /sys/class/fc_host/host*/symbolic_name
```

```
QLE2742 FW:v9.14.00 DVR:v10.02.09.200-k
QLE2742 FW:v9.14.00 DVR:v10.02.09.200-k
```

2. Verify that `ql2xnvmeenable` is set. This enables the Marvell adapter to function as an NVMe/FC initiator:

```
cat /sys/module/qla2xxx/parameters/ql2xnvmeenable
```

The expected ouptut is 1.

Enable 1MB I/O (Optional)

ONTAP reports an MDTS (Max Data Transfer Size) of 8 in the Identify Controller data. This means the maximum I/O request size can be up to 1MB. To issue I/O requests of size 1MB for a Broadcom NVMe/FC host, you should increase the `lpfc` value of the `lpfc_sg_seg_cnt` parameter to 256 from the default value of 64.

> These steps don't apply to Qlogic NVMe/FC hosts.

Steps

1. Set the `lpfc_sg_seg_cnt` parameter to 256:

```
cat /etc/modprobe.d/lpfc.conf
```

You should see an output similar to the following example:

```
options lpfc lpfc_sg_seg_cnt=256
```

2. Run the `dracut -f` command, and reboot the host.
3. Verify that the value for `lpfc_sg_seg_cnt` is 256:

```
cat /sys/module/lpfc/parameters/lpfc_sg_seg_cnt
```

Configure NVMe/TCP

The NVMe/TCP protocol doesn’t support the `auto-connect` operation. Instead, you can discover the NVMe/TCP subsystems and namespaces by performing the NVMe/TCP `connect` or `connect-all` operations manually.

Steps

1. Verify that the initiator port can fetch the discovery log page data across the supported NVMe/TCP LIFs:

   ```
   nvme discover -t tcp -w host-traddr -a traddr
   ```

Show example

```
nvme discover -t tcp -w 192.168.1.31 -a 192.168.1.24

Discovery Log Number of Records 20, Generation counter 25
=====Discovery Log Entry 0======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  4
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr:  192.168.2.25
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 1======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  2
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr:  192.168.1.25
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 2======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  5
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr:  192.168.2.24
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 3======
trtype:  tcp
```

```
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  1
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr:  192.168.1.24
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 4======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_1
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 5======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_1
traddr:  192.168.1.25
eflags:  none
sectype: none
=====Discovery Log Entry 6======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  5
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
```

```
_1
traddr:  192.168.2.24
eflags:  none
sectype: none
=====Discovery Log Entry 7======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_1
traddr:  192.168.1.24
eflags:  none
sectype: none
=====Discovery Log Entry 8======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_4
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 9======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_4
traddr:  192.168.1.25
eflags:  none
sectype: none
=====Discovery Log Entry 10======
trtype:  tcp
adrfam:  ipv4
```

```
subtype: nvme subsystem
treq:    not specified
portid:  5
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_4
traddr:  192.168.2.24
eflags:  none
sectype: none
=====Discovery Log Entry 11======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_4
traddr:  192.168.1.24
eflags:  none
sectype: none
=====Discovery Log Entry 12======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_3
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 13======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_3
```

```
traddr:  192.168.1.25
eflags:  none
sectype: none
=====Discovery Log Entry 14======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  5
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_3
traddr:  192.168.2.24
eflags:  none
sectype: none
=====Discovery Log Entry 15======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_3
traddr:  192.168.1.24
eflags:  none
sectype: none
=====Discovery Log Entry 16======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_2
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 17======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
```

```
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_2
traddr:  192.168.1.25
eflags:  none
sectype: none
=====Discovery Log Entry 18======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  5
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_2
traddr:  192.168.2.24
eflags:  none
sectype: none
=====Discovery Log Entry 19======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_2
traddr:  192.168.1.24
eflags:  none
sectype: none
```

2. Verify that the other NVMe/TCP initiator-target LIF combinations are able to successfully fetch discovery log page data:

```
nvme discover -t tcp -w host-traddr -a traddr
```

Show example

```
nvme discover -t tcp -w 192.168.1.31 -a 192.168.1.24
nvme discover -t tcp -w 192.168.2.31 -a 192.168.2.24
nvme discover -t tcp -w 192.168.1.31 -a 192.168.1.25
nvme discover -t tcp -w 192.168.2.31 -a 192.168.2.25
```

3. Run the `nvme connect-all` command across all the supported NVMe/TCP initiator-target LIFs across the nodes:

```
nvme connect-all -t tcp -w host-traddr -a traddr
```

Show example

```
nvme connect-all -t tcp -w 192.168.1.31 -a 192.168.1.24
nvme connect-all -t tcp -w 192.168.2.31 -a 192.168.2.24
nvme connect-all -t tcp -w 192.168.1.31 -a 192.168.1.25
nvme connect-all -t tcp -w 192.168.2.31 -a 192.168.2.25
```

Beginning with RHEL 9.4, the setting for the NVMe/TCP `ctrl_loss_tmo timeout` is automatically set to "off". As a result:

- There are no limits on the number of retries (indefinite retry).
- You don’t need to manually configure a specific `ctrl_loss_tmo timeout` duration when using the `nvme connect` or `nvme connect-all` commands (option -l).
- The NVMe/TCP controllers don’t experience timeouts in the event of a path failure and remain connected indefinitely.

Validate NVMe-oF

Verify that the in-kernel NVMe multipath status, ANA status, and ONTAP namespaces are correct for the NVMe-oF configuration.

Steps

1. Verify that the in-kernel NVMe multipath is enabled:

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

You should see the following output:

```
Y
```

2. Verify that the appropriate NVMe-oF settings (such as, model set to NetApp ONTAP Controller and load balancing iopolicy set to round-robin) for the respective ONTAP namespaces correctly reflect on the host:
   a. Display the subsystems:

      ```
      cat /sys/class/nvme-subsystem/nvme-subsys*/model
      ```

      You should see the following output:

      ```
      NetApp ONTAP Controller
      NetApp ONTAP Controller
      ```

   b. Display the policy:

      ```
      cat /sys/class/nvme-subsystem/nvme-subsys*/iopolicy
      ```

      You should see the following output:

      ```
      round-robin
      round-robin
      ```

3. Verify that the namespaces are created and correctly discovered on the host:

   ```
   nvme list
   ```

   Show example

   ```
   Node         SN                   Model
   ---------------------------------------------------------
   /dev/nvme4n1 81Ix2BVuekWcAAAAAAB   NetApp ONTAP Controller


   Namespace Usage    Format             FW             Rev
   -----------------------------------------------------------
   1                 21.47 GB / 21.47 GB   4 KiB + 0 B   FFFFFFFF
   ```

4. Verify that the controller state of each path is live and has the correct ANA status:

NVMe/FC

```
nvme list-subsys /dev/nvme4n5
```

Show example

```
nvme-subsys4 - NQN=nqn.1992-
08.com.netapp:sn.3a5d31f5502c11ef9f50d039eab6cb6d:subsystem.nvme
_1
               hostnqn=nqn.2014-08.org.nvmexpress:uuid:e6dade64-
216d-
11ec-b7bb-7ed30a5482c3
iopolicy=round-robin\
+- nvme1 fc traddr=nn-0x2082d039eaa7dfc8:pn-
0x2088d039eaa7dfc8,host_traddr=nn-0x20000024ff752e6d:pn-
0x21000024ff752e6d live optimized
+- nvme12 fc traddr=nn-0x2082d039eaa7dfc8:pn-
0x208ad039eaa7dfc8,host_traddr=nn-0x20000024ff752e6d:pn-
0x21000024ff752e6d live non-optimized
+- nvme10 fc traddr=nn-0x2082d039eaa7dfc8:pn-
0x2087d039eaa7dfc8,host_traddr=nn-0x20000024ff752e6c:pn-
0x21000024ff752e6c live non-optimized
+- nvme3 fc traddr=nn-0x2082d039eaa7dfc8:pn-
0x2083d039eaa7dfc8,host_traddr=nn-0x20000024ff752e6c:pn-
0x21000024ff752e6c live optimized
```

NVMe/TCP

```
nvme list-subsys /dev/nvme1n1
```

Show example

```
nvme-subsys5 - NQN=nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme
_tcp_3
hostnqn=nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0035-5910-804b-
b5c04f444d33
iopolicy=round-robin
\
+- nvme13 tcp
traddr=192.168.2.25,trsvcid=4420,host_traddr=192.168.2.31,
src_addr=192.168.2.31 live optimized
+- nvme14 tcp
traddr=192.168.2.24,trsvcid=4420,host_traddr=192.168.2.31,
src_addr=192.168.2.31 live non-optimized
+- nvme5 tcp
traddr=192.168.1.25,trsvcid=4420,host_traddr=192.168.1.31,
src_addr=192.168.1.31 live optimized
+- nvme6 tcp
traddr=192.168.1.24,trsvcid=4420,host_traddr=192.168.1.31,
src_addr=192.168.1.31 live non-optimized
```

5. Verify that the NetApp plug-in displays the correct values for each ONTAP namespace device:

Column

```
nvme netapp ontapdevices -o column
```

Show example

```
Device        Vserver   Namespace Path
----------------------- ------------------------------
/dev/nvme1n1     linux_tcnvme_iscsi
/vol/tcpnvme_1_0_0/tcpnvme_ns

NSID       UUID                                   Size
------------------------------------------------------------
1    5f7f630d-8ea5-407f-a490-484b95b15dd6   21.47GB
```

JSON

```
nvme netapp ontapdevices -o json
```

Show example

```
{
  "ONTAPdevices":[
    {
      "Device":"/dev/nvme1n1",
      "Vserver":"linux_tcnvme_iscsi",
      "Namespace_Path":"/vol/tcpnvme_1_0_0/tcpnvme_ns",
      "NSID":1,
      "UUID":"5f7f630d-8ea5-407f-a490-484b95b15dd6",
      "Size":"21.47GB",
      "LBA_Data_Size":4096,
      "Namespace_Size":5242880
    },
]
}
```

Set up secure in-band authentication

Beginning with ONTAP 9.12.1, secure in-band authentication is supported over NVMe/TCP and NVMe/FC between a RHEL 9.5 host and an ONTAP controller.

To set up secure authentication, each host or controller must be associated with a `DH-HMAC-CHAP` key, which is a combination of the NQN of the NVMe host or controller and an authentication secret configured by the administrator. To authenticate its peer, an NVMe host or controller must recognize the key associated with the peer.

You can set up secure in-band authentication using the CLI or a config JSON file. If you need to specify different dhchap keys for different subsystems, you must use a config JSON file.

CLI

Set up secure in-band authentication using the CLI.

Steps

1. Obtain the host NQN:

```
cat /etc/nvme/hostnqn
```

2. Generate the dhchap key for the RHEL 9.5 host.

   The following output describes the `gen-dhchap-key` command paramters:

```
nvme gen-dhchap-key -s optional_secret -l key_length {32|48|64} -m
HMAC_function {0|1|2|3} -n host_nqn
•   -s secret key in hexadecimal characters to be used to initialize
the host key
•   -l length of the resulting key in bytes
•   -m HMAC function to use for key transformation
0 = none, 1- SHA-256, 2 = SHA-384, 3=SHA-512
•   -n host NQN to use for key transformation
```

   In the following example, a random dhchap key with HMAC set to 3 (SHA-512) is generated.

```
# nvme gen-dhchap-key -m 3 -n nqn.2014-
08.org.nvmexpress:uuid:e6dade64-216d-11ec-b7bb-7ed30a5482c3
DHHC-
1:03:1CFivw9ccz58gAcOUJrM7Vs98hd2ZHSr+iw+Amg6xZPl5D2Yk+HDTZiUAg1iGgx
TYqnxukqvYedA55Bw3wtz6sJNpR4=:
```

3. On the ONTAP controller, add the host and specify both dhchap keys:

```
vserver nvme subsystem host add -vserver <svm_name> -subsystem
<subsystem> -host-nqn <host_nqn> -dhchap-host-secret
<authentication_host_secret> -dhchap-controller-secret
<authentication_controller_secret> -dhchap-hash-function {sha-
256|sha-512} -dhchap-group {none|2048-bit|3072-bit|4096-bit|6144-
bit|8192-bit}
```

4. A host supports two types of authentication methods, unidirectional and bidirectional. On the host, connect to the ONTAP controller and specify dhchap keys based on the chosen authentication method:

```
nvme connect -t tcp -w <host-traddr> -a <tr-addr> -n <host_nqn> -S
<authentication_host_secret> -C <authentication_controller_secret>
```

5. Validate the `nvme connect authentication` command by verifying the host and controller dhchap keys:

   a. Verify the host dhchap keys:

   ```
   cat /sys/class/nvme-subsystem/<nvme-subsysX>/nvme*/dhchap_secret
   ```

   Show example output for a unidirectional configuration

   ```
   # cat /sys/class/nvme-subsystem/nvme-
   subsys1/nvme*/dhchap_secret
   DHHC-1:01:iM63E6cX7G5SOKKOju8gmzM53qywsy+C/YwtzxhIt9ZRz+ky:
   DHHC-1:01:iM63E6cX7G5SOKKOju8gmzM53qywsy+C/YwtzxhIt9ZRz+ky:
   DHHC-1:01:iM63E6cX7G5SOKKOju8gmzM53qywsy+C/YwtzxhIt9ZRz+ky:
   DHHC-1:01:iM63E6cX7G5SOKKOju8gmzM53qywsy+C/YwtzxhIt9ZRz+ky:
   ```

   b. Verify the controller dhchap keys:

   ```
   cat /sys/class/nvme-subsystem/<nvme-
   subsysX>/nvme*/dhchap_ctrl_secret
   ```

   Show example output for a bidirectional configuration

   ```
   # cat /sys/class/nvme-subsystem/nvme-
   subsys6/nvme*/dhchap_ctrl_secret
   DHHC-
   1:03:1CFivw9ccz58gAcOUJrM7Vs98hd2ZHSr+iw+Amg6xZPl5D2Yk+HDTZiUA
   g1iGgxTYqnxukqvYedA55Bw3wtz6sJNpR4=:
   DHHC-
   1:03:1CFivw9ccz58gAcOUJrM7Vs98hd2ZHSr+iw+Amg6xZPl5D2Yk+HDTZiUA
   g1iGgxTYqnxukqvYedA55Bw3wtz6sJNpR4=:
   DHHC-
   1:03:1CFivw9ccz58gAcOUJrM7Vs98hd2ZHSr+iw+Amg6xZPl5D2Yk+HDTZiUA
   g1iGgxTYqnxukqvYedA55Bw3wtz6sJNpR4=:
   DHHC-
   1:03:1CFivw9ccz58gAcOUJrM7Vs98hd2ZHSr+iw+Amg6xZPl5D2Yk+HDTZiUA
   g1iGgxTYqnxukqvYedA55Bw3wtz6sJNpR4=:
   ```

JSON file

When multiple NVMe subsystems are available on the ONTAP controller configuration, you can use the `/etc/nvme/config.json` file with the `nvme connect-all` command.

To generate the JSON file, you can use the `-o` option. See the NVMe connect-all manual pages for more syntax options.

Steps

1. Configure the JSON file:

Show example

```
# cat /etc/nvme/config.json
[
{
  "hostnqn":"nqn.2014-08.org.nvmexpress:uuid:9796c1ec-0d34-11eb-
b6b2-3a68dd3bab57",
  "hostid":"b033cd4fd6db4724adb48655bfb55448",
  "dhchap_key":"DHHC-
1:01:zGlgmRyWbplWfUCPMuaP3mAypX0+GHuSczx5vX4Yod9lMPim:"
},
{
  "hostnqn":"nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0035-5910-
804b-b5c04f444d33",
  "subsystems":[
       {
          "nqn":"nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.bidi
r_DHCP",
          "ports":[
               {
                    "transport":"tcp",
                     "traddr":" 192.168.1.24 ",
                    "host_traddr":" 192.168.1.31 ",
                    "trsvcid":"4420",
                    "dhchap_ctrl_key":"DHHC-
1:03:L52ymUoR32zYvnqZFe5OHhMg4gxD79jIyxSShHansXpVN+WiXE222aVc651
JxGZlQCI863iVOz5dNWvgb+14F4B4bTQ=:"
               },
               {
                    "transport":"tcp",
                    "traddr":" 192.168.1.24 ",
                    "host_traddr":" 192.168.1.31",
                    "trsvcid":"4420",
                    "dhchap_ctrl_key":"DHHC-
1:03:L52ymUoR32zYvnqZFe5OHhMg4gxD79jIyxSShHansXpVN+WiXE222aVc651
JxGZlQCI863iVOz5dNWvgb+14F4B4bTQ=:"
               },
               {
                    "transport":"tcp",
                   "traddr":" 192.168.1.24 ",
                    "host_traddr":" 192.168.1.31",
                    "trsvcid":"4420",
                    "dhchap_ctrl_key":"DHHC-
1:03:L52ymUoR32zYvnqZFe5OHhMg4gxD79jIyxSShHansXpVN+WiXE222aVc651
JxGZlQCI863iVOz5dNWvgb+14F4B4bTQ=:"
```

```
                },
                {
                    "transport":"tcp",
                    "traddr":" 192.168.1.24 ",
                     "host_traddr":" 192.168.1.31",
                    "trsvcid":"4420",
                    "dhchap_ctrl_key":"DHHC-
1:03:L52ymUoR32zYvnqZFe5OHhMg4gxD79jIyxSShHansXpVN+WiXE222aVc651
JxGZlQCI863iVOz5dNWvgb+14F4B4bTQ=:"
                }
            ]
        }
    ]
}
]
```

> In the preceding example, `dhchap_key` corresponds to `dhchap_secret` and `dhchap_ctrl_key` corresponds to `dhchap_ctrl_secret`.

2. Connect to the ONTAP controller using the config JSON file:

```
# nvme connect-all -J /etc/nvme/config.json
```

Show example

```
traddr=192.168.1.24 is already connected
traddr=192.168.1.24 is already connected
traddr=192.168.1.24 is already connected
traddr=192.168.1.24 is already connected
traddr=192.168.1.24 is already connected
traddr=192.168.1.24 is already connected
traddr=192.168.1.25 is already connected
traddr=192.168.1.25 is already connected
traddr=192.168.1.25 is already connected
traddr=192.168.1.25 is already connected
traddr=192.168.1.25 is already connected
traddr=192.168.1.25 is already connected
```

3. Verify that the dhchap secrets have been enabled for the respective controllers for each subsystem:

   a. Verify the host dhchap keys:

```
# cat /sys/class/nvme-subsystem/nvme-subsys0/nvme0/dhchap_secret
```

```
DHHC-1:01:zGlgmRyWbplWfUCPMuaP3mAypX0+GHuSczx5vX4Yod9lMPim:
```

b. Verify the controller dhchap keys:

```
# cat /sys/class/nvme-subsystem/nvme-
subsys0/nvme0/dhchap_ctrl_secret
```

```
DHHC-
1:03:L52ymUoR32zYvnqZFe5OHhMg4gxD79jIyxSShHansXpVN+WiXE222aVc651J
xGZlQCI863iVOz5dNWvgb+14F4B4bTQ=:
```

Known issues

No known issues exist for the NVMe-oF host configuration on RHEL 9.5 with ONTAP release.

NVMe-oF host configuration for RHEL 9.4 with ONTAP

NVMe over Fabrics (NVMe-oF), including NVMe over Fibre Channel (NVMe/FC) and other transports, is supported with Red Hat Enterprise Linux (RHEL) 9.4 with Asymmetric Namespace Access (ANA). In NVMe-oF environments, ANA is the equivalent of ALUA multipathing in iSCSI and FC environments and is implemented with in-kernel NVMe multipath.

The following support is available for NVMe-oF host configuration for RHEL 9.4 with ONTAP:

- Support for NVMe over TCP (NVMe/TCP) in addition to NVMe/FC. The NetApp plug-in in the native `nvme-cli` package displays ONTAP details for both NVMe/FC and NVMe/TCP namespaces.
- Use of NVMe and SCSI co-existent traffic on the same host on a given host bus adapter (HBA) without the explicit dm-multipath settings to prevent claiming NVMe namespaces.

For additional details on supported configurations, see the Interoperability Matrix Tool.

Features

- RHEL 9.4 has in-kernel NVMe multipath enabled for NVMe namespaces by default; therefore, there is no need for explicit settings.
- SAN booting using the NVMe/FC protocol is supported.

Known limitations

There are no known limitations.

Enable SAN booting

You can configure your host to use SAN booting to simplify deployment and improve scalability.

Before you begin

Use the Interoperability Matrix Tool to verify that your Linux OS, host bus adapter (HBA), HBA firmware, HBA boot BIOS, and ONTAP version support SAN booting.

Steps

1. Create a SAN boot namespace and map it to the host.

   See Provision NVMe storage.

2. Enable SAN booting in the server BIOS for the ports to which the SAN boot LUN is mapped.

   For information on how to enable the HBA BIOS, see your vendor-specific documentation.

3. Verify that the configuration was successful by rebooting the host and verifying that the OS is up and running.

Validate software versions

You can use the following procedure to validate the minimum supported RHEL 9.4 software versions.

Steps

1. Install RHEL 9.4 on the server. After the installation is complete, verify that you are running the specified RHEL 9.4 kernel:

   ```
   # uname -r
   ```

   Example output:

   ```
   5.14.0-423.el9.x86_64
   ```

2. Install the `nvme-cli` package:

   ```
   # rpm -qa|grep nvme-cli
   ```

   Example output:

   ```
   nvme-cli-2.6-4.el9.x86_64
   ```

3. Install the `libnvme` package:

   ```
   #rpm -qa|grep libnvme
   ```

Example output

```
libnvme-1.6-1.el9.x86_64
```

4. On the RHEL 9.4 host, check the hostnqn string at `/etc/nvme/hostnqn`:

```
# cat /etc/nvme/hostnqn
```

Example output

```
nqn.2014-08.org.nvmexpress:uuid: uuid:4c4c4544-0036-5610-804a-
c7c04f365a32
```

5. Verify that the `hostnqn` string matches the `hostnqn` string for the corresponding subsystem on the ONTAP array:

```
::> vserver nvme subsystem host show -vserver vs_coexistence_LPE36002
```

Example output:

```
Vserver     Subsystem          Host NQN
----------- ---------------
----------------------------------------------------------
vs_coexistence_LPE36002   nvme    nqn.2014-08.org.nvmexpress:uuid:
4c4c4544-0036-5610-804a-
```

If the `hostnqn` strings do not match, use the `vserver modify` command to update the `hostnqn` string on your corresponding ONTAP array subsystem to match the `hostnqn` string from `/etc/nvme/hostnqn` on the host.

Configure NVMe/FC

You can configure NVMe/FC for Broadcom/Emulex or Marvell/Qlogic adapters.

Broadcom/Emulex

Steps

1. Verify that you are using the supported adapter model:

   ```
   # cat /sys/class/scsi_host/host*/modelname
   ```

   Example output:

   ```
   LPe36002-M64
   LPe36002-M64
   ```

   ```
   # cat /sys/class/scsi_host/host*/modeldesc
   ```

   Example output:

   ```
   Emulex LightPulse LPe36002-M64 2-Port 64Gb Fibre Channel Adapter
   Emulex LightPulse LPe36002-M64 2-Port 64Gb Fibre Channel Adapter
   ```

2. Verify that you are using the recommended Broadcom `lpfc` firmware and inbox driver:

   ```
   # cat /sys/class/scsi_host/host*/fwrev
   14.2.673.40, sli-4:6:d
   14.2.673.40, sli-4:6:d


   # cat /sys/module/lpfc/version
   0:14.2.0.16
   ```

   For the current list of supported adapter driver and firmware versions, see the Interoperability Matrix Tool.

3. Verify that `lpfc_enable_fc4_type` is set to `3`:

   ```
   # cat /sys/module/lpfc/parameters/lpfc_enable_fc4_type
   3
   ```

4. Verify that the initiator ports are up and running and that you can see the target LIFs:

```
# cat /sys/class/fc_host/host*/port_name
0x100000109b3c081f
0x100000109b3c0820
```

```
# cat /sys/class/fc_host/host*/port_state
Online
Online
```

```
# cat /sys/class/scsi_host/host*/nvme_info
NVME Initiator Enabled
XRI Dist lpfc0 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc0 WWPN x100000109b3c081f WWNN x200000109b3c081f DID
x062300 ONLINE
NVME RPORT       WWPN x2143d039ea165877 WWNN x2142d039ea165877 DID
x061b15 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2145d039ea165877 WWNN x2142d039ea165877 DID
x061115 TARGET DISCSRVC ONLINE
NVME Statistics
LS: Xmt 000000040b Cmpl 000000040b Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000001f5c4538 Issue 000000001f58da22 OutIO
fffffffffffc94ea
abort 00000630 noxri 00000000 nondlp 00001071 qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 00000630 Err 0001bd4a
NVME Initiator Enabled
XRI Dist lpfc1 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc1 WWPN x100000109b3c0820 WWNN x200000109b3c0820 DID
x062c00 ONLINE
NVME RPORT       WWPN x2144d039ea165877 WWNN x2142d039ea165877 DID
x060215 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2146d039ea165877 WWNN x2142d039ea165877 DID
x061815 TARGET DISCSRVC ONLINE
NVME Statistics
LS: Xmt 000000040b Cmpl 000000040b Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000001f5c3618 Issue 000000001f5967a4 OutIO
fffffffffffd318c
abort 00000629 noxri 00000000 nondlp 0000044e qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 00000629 Err 0001bd3d
```

Marvell/QLogic FC Adapter for NVMe/FC

The native inbox qla2xxx driver included in the RHEL 9.4 GA kernel has the latest fixes. These fixes are essential for ONTAP support.

Steps

1. Verify that you are running the supported adapter driver and firmware versions:

   ```
   # cat /sys/class/fc_host/host*/symbolic_name
   ```

   Example output

   ```
   QLE2872 FW:v9.12.01 DVR:v10.02.09.100-k
   QLE2872 FW:v9.12.01 DVR:v10.02.09.100-k
   ```

2. Verify that `ql2xnvmeenable` is set. This enables the Marvell adapter to function as an NVMe/FC initiator:

   ```
   # cat /sys/module/qla2xxx/parameters/ql2xnvmeenable
   1
   ```

Enable 1MB I/O (Optional)

ONTAP reports an MDTS (Max Data Transfer Size) of 8 in the Identify Controller data. This means the maximum I/O request size can be up to 1MB. To issue I/O requests of size 1MB for a Broadcom NVMe/FC host, you should increase the `lpfc` value of the `lpfc_sg_seg_cnt` parameter to 256 from the default value of 64.

> ⓘ These steps don't apply to Qlogic NVMe/FC hosts.

Steps

1. Set the `lpfc_sg_seg_cnt` parameter to 256:

   ```
   cat /etc/modprobe.d/lpfc.conf
   ```

   You should see an output similar to the following example:

   ```
   options lpfc lpfc_sg_seg_cnt=256
   ```

2. Run the `dracut -f` command, and reboot the host.
3. Verify that the value for `lpfc_sg_seg_cnt` is 256:

```
cat /sys/module/lpfc/parameters/lpfc_sg_seg_cnt
```

Configure NVMe/TCP

NVMe/TCP does not have an auto-connect functionality. Instead, you can discover the NVMe/TCP subsystems and namespaces by performing the NVMe/TCP `connect` or `connect-all` operations manually.

Steps

1. Verify that the initiator port can fetch the discovery log page data across the supported NVMe/TCP LIFs:

   ```
   nvme discover -t tcp -w host-traddr -a traddr
   ```

   Example output:

   ```
   # nvme discover -t tcp -w 192.168.167.1 -a 192.168.167.16

   Discovery Log Number of Records 8, Generation counter 10
   =====Discovery Log Entry 0======
   trtype:  tcp
   adrfam:  ipv4
   subtype: current discovery subsystem
   treq:    not specified
   portid:  11
   trsvcid: 8009
   subnqn:  nqn.1992-08.com.netapp:sn.983de7f4b39411ee871ed039ea954d18:
   discovery
   traddr:  192.168.167.8
   eflags:  explicit discovery connections, duplicate discovery information
   sectype: none
   =====Discovery Log Entry 1======
   trtype:  tcp
   adrfam:  ipv4
   subtype: current discovery subsystem
   treq:    not specified
   portid:  9
   trsvcid: 8009
   subnqn:  nqn.1992-08.com.netapp:sn.983de7f4b39411ee871ed039ea954d18:
   discovery
   traddr:  192.168.166.8
   eflags:  explicit discovery connections, duplicate discovery information
   sectype: none
   =====Discovery Log Entry 2======
   trtype:  tcp
   adrfam:  ipv4
   ```

```
subtype: current discovery subsystem
treq:    not specified
portid:  12
trsvcid: 8009
subnqn:  nqn.1992-08.com.netapp:sn.983de7f4b39411ee871ed039ea954d18:
discovery
traddr:  192.168.167.7
eflags:  explicit discovery connections, duplicate discovery information
sectype: none
=====Discovery Log Entry 3======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  10
trsvcid: 8009
subnqn:  nqn.1992-08.com.netapp:sn.983de7f4b39411ee871ed039ea954d18:
discovery
traddr:  192.168.166.7
eflags:  explicit discovery connections, duplicate discovery information
sectype: none
=====Discovery Log Entry 4======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  11
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.983de7f4b39411ee871ed039ea954d18:subsystem.nvme_tcp_1
traddr:  192.168.167.8
eflags:  none
sectype: none
=====Discovery Log Entry 5======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  9
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.983de7f4b39411ee871ed039ea954d18:subsystem.nvme_tcp_1
traddr:  192.168.166.8
eflags:  none
sectype: none
=====Discovery Log Entry 6======
```

```
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  12
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.983de7f4b39411ee871ed039ea954d18:subsystem.nvme_tcp_1
traddr:  192.168.167.7
eflags:  none
sectype: none
=====Discovery Log Entry 7======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  10
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.983de7f4b39411ee871ed039ea954d18:subsystem.nvme_tcp_1
traddr:  192.168.166.7
eflags:  none
sectype: none
```

2. Verify that the other NVMe/TCP initiator-target LIF combinations are able to successfully fetch discovery log page data:

```
nvme discover -t tcp -w host-traddr -a traddr
```

Example output:

```
#nvme discover -t tcp -w 192.168.166.6 -a 192.168.166.7
#nvme discover -t tcp -w 192.168.166.6 -a 192.168.166.8
#nvme discover -t tcp -w 192.168.167.6 -a 192.168.167.7
#nvme discover -t tcp -w 192.168.167.6 -a 192.168.167.8
```

3. Run the `nvme connect-all` command across all the supported NVMe/TCP initiator-target LIFs across the nodes:

```
nvme connect-all -t tcp -w host-traddr -a traddr
```

Example output:

```
#   nvme    connect-all -t  tcp -w  192.168.166.6   -a  192.168.166.7
#   nvme    connect-all -t  tcp -w  192.168.166.6   -a  192.168.166.8
#   nvme    connect-all -t  tcp -w  192.168.167.6   -a  192.168.167.7
#   nvme    connect-all -t  tcp -w  192.168.167.6   -a  192.168.167.8
```

> Beginning with RHEL 9.4, the setting for the NVMe/TCP `ctrl_loss_tmo timeout` is automatically set to "off". As a result:
>
> - There are no limits on the number of retries (indefinite retry).
> - You don't need to manually configure a specific `ctrl_loss_tmo timeout` duration when using the `nvme connect` or `nvme connect-all` commands (option -l).
> - The NVMe/TCP controllers don't experience timeouts in the event of a path failure and remain connected indefinitely.

Validate NVMe-oF

You can use the following procedure to validate NVME-oF.

Steps

1. Verify that the in-kernel NVMe multipath is enabled:

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

2. Verify that the appropriate NVMe-oF settings (such as, model set to NetApp ONTAP Controller and load balancing iopolicy set to round-robin) for the respective ONTAP namespaces correctly reflect on the host:

```
# cat /sys/class/nvme-subsystem/nvme-subsys*/model
NetApp ONTAP Controller
NetApp ONTAP Controller
```

```
# cat /sys/class/nvme-subsystem/nvme-subsys*/iopolicy
round-robin
round-robin
```

3. Verify that the namespaces are created and correctly discovered on the host:

```
# nvme list
```

Example output:

```
Node         SN                   Model
---------------------------------------------------------
/dev/nvme4n1 81Ix2BVuekWcAAAAAAB   NetApp ONTAP Controller


Namespace Usage    Format             FW             Rev
-----------------------------------------------------------
1                 21.47 GB / 21.47 GB   4 KiB + 0 B   FFFFFFFF
```

4. Verify that the controller state of each path is live and has the correct ANA status:

NVMe/FC

```
# nvme list-subsys /dev/nvme5n21
```

Example output:

```
nvme-subsys4 - NQN=nqn.1992-
08.com.netapp:sn.efd7989cb10111ee871ed039ea954d18:subsystem.nvme
            hostnqn=nqn.2014-08.org.nvmexpress:uuid:d3b581b4-c975-
11e6-8425-0894ef31a074
 iopolicy=round-robin
 \
  +- nvme2 fc traddr=nn-0x2013d039ea951c45:pn-
0x2018d039ea951c45,host_traddr=nn-0x200000109bdacc76:pn-
0x100000109bdacc76 live non-optimized
  +- nvme3 fc traddr=nn-0x2013d039ea951c45:pn-
0x2017d039ea951c45,host_traddr=nn-0x200000109bdacc75:pn-
0x100000109bdacc75 live non-optimized
  +- nvme5 fc traddr=nn-0x2013d039ea951c45:pn-
0x2016d039ea951c45,host_traddr=nn-   0x200000109bdacc76:pn-
0x100000109bdacc76 live optimized
  +- nvme6 fc traddr=nn-0x2013d039ea951c45:pn-
0x2014d039ea951c45,host_traddr=nn-  0x200000109bdacc75:pn-
0x100000109bdacc75 live optimized
```

NVMe/TCP

```
# nvme list-subsys /dev/nvme1n1
```

Example output:

```
nvme-subsys1 -NQN=nqn.1992-08.com.netapp:
sn.983de7f4b39411ee871ed039ea954d18:subsystem.nvme_tcp_1
hostnqn=nqn.2014-08.org.nvmexpress:uuid:
4c4c4544-0035-5910-804b-c2c04f444d33
iopolicy=round-robin
\
+- nvme5 tcp
traddr=192.168.166.7,trsvcid=4420,host_traddr=192.168.166.6,src_addr
=192.168.166.6 live
+- nvme4 tcp
traddr=192.168.166.8,trsvcid=4420,host_traddr=192.168.166.6,src_addr
=192.168.166.6 live
+- nvme2 tcp
traddr=192.168.167.7,trsvcid=4420,host_traddr=192.168.167.6,src_addr
=192.168.167.6 live
+- nvme1 tcp
traddr=192.168.167.8,trsvcid=4420,host_traddr=192.168.167.6,src_addr
=192.168.167.6 live
```

5. Verify that the NetApp plug-in displays the correct values for each ONTAP namespace device:

Column

```
# nvme netapp ontapdevices -o column
```

Example output:

```
Device        Vserver   Namespace Path
----------------------- ------------------------------
/dev/nvme0n1 vs_tcp           /vol/vol1/ns1



NSID       UUID                                   Size
------------------------------------------------------------
1          6fcb8ea0-dc1e-4933-b798-8a62a626cb7f 21.47GB
```

JSON

```
# nvme netapp ontapdevices -o json
```

Example output

```
{

"ONTAPdevices" : [
{

"Device" : "/dev/nvme1n1", "Vserver" : "linux_tcnvme_iscsi",
"Namespace_Path" : "/vol/tcpnvme_1_0_0/tcpnvme_ns", "NSID" : 1,
"UUID" : "1a42c652-1450-4a29-886a-b4ccc23e637d", "Size" : "21.47GB",
"LBA_Data_Size" : 4096,
"Namespace_Size" : 5242880
},

]
}
```

Known issues

There are no known issues for the NVMe-oF host configuration for RHEL 9.4 with ONTAP release.

NVMe-oF host configuration for RHEL 9.3 with ONTAP

NVMe over Fabrics (NVMe-oF), including NVMe over Fibre Channel (NVMe/FC) and other transports, is supported with Red Hat Enterprise Linux (RHEL) 9.3 with Asymmetric Namespace Access (ANA). In NVMe-oF environments, ANA is the equivalent of ALUA multipathing in iSCSI and FC environments and is implemented with in-kernel NVMe multipath.

The following support is available for NVMe-oF host configuration for RHEL 9.3 with ONTAP:

- Support for NVMe over TCP (NVMe/TCP) in addition to NVMe/FC. The NetApp plug-in in the native nvme-cli package displays ONTAP details for both NVMe/FC and NVMe/TCP namespaces.
- Use of NVMe and SCSI co-existent traffic on the same host on a given host bus adapter (HBA)without the explicit dm-multipath settings to prevent claiming NVMe namespaces.

For additional details on supported configurations, see the Interoperability Matrix Tool.

Features

RHEL 9.3 has in-kernel NVMe multipath enabled for NVMe namespaces by default; therefore, there is no need for explicit settings.

Known limitations

SAN booting using the NVMe-oF protocol is currently not supported.

Enable SAN booting

You can configure your host to use SAN booting to simplify deployment and improve scalability.

Before you begin

Use the Interoperability Matrix Tool to verify that your Linux OS, host bus adapter (HBA), HBA firmware, HBA boot BIOS, and ONTAP version support SAN booting.

Steps

1. Create a SAN boot namespace and map it to the host.

   See Provision NVMe storage.

2. Enable SAN booting in the server BIOS for the ports to which the SAN boot LUN is mapped.

   For information on how to enable the HBA BIOS, see your vendor-specific documentation.

3. Verify that the configuration was successful by rebooting the host and verifying that the OS is up and running.

Validate software versions

You can use the following procedure to validate the minimum supported RHEL 9.3 software versions.

Steps

1. Install RHEL 9.3 on the server. After the installation is complete, verify that you are running the specified RHEL 9.3 kernel:

```
# uname -r
```

Example output:

```
5.14.0-362.8.1.el9_3.x86_64
```

2. Install the `nvme-cli` package:

```
# rpm -qa|grep nvme-cli
```

Example output:

```
nvme-cli-2.4-10.el9.x86_64
```

3. Install the `libnvme` package:

```
#rpm -qa|grep libnvme
```

Example output

```
libnvme-1.4-7.el9.x86_64
```

4. On the RHEL 9.3 host, check the hostnqn string at `/etc/nvme/hostnqn`:

```
# cat /etc/nvme/hostnqn
```

Example output

```
nqn.2014-08.org.nvmexpress:uuid:060fd513-83be-4c3e-aba1-52e169056dcf
```

5. Verify that the `hostnqn` string matches the `hostnqn` string for the corresponding subsystem on the ONTAP array:

```
::> vserver nvme subsystem host show -vserver vs_nvme147
```

Example output:

```
Vserver     Subsystem          Host NQN
----------- ---------------
----------------------------------------------------------
vs_nvme147   rhel_147_LPe32002    nqn.2014-
08.org.nvmexpress:uuid:060fd513-83be-4c3e-aba1-52e169056dcf
```

> If the `hostnqn` strings do not match, use the `vserver modify` command to update the `hostnqn` string on your corresponding ONTAP array subsystem to match the `hostnqn` string from `/etc/nvme/hostnqn` on the host.

Configure NVMe/FC

You can configure NVMe/FC for Broadcom/Emulex or Marvell/Qlogic adapters.

Broadcom/Emulex

Steps

1. Verify that you are using the supported adapter model:

   ```
   # cat /sys/class/scsi_host/host*/modelname
   ```

   Example output:

   ```
   LPe32002-M2
   LPe32002-M2
   ```

   ```
   # cat /sys/class/scsi_host/host*/modeldesc
   ```

   Example output:

   ```
   Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
   Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
   ```

2. Verify that you are using the recommended Broadcom `lpfc` firmware and inbox driver:

   ```
   # cat /sys/class/scsi_host/host*/fwrev
   14.2.539.16, sli-4:2:c
   14.2.539.16, sli-4:2:c

   # cat /sys/module/lpfc/version
   0:14.2.0.12
   ```

   For the current list of supported adapter driver and firmware versions, see the Interoperability Matrix Tool.

3. Verify that `lpfc_enable_fc4_type` is set to `3`:

   ```
   # cat /sys/module/lpfc/parameters/lpfc_enable_fc4_type
   3
   ```

4. Verify that the initiator ports are up and running and that you can see the target LIFs:

```
# cat /sys/class/fc_host/host*/port_name
0x100000109b3c081f
0x100000109b3c0820
```

```
# cat /sys/class/fc_host/host*/port_state
Online
Online
```

```
# cat /sys/class/scsi_host/host*/nvme_info
NVME Initiator Enabled
XRI Dist lpfc0 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc0 WWPN x100000109b3c081f WWNN x200000109b3c081f DID
x062300 ONLINE
NVME RPORT       WWPN x2143d039ea165877 WWNN x2142d039ea165877 DID
x061b15 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2145d039ea165877 WWNN x2142d039ea165877 DID
x061115 TARGET DISCSRVC ONLINE
NVME Statistics
LS: Xmt 000000040b Cmpl 000000040b Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000001f5c4538 Issue 000000001f58da22 OutIO
fffffffffffc94ea
abort 00000630 noxri 00000000 nondlp 00001071 qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 00000630 Err 0001bd4a
NVME Initiator Enabled
XRI Dist lpfc1 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc1 WWPN x100000109b3c0820 WWNN x200000109b3c0820 DID
x062c00 ONLINE
NVME RPORT       WWPN x2144d039ea165877 WWNN x2142d039ea165877 DID
x060215 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2146d039ea165877 WWNN x2142d039ea165877 DID
x061815 TARGET DISCSRVC ONLINE
NVME Statistics
LS: Xmt 000000040b Cmpl 000000040b Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000001f5c3618 Issue 000000001f5967a4 OutIO
fffffffffffd318c
abort 00000629 noxri 00000000 nondlp 0000044e qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 00000629 Err 0001bd3d
```

Marvell/QLogic FC Adapter for NVMe/FC

The native inbox qla2xxx driver included in the RHEL 9.3 GA kernel has the latest fixes. These fixes are essential for ONTAP support.

Steps

1. Verify that you are running the supported adapter driver and firmware versions:

   ```
   # cat /sys/class/fc_host/host*/symbolic_name
   ```

   Example output

   ```
   QLE2772 FW:v9.10.11 DVR:v10.02.08.200-k
   QLE2772 FW:v9.10.11 DVR:v10.02.08.200-k
   ```

2. Verify that `ql2xnvmeenable` is set. This enables the Marvell adapter to function as an NVMe/FC initiator:

   ```
   # cat /sys/module/qla2xxx/parameters/ql2xnvmeenable
   1
   ```

Enable 1MB I/O (Optional)

ONTAP reports an MDTS (Max Data Transfer Size) of 8 in the Identify Controller data. This means the maximum I/O request size can be up to 1MB. To issue I/O requests of size 1MB for a Broadcom NVMe/FC host, you should increase the `lpfc` value of the `lpfc_sg_seg_cnt` parameter to 256 from the default value of 64.

> ⓘ These steps don't apply to Qlogic NVMe/FC hosts.

Steps

1. Set the `lpfc_sg_seg_cnt` parameter to 256:

   ```
   cat /etc/modprobe.d/lpfc.conf
   ```

   You should see an output similar to the following example:

   ```
   options lpfc lpfc_sg_seg_cnt=256
   ```

2. Run the `dracut -f` command, and reboot the host.
3. Verify that the value for `lpfc_sg_seg_cnt` is 256:

```
cat /sys/module/lpfc/parameters/lpfc_sg_seg_cnt
```

Configure NVMe/TCP

NVMe/TCP does not have auto-connect functionality. Therefore, if a path goes down and is not reinstated within the default time out period of 10 minutes, NVMe/TCP cannot automatically reconnect. To prevent a time out, you should set the retry period for failover events to at least 30 minutes.

Steps

1. Verify that the initiator port can fetch the discovery log page data across the supported NVMe/TCP LIFs:

```
nvme discover -t tcp -w host-traddr -a traddr
```

Example output:

```
# nvme discover -t tcp -w 192.168.167.1 -a 192.168.167.16

Discovery Log Number of Records 8, Generation counter 10
=====Discovery Log Entry 0======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  0
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.bbfb4ee8dfb611edbd07d039ea165590:discovery
traddr:  192.168.166.17
eflags:  explicit discovery connections, duplicate discovery information
sectype: none
=====Discovery Log Entry 1======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  1
trsvcid: 8009
subnqn:  nqn.1992
08.com.netapp:sn.bbfb4ee8dfb611edbd07d039ea165590:discovery
traddr:  192.168.167.17
eflags:  explicit discovery connections, duplicate discovery information
sectype: none
=====Discovery Log Entry 2======
trtype:  tcp
```

```
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  2
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.bbfb4ee8dfb611edbd07d039ea165590:discovery
traddr:  192.168.166.16
eflags: explicit discovery connections, duplicate discovery information
sectype: none
=====Discovery Log Entry 3======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  3
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.bbfb4ee8dfb611edbd07d039ea165590:discovery
traddr:  192.168.167.16
eflags:  explicit discovery connections, duplicate discovery information
sectype: none
...
```

2. Verify that the other NVMe/TCP initiator-target LIF combinations are able to successfully fetch discovery log page data:

```
nvme discover -t tcp -w host-traddr -a traddr
```

Example output:

```
#nvme discover -t tcp -w 192.168.166.5 -a 192.168.166.22
#nvme discover -t tcp -w 192.168.166.5 -a 192.168.166.23
#nvme discover -t tcp -w 192.168.167.5 -a 192.168.167.22
#nvme discover -t tcp -w 192.168.167.5 -a 192.168.167.23
```

3. Run the `nvme connect-all` command across all the supported NVMe/TCP initiator-target LIFs across the nodes, and set the controller loss timeout period for at least 30 minutes or 1800 seconds:

```
nvme connect-all -t tcp -w host-traddr -a traddr -l 1800
```

Example output:

```
#   nvme    connect-all -t  tcp -w  192.168.166.1   -a  192.168.166.16
-l   1800
#   nvme    connect-all -t  tcp -w  192.168.166.1   -a  192.168.166.17
-l   1800
#   nvme    connect-all -t  tcp -w  192.168.167.1   -a  192.168.167.16
-l   1800
#   nvme    connect-all -t  tcp -w  192.168.167.1   -a  192.168.167.17
-l   1800
```

Validate NVMe-oF

You can use the following procedure to validate NVME-oF.

Steps

1. Verify that the in-kernel NVMe multipath is enabled:

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

2. Verify that the appropriate NVMe-oF settings (such as, model set to NetApp ONTAP Controller and load balancing iopolicy set to round-robin) for the respective ONTAP namespaces correctly reflect on the host:

```
# cat /sys/class/nvme-subsystem/nvme-subsys*/model
NetApp ONTAP Controller
NetApp ONTAP Controller
```

```
# cat /sys/class/nvme-subsystem/nvme-subsys*/iopolicy
round-robin
round-robin
```

3. Verify that the namespaces are created and correctly discovered on the host:

```
# nvme list
```

Example output:

```
Node          SN                   Model
---------------------------------------------------------
/dev/nvme5n21 81CYrNQlis3WAAAAAAAB  NetApp ONTAP Controller


Namespace Usage    Format             FW            Rev
-----------------------------------------------------------
1                 21.47 GB / 21.47 GB   4 KiB + 0 B   FFFFFFFF
```

4. Verify that the controller state of each path is live and has the correct ANA status:

NVMe/FC

```
# nvme list-subsys /dev/nvme5n21
```

Example output:

```
nvme-subsys4 - NQN=nqn.1992-
08.com.netapp:sn.e80cc121ca6911ed8cbdd039ea165590:subsystem.rhel_
147_LPE32002
\
 +- nvme2 fc traddr=nn-0x2142d039ea165877:pn-
0x2144d039ea165877,host_traddr=nn-0x200000109b3c0820:pn-
0x100000109b3c0820 live optimized
 +- nvme3 fc traddr=nn-0x2142d039ea165877:pn-
0x2145d039ea165877,host_traddr=nn-0x200000109b3c081f:pn-
0x100000109b3c081f live non-optimized
 +- nvme4 fc traddr=nn-0x2142d039ea165877:pn-
0x2146d039ea165877,host_traddr=nn-0x200000109b3c0820:pn-
0x100000109b3c0820 live non-optimized
 +- nvme6 fc traddr=nn-0x2142d039ea165877:pn-
0x2143d039ea165877,host_traddr=nn-0x200000109b3c081f:pn-
0x100000109b3c081f live optimized
```

NVMe/TCP

```
# nvme list-subsys /dev/nvme1n1
```

Example output:

```
nvme-subsys1 - NQN=nqn.1992- 08.com.netapp:sn.
bbfb4ee8dfb611edbd07d039ea165590:subsystem.rhel_tcp_95
+- nvme1 tcp
traddr=192.168.167.16,trsvcid=4420,host_traddr=192.168.167.1,src_add
r=192.168.167.1 live
+- nvme2 tcp
traddr=192.168.167.17,trsvcid=4420,host_traddr=192.168.167.1,src_add
r=192.168.167.1 live
+- nvme3 tcp
traddr=192.168.167.17,trsvcid=4420,host_traddr=192.168.166.1,src_add
r=192.168.166.1 live
+- nvme4 tcp
traddr=192.168.166.16,trsvcid=4420,host_traddr=192.168.166.1,src_add
r=192.168.166.1 live
```

5. Verify that the NetApp plug-in displays the correct values for each ONTAP namespace device:

Column

```
# nvme netapp ontapdevices -o column
```

Example output:

```
Device        Vserver   Namespace Path
----------------------- ------------------------------
/dev/nvme0n1 vs_tcp           /vol/vol1/ns1



NSID       UUID                                   Size
------------------------------------------------------------
1          6fcb8ea0-dc1e-4933-b798-8a62a626cb7f 21.47GB
```

JSON

```
# nvme netapp ontapdevices -o json
```

Example output

```
{

"ONTAPdevices" : [
{

"Device" : "/dev/nvme1n1",
"Vserver" : "vs_tcp_95",
"Namespace_Path" : "/vol/vol1/ns1",
"NSID" : 1,
"UUID" : "6fcb8ea0-dc1e-4933-b798-8a62a626cb7f",
"Size" : "21.47GB",
"LBA_Data_Size" : 4096,
"Namespace_Size" : 5242880
},

]
}
```

Known issues

There are no known issues for the NVMe-oF host configuration for RHEL 9.3 with ONTAP release.

NVMe-oF host configuration for RHEL 9.2 with ONTAP

NVMe over Fabrics (NVMe-oF), including NVMe over Fibre Channel (NVMe/FC) and other transports, is supported with Red Hat Enterprise Linux (RHEL) 9.2 with Asymmetric Namespace Access (ANA). In NVMe-oF environments, ANA is the equivalent of ALUA multipathing in iSCSI and FC environments and is implemented with in-kernel NVMe multipath.

The following support is available for the NVMe-oF host configuration for RHEL 9.2 with ONTAP:

- Support for NVMe over TCP (NVMe/TCP) in addition to NVMe/FC. The NetApp plug-in in the native nvme-cli package displays ONTAP details for both NVMe/FC and NVMe/TCP namespaces.
- Use of NVMe and SCSI co-existent traffic on the same host on a given host bus adapter (HBA), without the explicit dm-multipath settings to prevent claiming NVMe namespaces.

For additional details on supported configurations, see the Interoperability Matrix Tool.

Features

- RHEL 9.2 has in-kernel NVMe multipath enabled for NVMe namespaces by default, therefore, there is no need for explicit settings.

Known limitations

SAN booting using the NVMe-oF protocol is currently not supported.

Enable SAN booting

You can configure your host to use SAN booting to simplify deployment and improve scalability.

Before you begin

Use the Interoperability Matrix Tool to verify that your Linux OS, host bus adapter (HBA), HBA firmware, HBA boot BIOS, and ONTAP version support SAN booting.

Steps

1. Create a SAN boot namespace and map it to the host.

   See Provision NVMe storage.

2. Enable SAN booting in the server BIOS for the ports to which the SAN boot LUN is mapped.

   For information on how to enable the HBA BIOS, see your vendor-specific documentation.

3. Verify that the configuration was successful by rebooting the host and verifying that the OS is up and running.

Validate software versions

You can use the following procedure to validate the minimum supported RHEL 9.2 software versions.

Steps

1. Install RHEL 9.2 on the server. After the installation is complete, verify that you are running the specified RHEL 9.2 kernel.

```
# uname -r
```

Example output:

```
5.14.0-284.11.1.el9_2.x86_64
```

2. Install the `nvme-cli` package:

```
# rpm -qa|grep nvme-cli
```

Example output:

```
nvme-cli-2.2.1-2.el9.x86_64
```

3. Install the `libnvme` package:

```
#rpm -qa|grep libnvme
```

Example output

```
libnvme-1.2-2.el9.x86_64
```

4. On the RHEL 9.2 host, check the hostnqn string at `/etc/nvme/hostnqn`:

```
# cat /etc/nvme/hostnqn
```

Example output

```
nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0032-3310-8033-b8c04f4c5132
```

5. Verify that the `hostnqn` string matches the `hostnqn` string for the corresponding subsystem on the ONTAP array:

```
::> vserver nvme subsystem host show -vserver vs_ol_nvme
```

Example output:

```
Vserver     Subsystem          Host NQN
----------- ---------------
----------------------------------------------------------
vs_nvme207   rhel_207_LPe32002    nqn.2014-
08.org.nvmexpress:uuid:325e7554-1f9b-11ec-8489-3a68dd61a4df
```

If the `hostnqn` strings do not match, use the `vserver modify` command to update the `hostnqn` string on your corresponding ONTAP array subsystem to match the `hostnqn` string from `/etc/nvme/hostnqn` on the host.

Configure NVMe/FC

You can configure NVMe/FC for Broadcom/Emulex or Marvell/Qlogic adapters.

Broadcom/Emulex

Steps

1. Verify that you are using the supported adapter model.

```
# cat /sys/class/scsi_host/host*/modelname
```

Example output:

```
LPe32002-M2
LPe32002-M2
```

```
# cat /sys/class/scsi_host/host*/modeldesc
```

Example output:

```
Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
```

2. Verify that you are using the recommended Broadcom `lpfc` firmware and inbox driver.

```
# cat /sys/class/scsi_host/host*/fwrev
14.0.639.18, sli-4:2:c
14.0.639.18, sli-4:2:c

# cat /sys/module/lpfc/version
0:12.8.0.11
```

For the current list of supported adapter driver and firmware versions, see the Interoperability Matrix Tool.

3. Verify that `lpfc_enable_fc4_type` is set to `3`:

```
# cat /sys/module/lpfc/parameters/lpfc_enable_fc4_type
3
```

4. Verify that the initiator ports are up and running and that you can see the target LIFs:

```
# cat /sys/class/fc_host/host*/port_name
0x100000109b1c1204
0x100000109b1c1205
```

```
# cat /sys/class/fc_host/host*/port_state
Online
Online
```

```
# cat /sys/class/scsi_host/host*/nvme_info
NVME Initiator Enabled
XRI Dist lpfc0 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc0 WWPN x100000109b1c1204 WWNN x200000109b1c1204 DID
x011d00 ONLINE
NVME RPORT WWPN x203800a098dfdd91 WWNN x203700a098dfdd91 DID x010c07
TARGET DISCSRVC ONLINE
NVME RPORT WWPN x203900a098dfdd91 WWNN x203700a098dfdd91 DID x011507
TARGET DISCSRVC ONLINE
NVME Statistics
LS: Xmt 0000000f78 Cmpl 0000000f78 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000002fe29bba Issue 000000002fe29bc4 OutIO
000000000000000a
abort 00001bc7 noxri 00000000 nondlp 00000000 qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 00001e15 Err 0000d906
NVME Initiator Enabled
XRI Dist lpfc1 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc1 WWPN x100000109b1c1205 WWNN x200000109b1c1205 DID
x011900 ONLINE
NVME RPORT WWPN x203d00a098dfdd91 WWNN x203700a098dfdd91 DID x010007
TARGET DISCSRVC ONLINE
NVME RPORT WWPN x203a00a098dfdd91 WWNN x203700a098dfdd91 DID x012a07
TARGET DISCSRVC ONLINE
NVME Statistics
LS: Xmt 0000000fa8 Cmpl 0000000fa8 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000002e14f170 Issue 000000002e14f17a OutIO
000000000000000a
abort 000016bb noxri 00000000 nondlp 00000000 qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 00001f50 Err 0000d9f8
```

Marvell/QLogic FC Adapter for NVMe/FC

The native inbox qla2xxx driver included in the RHEL 9.2 GA kernel has the latest upstream fixes. These fixes are essential for ONTAP support.

Steps

1. Verify that you are running the supported adapter driver and firmware versions:

```
# cat /sys/class/fc_host/host*/symbolic_name
```

   Example output

```
QLE2742 FW:v9.08.02 DVR:v10.02.00.106-k
QLE2742 FW:v9.08.02 DVR:v10.02.00.106-k
```

2. Verify that `ql2xnvmeenable` is set. This enables the Marvell adapter to function as an NVMe/FC initiator:

```
# cat /sys/module/qla2xxx/parameters/ql2xnvmeenable
1
```

Enable 1MB I/O (Optional)

ONTAP reports an MDTS (Max Data Transfer Size) of 8 in the Identify Controller data. This means the maximum I/O request size can be up to 1MB. To issue I/O requests of size 1MB for a Broadcom NVMe/FC host, you should increase the `lpfc` value of the `lpfc_sg_seg_cnt` parameter to 256 from the default value of 64.

> These steps don't apply to Qlogic NVMe/FC hosts.

Steps

1. Set the `lpfc_sg_seg_cnt` parameter to 256:

```
cat /etc/modprobe.d/lpfc.conf
```

   You should see an output similar to the following example:

```
options lpfc lpfc_sg_seg_cnt=256
```

2. Run the `dracut -f` command, and reboot the host.
3. Verify that the value for `lpfc_sg_seg_cnt` is 256:

```
cat /sys/module/lpfc/parameters/lpfc_sg_seg_cnt
```

Configure NVMe/TCP

NVMe/TCP does not have auto-connect functionality. Therefore, if a path goes down and is not reinstated within the default time out period of 10 minutes, NVMe/TCP cannot automatically reconnect. To prevent a time out, you should set the retry period for failover events to at least 30 minutes.

Steps

1. Verify that the initiator port can fetch the discovery log page data across the supported NVMe/TCP LIFs:

   ```
   nvme discover -t tcp -w host-traddr -a traddr
   ```

   Example output:

```
# nvme discover -t tcp -w 192.168.167.5 -a 192.168.167.22

Discovery Log Number of Records 8, Generation counter 18
=====Discovery Log Entry 0======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  0
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.c680f5bcae1411ed8639d039ea951c46:discovery
traddr:  192.168.166.23
eflags:  explicit discovery connections, duplicate discovery information
sectype: none
=====Discovery Log Entry 1======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  1
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.c680f5bcae1411ed8639d039ea951c46:discovery
traddr:  192.168.166.22
eflags:  explicit discovery connections, duplicate discovery information
sectype: none
=====Discovery Log Entry 2======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  2
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.c680f5bcae1411ed8639d039ea951c46:discovery
traddr:  192.168.167.23
eflags:  explicit discovery connections, duplicate discovery information
sectype: none
..........
```

2. Verify that the other NVMe/TCP initiator-target LIF combinations are able to successfully fetch discovery log page data.

```
nvme discover -t tcp -w host-traddr -a traddr
```

Example output:

```
#nvme discover -t tcp -w 192.168.166.5 -a 192.168.166.22
#nvme discover -t tcp -w 192.168.166.5 -a 192.168.166.23
#nvme discover -t tcp -w 192.168.167.5 -a 192.168.167.22
#nvme discover -t tcp -w 192.168.167.5 -a 192.168.167.23
```

3. Run the `nvme connect-all` command across all the supported NVMe/TCP initiator-target LIFs across the nodes, and set the controller loss timeout period for at least 30 minutes or 1800 seconds:

```
nvme connect-all -t tcp -w host-traddr -a traddr -l 1800
```

Example output:

```
#   nvme    connect-all -t  tcp -w  192.168.166.5   -a  192.168.166.22
-l  1800
#   nvme    connect-all -t  tcp -w  192.168.166.5   -a  192.168.166.23
-l  1800
#   nvme    connect-all -t  tcp -w  192.168.167.5   -a  192.168.167.22
-l  1800
#   nvme    connect-all -t  tcp -w  192.168.167.5   -a  192.168.167.23
-l  1800
```

Validate NVMe-oF

You can use the following procedure to validate NVME-oF.

Steps

1. Verify that the in-kernel NVMe multipath is enabled:

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

2. Verify that the appropriate NVMe-oF settings (such as, model set to NetApp ONTAP Controller and load balancing iopolicy set to round-robin) for the respective ONTAP namespaces correctly reflect on the host:

```
# cat /sys/class/nvme-subsystem/nvme-subsys*/model
NetApp ONTAP Controller
NetApp ONTAP Controller
```

```
# cat /sys/class/nvme-subsystem/nvme-subsys*/iopolicy
round-robin
round-robin
```

3. Verify that the namespaces are created and correctly discovered on the host:

```
# nvme list
```

Example output:

```
Node         SN                   Model
---------------------------------------------------------
/dev/nvme0n1 81CZ5BQuUNfGAAAAAAAB   NetApp ONTAP Controller


Namespace Usage    Format             FW             Rev
-----------------------------------------------------------
1                 21.47 GB / 21.47 GB   4 KiB + 0 B   FFFFFFFF
```

4. Verify that the controller state of each path is live and has the correct ANA status:

NVMe/FC

```
# nvme list-subsys /dev/nvme0n1
```

Example output:

```
nvme-subsys4 - NQN=nqn.1992-
08.com.netapp:sn.8763d311b2ac11ed950ed039ea951c46:subsystem.rhel_207
_LB \
+- nvme1 fc traddr=nn-0x20a6d039ea954d17:pn-
0x20a7d039ea954d17,host_traddr=nn-0x200000109b1b95ef:pn-
0x100000109b1b95ef live optimized
+- nvme2 fc traddr=nn-0x20a6d039ea954d17:pn-
0x20a8d039ea954d17,host_traddr=nn-0x200000109b1b95f0:pn-
0x100000109b1b95f0 live optimized
+- nvme3 fc traddr=nn-0x20a6d039ea954d17:pn-
0x20aad039ea954d17,host_traddr=nn-0x200000109b1b95f0:pn-
0x100000109b1b95f0 live non-optimized
+- nvme5 fc traddr=nn-0x20a6d039ea954d17:pn-
0x20a9d039ea954d17,host_traddr=nn-0x200000109b1b95ef:pn-
0x100000109b1b95ef live non-optimized
```

NVMe/TCP

```
# nvme list-subsys /dev/nvme1n1
```

Example output:

```
nvme-subsys1 - NQN=nqn.1992-
08.com.netapp:sn.c680f5bcae1411ed8639d039ea951c46:subsystem.rhel_tcp
97 \
+- nvme1 tcp
traddr=192.168.167.23,trsvcid=4420,host_traddr=192.168.167.5 live
non-optimized
+- nvme2 tcp
traddr=192.168.167.22,trsvcid=4420,host_traddr=192.168.167.5 live
non-optimized
+- nvme3 tcp
traddr=192.168.166.23,trsvcid=4420,host_traddr=192.168.166.5 live
optimized
+- nvme4 tcp
traddr=192.168.166.22,trsvcid=4420,host_traddr=192.168.166.5 live
optimized
```

5. Verify that the NetApp plug-in displays the correct values for each ONTAP namespace device:

Column

```
# nvme netapp ontapdevices -o column
```

Example output:

```
Device        Vserver   Namespace Path
----------------------- ------------------------------
/dev/nvme0n1 vs_tcp           /vol/vol1/ns1



NSID       UUID                                   Size
------------------------------------------------------------
1          79c2c569-b7fa-42d5-b870-d9d6d7e5fa84 21.47GB
```

JSON

```
# nvme netapp ontapdevices -o json
```

Example output

```
{

"ONTAPdevices" : [
{

"Device" : "/dev/nvme0n1",
"Vserver" : "vs_tcp79",
"Namespace_Path" : "/vol/vol1/ns1",
"NSID" : 1,
"UUID" : "79c2c569-b7fa-42d5-b870-d9d6d7e5fa84",
"Size" : "21.47GB",
"LBA_Data_Size" : 4096,
"Namespace_Size" : 5242880
},

]
}
```

Known issues

There are no known issues.

NVMe-oF host configuration for RHEL 9.1 with ONTAP

NVMe over Fabrics or NVMe-oF (including NVMe/FC and NVMe/TCP) is supported with RHEL 9.1 with Asymmetric Namespace Access (ANA) that is required for surviving storage failovers (SFOs) on the ONTAP array. ANA is the asymmetric logical unit access (ALUA) equivalent in the NVMe-oF environment, and is currently implemented with in-kernel NVMe Multipath. Using this procedure, you can enable NVMe-oF with in-kernel NVMe Multipath using ANA on RHEL 9.1 and ONTAP as the target.

The following support is available for the NVMe-oF host configuration for RHEL 9.1 with ONTAP:

- Support for NVMe over TCP (NVMe/TCP) in addition to NVMe/FC. The NetApp plug-in in the native nvme-cli package displays ONTAP details for both NVMe/FC and NVMe/TCP namespaces.
- Use of NVMe and SCSI co-existent traffic on the same host on a given host bus adapter (HBA), without the explicit dm-multipath settings to prevent claiming NVMe namespaces.

Refer to the Interoperability Matrix Tool for accurate details regarding supported configurations.

Features

RHEL 9.1 includes support for in-kernel NVMe multipath for NVMe namespaces enabled by default, without the need for explicit settings.

Known limitations

SAN booting using the NVMe-oF protocol is currently not supported.

Enable SAN booting

You can configure your host to use SAN booting to simplify deployment and improve scalability.

Before you begin

Use the Interoperability Matrix Tool to verify that your Linux OS, host bus adapter (HBA), HBA firmware, HBA boot BIOS, and ONTAP version support SAN booting.

Steps

1. Create a SAN boot namespace and map it to the host.

   See Provision NVMe storage.

2. Enable SAN booting in the server BIOS for the ports to which the SAN boot LUN is mapped.

   For information on how to enable the HBA BIOS, see your vendor-specific documentation.

3. Verify that the configuration was successful by rebooting the host and verifying that the OS is up and running.

Enable in-kernel NVMe multipath

You can use the following procedure to enable in-kernel NVMe multipath.

Steps

1. Install RHEL 9.1 on the server.
2. After the installation is complete, verify that you are running the specified RHEL 9.1 kernel. See the Interoperability Matrix Tool for the current list of supported versions.

   Example:

   ```
   # uname -r
    5.14.0-162.6.1.el9_1.x86_64
   ```

3. Install the `nvme-cli` package:

   Example:

   ```
   # rpm -qa|grep nvme-cli
   nvme-cli-2.0-4.el9.x86_64
   ```

4. On the host, check the host NQN string at `/etc/nvme/hostnqn` and verify that it matches the host NQN string for the corresponding subsystem on the ONTAP array. Example:

   ```
   # cat /etc/nvme/hostnqn
   nqn.2014-08.org.nvmexpress:uuid:325e7554-1f9b-11ec-8489-3a68dd61a4df


   ::> vserver nvme subsystem host show -vserver vs_nvme207
   Vserver     Subsystem       Host NQN
   ----------- ---------------
   ----------------------------------------------------------
   vs_nvme207 rhel_207_LPe32002     nqn.2014-
   08.org.nvmexpress:uuid:325e7554-1f9b-11ec-8489-3a68dd61a4df
   ```

   If the host NQN strings do not match, you should use the `vserver modify` command to update the host NQN string on your corresponding ONTAP NVMe subsystem to match the host NQN string `/etc/nvme/hostnqn` on the host.

5. Reboot the host.

Configure NVMe/FC

You can configure NVMe/FC for Broadcom/Emulex or Marvell/Qlogic adapters.

Broadcom/Emulex

Steps

1. Verify that you are using the supported adapter. See the Interoperability Matrix Tool for the current list of supported adapters.

```
# cat /sys/class/scsi_host/host*/modelname
LPe32002-M2
LPe32002-M2

# cat /sys/class/scsi_host/host*/modeldesc

Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
```

2. Verify that you are using the recommended Broadcom lpfc firmware and inbox driver. See the Interoperability Matrix Tool for the current list of supported adapter driver and firmware versions.

```
# cat /sys/class/scsi_host/host*/fwrev
14.0.505.11, sli-4:2:c
14.0.505.11, sli-4:2:c
```

```
# cat /sys/module/lpfc/version
0:14.2.0.5
```

3. Verify that `lpfc_enable_fc4_type` is set to 3

```
# cat /sys/module/lpfc/parameters/lpfc_enable_fc4_type
3
```

4. Verify that the initiator ports are up and running, and that you can see the target LIFs.

```
# cat /sys/class/fc_host/host*/port_name
0x100000109b1b95ef
0x100000109b1b95f0
```

```
# cat /sys/class/fc_host/host*/port_state
Online
Online
```

```
# cat /sys/class/scsi_host/host*/nvme_info
NVME Initiator Enabled
XRI Dist lpfc0 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc0 WWPN x100000109b1b95ef WWNN x200000109b1b95ef DID
x061700 ONLINE
NVME RPORT       WWPN x2035d039ea1308e5 WWNN x2082d039ea1308e5 DID
x062f05 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2083d039ea1308e5 WWNN x2082d039ea1308e5 DID
x062407 TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 000000000e Cmpl 000000000e Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000000001df6c Issue 000000000001df6e OutIO
0000000000000002
        abort 00000000 noxri 00000000 nondlp 00000000 qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 00000000 Err 00000004

NVME Initiator Enabled
XRI Dist lpfc1 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc1 WWPN x100000109b1b95f0 WWNN x200000109b1b95f0 DID
x061400 ONLINE
NVME RPORT       WWPN x2036d039ea1308e5 WWNN x2082d039ea1308e5 DID
x061605 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2037d039ea1308e5 WWNN x2082d039ea1308e5 DID
x062007 TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 000000000e Cmpl 000000000e Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000000001dd28 Issue 000000000001dd29 OutIO
0000000000000001
        abort 00000000 noxri 00000000 nondlp 00000000 qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 00000000 Err 00000004
```

Marvell/QLogic FC adapter for NVMe/FC

The native inbox `qla2xxx` driver included in the RHEL 9.1 kernel has the latest fixes which are. These fixes are essential for ONTAP support.

Steps

1. Verify that you are running the supported adapter driver and firmware versions using the following command:

```
# cat /sys/class/fc_host/host*/symbolic_name
QLE2772 FW:v9.08.02 DVR:v10.02.07.400-k-debug
QLE2772 FW:v9.08.02 DVR:v10.02.07.400-k-debug
```

2. Verify `ql2xnvmeenable` is set which enables the Marvell adapter to function as an NVMe/FC initiator using the following command:

   ```
   # cat /sys/module/qla2xxx/parameters/ql2xnvmeenable
   1
   ```

Enable 1MB I/O (Optional)

ONTAP reports an MDTS (Max Data Transfer Size) of 8 in the Identify Controller data. This means the maximum I/O request size can be up to 1MB. To issue I/O requests of size 1MB for a Broadcom NVMe/FC host, you should increase the `lpfc` value of the `lpfc_sg_seg_cnt` parameter to 256 from the default value of 64.

> These steps don't apply to Qlogic NVMe/FC hosts.

Steps

1. Set the `lpfc_sg_seg_cnt` parameter to 256:

   ```
   cat /etc/modprobe.d/lpfc.conf
   ```

   You should see an output similar to the following example:

   ```
   options lpfc lpfc_sg_seg_cnt=256
   ```

2. Run the `dracut -f` command, and reboot the host.
3. Verify that the value for `lpfc_sg_seg_cnt` is 256:

   ```
   cat /sys/module/lpfc/parameters/lpfc_sg_seg_cnt
   ```

Configure NVMe/TCP

NVMe/TCP does not have auto-connect functionality. Therefore, if a path goes down and is not reinstated within the default time out period of 10 minutes, NVMe/TCP cannot automatically reconnect. To prevent a time out, you should set the retry period for failover events to at least 30 minutes.

Steps

1. Verify whether the initiator port can fetch the discovery log page data across the supported NVMe/TCP

LIFs:

```
# nvme discover -t tcp -w 192.168.1.8 -a 192.168.1.51

Discovery Log Number of Records 10, Generation counter 119
=====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.56e362e9bb4f11ebbaded039ea165abc:subsystem.nvme_118_tcp
_1
traddr: 192.168.2.56
sectype: none
=====Discovery Log Entry 1======
trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 1
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.56e362e9bb4f11ebbaded039ea165abc:subsystem.nvme_118_tcp
_1
traddr: 192.168.1.51
sectype: none
=====Discovery Log Entry 2======
trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 0
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.56e362e9bb4f11ebbaded039ea165abc:subsystem.nvme_118_tcp
_2
traddr: 192.168.2.56
sectype: none
...
```

2. Verify that the other NVMe/TCP initiator-target LIF combos can successfully fetch discovery log page data. For example:

```
# nvme discover -t tcp -w 192.168.1.8 -a 192.168.1.51
# nvme discover -t tcp -w 192.168.1.8 -a 192.168.1.52
# nvme discover -t tcp -w 192.168.2.9 -a 192.168.2.56
# nvme discover -t tcp -w 192.168.2.9 -a 192.168.2.57
```

3. Run `nvme connect-all` command across all the supported NVMe/TCP initiator-target LIFs across the nodes. Make sure you set a longer `ctrl_loss_tmo` timer retry period (for example, 30 minutes, which can be set through `-l 1800`) while running the `connect-all` command so that it would retry for a longer period of time in the event of a path loss. For example:

```
# nvme connect-all -t tcp -w 192.168.1.8 -a 192.168.1.51 -l 1800
# nvme connect-all -t tcp -w 192.168.1.8 -a 192.168.1.52 -l 1800
# nvme connect-all -t tcp -w 192.168.2.9 -a 192.168.2.56 -l 1800
# nvme connect-all -t tcp -w 192.168.2.9 -a 192.168.2.57 -l 1800
```

Validate NVMe-oF

You can use the following procedure to validate NVMe-oF.

Steps

1. Verify that in-kernel NVMe multipath is indeed enabled by checking:

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

2. Verify that the appropriate NVMe-oF settings (such as, `model` set to `NetApp ONTAP Controller` and load balancing `iopolicy` set to `round-robin`) for the respective ONTAP namespaces properly reflect on the host:

```
# cat /sys/class/nvme-subsystem/nvme-subsys*/model
NetApp ONTAP Controller
NetApp ONTAP Controller
```

```
# cat /sys/class/nvme-subsystem/nvme-subsys*/iopolicy
round-robin
round-robin
```

3. Verify that the ONTAP namespaces properly reflect on the host. For example:

```
# nvme list
Node           SN                    Model                   Namespace
------------   --------------------- ---------------------------------
/dev/nvme0n1   81CZ5BQuUNfGAAAAAAAB   NetApp ONTAP Controller   1

Usage                Format         FW Rev
-------------------  -----------    --------
85.90 GB / 85.90 GB  4 KiB + 0 B    FFFFFFFF
```

4. Verify that the controller state of each path is live and has proper ANA status. For example:

   Example (a):

```
# nvme list-subsys /dev/nvme0n1
nvme-subsys10 - NQN=nqn.1992-
08.com.netapp:sn.82e7f9edc72311ec8187d039ea14107d:subsystem.rhel_131_QLe
2742
\
 +- nvme2 fc traddr=nn-0x2038d039ea1308e5:pn-
0x2039d039ea1308e5,host_traddr=nn-0x20000024ff171d30:pn-
0x21000024ff171d30 live non-optimized
 +- nvme3 fc traddr=nn-0x2038d039ea1308e5:pn-
0x203cd039ea1308e5,host_traddr=nn-0x20000024ff171d31:pn-
0x21000024ff171d31 live optimized
 +- nvme4 fc traddr=nn-0x2038d039ea1308e5:pn-
0x203bd039ea1308e5,host_traddr=nn-0x20000024ff171d30:pn-
0x21000024ff171d30 live optimized
 +- nvme5 fc traddr=nn-0x2038d039ea1308e5:pn-
0x203ad039ea1308e5,host_traddr=nn-0x20000024ff171d31:pn-
0x21000024ff171d31 live non-optimized
```

   Example (b):

```
# nvme list-subsys /dev/nvme0n1
nvme-subsys1 - NQN=nqn.1992-
08.com.netapp:sn.bf0691a7c74411ec8187d039ea14107d:subsystem.rhel_tcp_133
\
 +- nvme1 tcp
traddr=192.168.166.21,trsvcid=4420,host_traddr=192.168.166.5 live non-
optimized
 +- nvme2 tcp
traddr=192.168.166.20,trsvcid=4420,host_traddr=192.168.166.5 live
optimized
 +- nvme3 tcp
traddr=192.168.167.21,trsvcid=4420,host_traddr=192.168.167.5 live non-
optimized
 +- nvme4 tcp
traddr=192.168.167.20,trsvcid=4420,host_traddr=192.168.167.5 live
optimized
```

5. Verify that the NetApp plug-in displays proper values for each ONTAP namespace device.

   Example (a):

```
# nvme netapp ontapdevices -o column
Device        Vserver          Namespace Path
---------     -------
--------------------------------------------------
/dev/nvme0n1 vs_tcp79     /vol/vol1/ns1

NSID  UUID                                   Size
----  ------------------------------          ------
1     79c2c569-b7fa-42d5-b870-d9d6d7e5fa84  21.47GB


# nvme netapp ontapdevices -o json
{

  "ONTAPdevices" : [
  {

      "Device" : "/dev/nvme0n1",
      "Vserver" : "vs_tcp79",
      "Namespace_Path" : "/vol/vol1/ns1",
      "NSID" : 1,
      "UUID" : "79c2c569-b7fa-42d5-b870-d9d6d7e5fa84",
      "Size" : "21.47GB",
      "LBA_Data_Size" : 4096,
      "Namespace_Size" : 5242880
    },

]

}
```

Example (b):

```
# nvme netapp ontapdevices -o column

Device           Vserver                   Namespace Path
---------------- -------------------------
-----------------------------------
/dev/nvme1n1     vs_tcp_133                /vol/vol1/ns1

NSID UUUID                                  Size
-------------------------------------------------------
1    1ef7cb56-bfed-43c1-97c1-ef22eeb92657   21.47GB

# nvme netapp ontapdevices -o json
{
  "ONTAPdevices":[
    {
      "Device":"/dev/nvme1n1",
      "Vserver":"vs_tcp_133",
      "Namespace_Path":"/vol/vol1/ns1",
      "NSID":1,
      "UUID":"1ef7cb56-bfed-43c1-97c1-ef22eeb92657",
      "Size":"21.47GB",
      "LBA_Data_Size":4096,
      "Namespace_Size":5242880
    },
  ]

}
```

Known issues

The NVMe-oF host configuration for RHEL 9.1 with ONTAP has the following known issues:

NetApp Bug ID	Title	Description
1503468	`nvme list-subsys` command returns repeated nvme controller list for a given subsystem	The `nvme list-subsys` command should return a unique list of nvme controllers associated to a given subsystem. In RHEL 9.1, the `nvme list-subsys` command returns nvme controllers with its respective ANA state for all namespaces that belong to a given subsystem. However, the ANA state is a per-namespace attribute therefore, it would be ideal to display unique nvme controller entries with the path state if you list the subsystem command syntax for a given namespace.

NVMe-oF host configuration for RHEL 9.0 with ONTAP

NVMe-oF (including NVMe/FC and NVMe/TCP) is supported with RHEL 9.0 with Asymmetric Namespace Access (ANA) required for surviving storage failovers (SFOs) on the ONTAP array. ANA is the ALUA equivalent in the NVM-oF environment, and is currently implemented with in-kernel NVMe Multipath. Using this procedure, you can enable NVMe-oF with in-kernel NVMe Multipath using ANA on RHEL 9.0 and ONTAP as the target.

For additional details on supported configurations, see the Interoperability Matrix Tool.

Features

- Beginning with RHEL 9.0, NVMe/TCP is no longer a technology preview feature (unlike RHEL 8) but a fully supported enterprise feature itself.
- Beginning with RHEL 9.0, in-kernel NVMe multipath is enabled for NVMe namespaces by default, without the need for explicit settings (unlike RHEL 8).

Known limitations

SAN booting using the NVMe-oF protocol is currently not supported.

Enable SAN booting

You can configure your host to use SAN booting to simplify deployment and improve scalability.

Before you begin

Use the Interoperability Matrix Tool to verify that your Linux OS, host bus adapter (HBA), HBA firmware, HBA boot BIOS, and ONTAP version support SAN booting.

Steps

1. Create a SAN boot namespace and map it to the host.

   See Provision NVMe storage.

2. Enable SAN booting in the server BIOS for the ports to which the SAN boot LUN is mapped.

   For information on how to enable the HBA BIOS, see your vendor-specific documentation.

3. Verify that the configuration was successful by rebooting the host and verifying that the OS is up and running.

Enable in-kernel NVMe Multipath

You can use the following procedure to enable in-kernel NVMe multipath.

Steps

1. Install RHEL 9.0 on the server.
2. After the installation is complete, verify that you are running the specified RHEL 9.0 kernel. See Interoperability Matrix Tool for the current list of supported versions.

```
# uname -r
5.14.0-70.13.1.el9_0.x86_64
```

3. Install the `nvme-cli` package.

```
# rpm -qa|grep nvme-cli
nvme-cli-1.16-3.el9.x86_64
```

4. On the host, check the host NQN string at `/etc/nvme/hostnqn` and verify that it matches the host NQN string for the corresponding subsystem on the ONTAP array. For example,

```
# cat /etc/nvme/hostnqn
nqn.2014-08.org.nvmexpress:uuid:9ed5b327-b9fc-4cf5-97b3-1b5d986345d1
```

```
::> vserver nvme subsystem host show -vserver vs_fcnvme_141
Vserver     Subsystem Host     NQN
----------- ---------------
----------------------------------------------------------
vs_fcnvme_14 nvme_141_1 nqn.2014-08.org.nvmexpress:uuid:9ed5b327-b9fc-
4cf5-97b3-1b5d986345d1
```

> If the host NQN strings do not match, you should use the `vserver modify` command to update the host NQN string on your corresponding ONTAP NVMe subsystem to match the host NQN string from `/etc/nvme/hostnqn` on the host.

5. Reboot the host.

Configure NVMe/FC

You can configure NVMe/FC for Broadcom/Emulex or Marvell/Qlogic adapters.

Broadcom/Emulex

Steps

1. Verify that you are using the supported adapter. For additional details on supported adapters, see the Interoperability Matrix Tool.

```
# cat /sys/class/scsi_host/host*/modelname
LPe32002-M2
LPe32002-M2
```

```
# cat /sys/class/scsi_host/host*/modeldesc
Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
```

2. Verify that you are using the recommended Broadcom lpfc firmware and inbox driver. For the current list of supported adapter driver and firmware versions, see Interoperability Matrix Tool.

```
# cat /sys/class/scsi_host/host*/fwrev
12.8.351.47, sli-4:2:c
12.8.351.47, sli-4:2:c
```

```
# cat /sys/module/lpfc/version
0:14.0.0.4
```

3. Verify that `lpfc_enable_fc4_type` is set to `3`.

```
# cat /sys/module/lpfc/parameters/lpfc_enable_fc4_type
3
```

4. Verify that the initiator ports are up and running, and you are able to see the target LIFs.

```
# cat /sys/class/fc_host/host*/port_name
0x100000109b1c1204
0x100000109b1c1205
```

```
# cat /sys/class/fc_host/host*/port_state
Online
Online
```

```
# cat /sys/class/scsi_host/host*/nvme_info

NVME Initiator Enabled
XRI Dist lpfc0 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc0 WWPN x100000109b1c1204 WWNN x200000109b1c1204 DID
x011d00 ONLINE
NVME RPORT WWPN x203800a098dfdd91 WWNN x203700a098dfdd91 DID x010c07
TARGET DISCSRVC ONLINE
NVME RPORT WWPN x203900a098dfdd91 WWNN x203700a098dfdd91 DID x011507
TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 0000000f78 Cmpl 0000000f78 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000002fe29bba Issue 000000002fe29bc4 OutIO
000000000000000a
abort 00001bc7 noxri 00000000 nondlp 00000000 qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 00001e15 Err 0000d906

NVME Initiator Enabled
XRI Dist lpfc1 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc1 WWPN x100000109b1c1205 WWNN x200000109b1c1205 DID
x011900 ONLINE
NVME RPORT WWPN x203d00a098dfdd91 WWNN x203700a098dfdd91 DID x010007
TARGET DISCSRVC ONLINE
NVME RPORT WWPN x203a00a098dfdd91 WWNN x203700a098dfdd91 DID x012a07
TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 0000000fa8 Cmpl 0000000fa8 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000002e14f170 Issue 000000002e14f17a OutIO
000000000000000a
abort 000016bb noxri 00000000 nondlp 00000000 qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 00001f50 Err 0000d9f8
```

Marvell/QLogic

The native inbox qla2xxx driver included in the RHEL 9.0 kernel has the latest fixes. These fixes are essential for ONTAP support.

Steps

1. Verify that you are running the supported adapter driver and firmware versions:

```
# cat /sys/class/fc_host/host*/symbolic_name
QLE2742 FW:v9.06.02 DVR:v10.02.00.200-k
QLE2742 FW:v9.06.02 DVR:v10.02.00.200-k
```

1. Verify `ql2xnvmeenable` is set which enables the Marvell adapter to function as a NVMe/FC initiator:

```
# cat /sys/module/qla2xxx/parameters/ql2xnvmeenable
1
```

Enable 1MB I/O (Optional)

ONTAP reports an MDTS (Max Data Transfer Size) of 8 in the Identify Controller data. This means the maximum I/O request size can be up to 1MB. To issue I/O requests of size 1MB for a Broadcom NVMe/FC host, you should increase the `lpfc` value of the `lpfc_sg_seg_cnt` parameter to 256 from the default value of 64.

> These steps don't apply to Qlogic NVMe/FC hosts.

Steps

1. Set the `lpfc_sg_seg_cnt` parameter to 256:

```
cat /etc/modprobe.d/lpfc.conf
```

   You should see an output similar to the following example:

```
options lpfc lpfc_sg_seg_cnt=256
```

2. Run the `dracut -f` command, and reboot the host.
3. Verify that the value for `lpfc_sg_seg_cnt` is 256:

```
cat /sys/module/lpfc/parameters/lpfc_sg_seg_cnt
```

Configure NVMe/TCP

NVMe/TCP does not have auto-connect functionality. Therefore, if a path goes down and is not reinstated within the default time out period of 10 minutes, NVMe/TCP cannot automatically reconnect. To prevent a time out, you should set the retry period for failover events to at least 30 minutes.

Steps

1. Verify whether the initiator port is able to fetch discovery log page data across the supported NVMe/TCP LIFs:

```
# nvme discover -t tcp -w 192.168.1.8 -a 192.168.1.51

Discovery Log Number of Records 10, Generation counter 119
=====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.56e362e9bb4f11ebbaded039ea165abc:subsystem.nvme_118_tcp
_1
traddr: 192.168.2.56
sectype: none
=====Discovery Log Entry 1======
trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 1
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.56e362e9bb4f11ebbaded039ea165abc:subsystem.nvme_118_tcp
_1
traddr: 192.168.1.51
sectype: none
=====Discovery Log Entry 2======
trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 0
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.56e362e9bb4f11ebbaded039ea165abc:subsystem.nvme_118_tcp
_2
traddr: 192.168.2.56
sectype: none
...
```

2. Similarly, verify that the other NVMe/TCP initiator-target LIF combos are able to successfully fetch the discovery log page data. For example,

```
# nvme discover -t tcp -w 192.168.1.8 -a 192.168.1.51
# nvme discover -t tcp -w 192.168.1.8 -a 192.168.1.52
# nvme discover -t tcp -w 192.168.2.9 -a 192.168.2.56
# nvme discover -t tcp -w 192.168.2.9 -a 192.168.2.57
```

3. Run `nvme connect-all` command across all the supported NVMe/TCP initiator-target LIFs across the nodes. Ensure you set a longer `ctrl_loss_tmo` timer retry period (for example, 30 minutes, which can be set through `-l 1800`) during the connect-all so that it would retry for a longer period of time in the event of a path loss. For example,

```
# nvme connect-all -t tcp -w 192.168.1.8 -a 192.168.1.51 -l 1800
# nvme connect-all -t tcp -w 192.168.1.8 -a 192.168.1.52 -l 1800
# nvme connect-all -t tcp -w 192.168.2.9 -a 192.168.2.56 -l 1800
# nvme connect-all -t tcp -w 192.168.2.9 -a 192.168.2.57 -l 1800
```

Validate NVMe-oF

You can use the following procedure to validate NVMe-oF.

Steps

1. Verify that in-kernel NVMe multipath is indeed enabled by checking:

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

2. Verify that the appropriate NVMf settings (for example, model set to `NetApp ONTAP Controller` and load balancing `iopolicy` set to `round-robin`) for the respective ONTAP namespaces properly reflect on the host:

```
# cat /sys/class/nvme-subsystem/nvme-subsys*/model
NetApp ONTAP Controller
NetApp ONTAP Controller
```

```
# cat /sys/class/nvme-subsystem/nvme-subsys*/iopolicy
round-robin
round-robin
```

3. Verify that the ONTAP namespaces properly reflect on the host.

   Example (a):

```
# nvme list
Node         SN                          Model                        Namespace
Usage
------      ---------------------------------------
------------------------
/dev/nvme0n1 814vWBNRwf9HAAAAAAAB  NetApp ONTAP Controller  1
85.90 GB / 85.90 GB

Format          FW Rev
---------------------
4 KiB + 0 B   FFFFFFFF
```

Example (b):

```
# nvme list
Node            SN                         Model                         Namespace
Usage
----------------------------------------------------
------------------------------------
/dev/nvme0n1   81CZ5BQuUNfGAAAAAAAB NetApp ONTAP Controller   1
85.90 GB / 85.90 GB

Format          FW Rev
-----------------------
4 KiB + 0 B   FFFFFFFF
```

4. Verify that the controller state of each path is live and has a proper ANA status.

   Example (a):

```
# nvme list-subsys /dev/nvme0n1
nvme-subsys0 - NQN=nqn.1992-
08.com.netapp:sn.5f5f2c4aa73b11e9967e00a098df41bd:subsystem.nvme_141_1
\
+- nvme0 fc traddr=nn-0x203700a098dfdd91:pn-0x203800a098dfdd91
host_traddr=nn-0x200000109b1c1204:pn-0x100000109b1c1204 live
inaccessible
+- nvme1 fc traddr=nn-0x203700a098dfdd91:pn-0x203900a098dfdd91
host_traddr=nn-0x200000109b1c1204:pn-0x100000109b1c1204 live
inaccessible
+- nvme2 fc traddr=nn-0x203700a098dfdd91:pn-0x203a00a098dfdd91
host_traddr=nn-0x200000109b1c1205:pn-0x100000109b1c1205 live optimized
+- nvme3 fc traddr=nn-0x203700a098dfdd91:pn-0x203d00a098dfdd91
host_traddr=nn-0x200000109b1c1205:pn-0x100000109b1c1205 live optimized
```

Example (b):

```
# nvme list-subsys /dev/nvme0n1
nvme-subsys0 - NQN=nqn.1992-
08.com.netapp:sn.56e362e9bb4f11ebbaded039ea165abc:subsystem.nvme_118_tcp
_1
\
+- nvme0 tcp traddr=192.168.1.51 trsvcid=4420 host_traddr=192.168.1.8
live optimized
+- nvme10 tcp traddr=192.168.2.56 trsvcid=4420 host_traddr=192.168.2.9
live optimized
+- nvme15 tcp traddr=192.168.2.57 trsvcid=4420 host_traddr=192.168.2.9
live non-optimized
+- nvme5 tcp traddr=192.168.1.52 trsvcid=4420 host_traddr=192.168.1.8
live non-optimized
```

5. Verify the NetApp plug-in displays proper values for each ONTAP namespace device.

   Example (a):

```
# nvme netapp ontapdevices -o column
Device       Vserver        Namespace Path
NSID
----------------------- ------------------------------
-------------------------
/dev/nvme0n1  vs_fcnvme_141  /vol/fcnvme_141_vol_1_1_0/fcnvme_141_ns   1

UUID                                   Size
--------------------------------------------
72b887b1-5fb6-47b8-be0b-33326e2542e2   85.90GB
```

```
# nvme netapp ontapdevices -o json
{
"ONTAPdevices" : [
    {
        "Device" : "/dev/nvme0n1",
        "Vserver" : "vs_fcnvme_141",
        "Namespace_Path" : "/vol/fcnvme_141_vol_1_1_0/fcnvme_141_ns",
        "NSID" : 1,
        "UUID" : "72b887b1-5fb6-47b8-be0b-33326e2542e2",
        "Size" : "85.90GB",
        "LBA_Data_Size" : 4096,
        "Namespace_Size" : 20971520
    }
  ]
}
```

Example (b):

```
# nvme netapp ontapdevices -o column
Device               Vserver                   Namespace Path
--------------------- -------------------------
------------------------------------
/dev/nvme0n1         vs_tcp_118
/vol/tcpnvme_118_1_0_0/tcpnvme_118_ns

NSID   UUID                               Size
-----------------------------------------------
1     4a3e89de-b239-45d8-be0c-b81f6418283c 85.90GB
```

```
# nvme netapp ontapdevices -o json
{
"ONTAPdevices" : [
    {
     "Device" : "/dev/nvme0n1",
      "Vserver" : "vs_tcp_118",
      "Namespace_Path" : "/vol/tcpnvme_118_1_0_0/tcpnvme_118_ns",
      "NSID" : 1,
      "UUID" : "4a3e89de-b239-45d8-be0c-b81f6418283c",
      "Size" : "85.90GB",
      "LBA_Data_Size" : 4096,
      "Namespace_Size" : 20971520
    },
  ]

}
```

Known issues

The NVMe-oF host configuration for RHEL 9.0 with ONTAP has the following known issues:

NetApp Bug ID	Title	Description
1479047	RHEL 9.0 NVMe-oF hosts create duplicate Persistent Discovery Controllers	On NVMe over Fabrics (NVMe-oF) hosts, you can use the "nvme discover -p" command to create Persistent Discovery Controllers (PDCs). When this command is used, only one PDC should be created per initiator-target combination. However, if you are running ONTAP 9.10.1 and Red Hat Enterprise Linux (RHEL) 9.0 with an NVMe-oF host, a duplicate PDC is created each time "nvme discover -p" is executed. This leads to unnecessary usage of resources on both the host and the target.

RHEL 8

NVMe-oF host configuration for RHEL 8.10 with ONTAP

NVMe over Fabrics (NVMe-oF), including NVMe over Fibre Channel (NVMe/FC) and other transports, is supported with Red Hat Enterprise Linux (RHEL) 8.10 with Asymmetric Namespace Access (ANA). In NVMe-oF environments, ANA is the equivalent of ALUA multipathing in iSCSI and FC environments and is implemented with in-kernel NVMe multipath.

The following support is available for NVMe-oF host configuration for RHEL 8.10 with ONTAP:

* Support for NVMe over TCP (NVMe/TCP) in addition to NVMe/FC. The NetApp plug-in in the native `nvme-cli` package displays ONTAP details for both NVMe/FC and NVMe/TCP namespaces.

For additional details on supported configurations, see the Interoperability Matrix Tool.

Known limitations

* In-kernel NVMe multipath is disabled by default for RHEL 8.10 NVMe-oF hosts. Therefore, you need to enable it manually.
* On RHEL 8.10 hosts, NVMe/TCP is a technology preview feature due to open issues.
* SAN booting using the NVMe-oF protocol is currently not supported.

Enable SAN booting

You can configure your host to use SAN booting to simplify deployment and improve scalability.

Before you begin

Use the Interoperability Matrix Tool to verify that your Linux OS, host bus adapter (HBA), HBA firmware, HBA boot BIOS, and ONTAP version support SAN booting.

Steps

1. Create a SAN boot namespace and map it to the host.

   See Provision NVMe storage.

2. Enable SAN booting in the server BIOS for the ports to which the SAN boot LUN is mapped.

   For information on how to enable the HBA BIOS, see your vendor-specific documentation.

3. Verify that the configuration was successful by rebooting the host and verifying that the OS is up and running.

Enable in-kernel multipath

You can use the following procedure to enable in-kernel multipath.

Steps

1. Install RHEL 8.10 on the host server.
2. After the installation is complete, verify that you are running the specified RHEL 8.10 kernel:

   ```
   # uname -r
   ```

   Example output

   ```
   4.18.0-553.el8_10.x86_64
   ```

3. Install the `nvme-cli` package:

```
rpm -qa|grep nvme-cli
```

Example output

```
nvme-cli-1.16-9.el8.x86_64
```

4. Enable in-kernel NVMe multipath:

   Example

```
# grubby --args=nvme_core.multipath=Y --update-kernel
/boot/vmlinuz-4.18.0-553.el8_10.x86_64
```

5. On the host, check the host NQN string at `/etc/nvme/hostnqn`:

```
# cat /etc/nvme/hostnqn
```

   Example output

```
nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0032-3410-8035-b8c04f4c5132
```

6. Verify that the `hostnqn` string matches the `hostnqn` string for the corresponding subsystem on the ONTAP array:

```
::> vserver nvme subsystem host show -vserver vs_fcnvme_141
```

   Example output

```
Vserver     Subsystem       Host NQN
----------- ---------------
----------------------------------------------------------
vs_25_2742 rhel_101_QLe2772    nqn.2014-08.org.nvmexpress:uuid:546399fc-
160f-11e5-89aa-98be942440ca
```

   If the host NQN strings do not match, you can use the `vserver modify` command to update the host NQN string on your corresponding ONTAP NVMe subsystem to match the host NQN string `/etc/nvme/hostnqn` on the host.

7. Reboot the host.

If you intend to run both NVMe and SCSI co-existent traffic on the same host, NetApp recommends using the in-kernel NVMe multipath for ONTAP namespaces and dm-multipath for ONTAP LUNs respectively. This should exclude the ONTAP namespaces from dm-multipath and prevent dm-multipath from claiming these namespace devices. You can do this by adding the `enable_foreign` setting to the `/etc/multipath.conf` file:

```
# cat /etc/multipath.conf
defaults {
  enable_foreign  NONE
}
```

Configure NVMe/FC

You can configure NVMe/FC for Broadcom/Emulex or Marvell/Qlogic adapters.

Broadcom/Emulex

Steps

1. Verify that you are using the supported adapter model:

   ```
   # cat /sys/class/scsi_host/host*/modelname
   ```

   Example output:

   ```
   LPe32002-M2
   LPe32002-M2
   ```

   ```
   # cat /sys/class/scsi_host/host*/modeldesc
   ```

   Example output:

   ```
   Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
   Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
   ```

2. Verify that you are using the recommended Broadcom `lpfc` firmware and inbox driver:

   ```
   # cat /sys/class/scsi_host/host*/fwrev
   14.2.539.21, sli-4:2:c
   14.2.539.21, sli-4:2:c
   ```

   ```
   # cat /sys/module/lpfc/version
   0:14.0.0.21
   ```

   For the current list of supported adapter driver and firmware versions, see the Interoperability Matrix Tool.

3. Verify that `lpfc_enable_fc4_type` is set to `3`:

   ```
   # cat /sys/module/lpfc/parameters/lpfc_enable_fc4_type
   3
   ```

4. Verify that the initiator ports are up and running and that you can see the target LIFs:

```
# cat /sys/class/fc_host/host*/port_name
0x10000090fae0ec88
0x10000090fae0ec89
```

```
# cat /sys/class/fc_host/host*/port_state
Online
Online
```

```
# cat /sys/class/scsi_host/host*/nvme_info
NVME Initiator Enabled
XRI Dist lpfc0 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc0 WWPN x100000109bf044b1 WWNN x200000109bf044b1 DID
x022a00 ONLINE
NVME RPORT       WWPN x211ad039eaa7dfc8 WWNN x2119d039eaa7dfc8 DID
x021302 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x211cd039eaa7dfc8 WWNN x2119d039eaa7dfc8 DID
x020b02 TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 00000001ff Cmpl 00000001ff Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 0000000001330ec7 Issue 0000000001330ec9 OutIO
0000000000000002
        abort 00000330 noxri 00000000 nondlp 0000000b qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 00000354 Err 00000361

NVME Initiator Enabled
XRI Dist lpfc1 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc1 WWPN x100000109bf044b2 WWNN x200000109bf044b2 DID
x021b00 ONLINE
NVME RPORT       WWPN x211bd039eaa7dfc8 WWNN x2119d039eaa7dfc8 DID
x022902 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x211dd039eaa7dfc8 WWNN x2119d039eaa7dfc8 DID
x020102 TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 00000001ff Cmpl 00000001ff Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 00000000012ec220 Issue 00000000012ec222 OutIO
0000000000000002
        abort 0000033b noxri 00000000 nondlp 00000085 qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 00000368 Err 00000382
```

Marvell/QLogic FC Adapter for NVMe/FC

The native inbox qla2xxx driver included in the RHEL 8.10 GA kernel has the latest upstream fixes. These fixes are essential for ONTAP support.

Steps

1. Verify that you are running the supported adapter driver and firmware versions:

```
# cat /sys/class/fc_host/host*/symbolic_name
```

Example output

```
QLE2742 FW: v9.10.11 DVR: v10.02.08.200-k
QLE2742 FW: v9.10.11 DVR: v10.02.08.200-k
```

2. Verify that `ql2xnvmeenable` is set. This enables the Marvell adapter to function as an NVMe/FC initiator:

```
# cat /sys/module/qla2xxx/parameters/ql2xnvmeenable
1
```

Enable 1MB I/O (Optional)

ONTAP reports an MDTS (Max Data Transfer Size) of 8 in the Identify Controller data. This means the maximum I/O request size can be up to 1MB. To issue I/O requests of size 1MB for a Broadcom NVMe/FC host, you should increase the `lpfc` value of the `lpfc_sg_seg_cnt` parameter to 256 from the default value of 64.

> These steps don't apply to Qlogic NVMe/FC hosts.

Steps

1. Set the `lpfc_sg_seg_cnt` parameter to 256:

```
cat /etc/modprobe.d/lpfc.conf
```

You should see an output similar to the following example:

```
options lpfc lpfc_sg_seg_cnt=256
```

2. Run the `dracut -f` command, and reboot the host.
3. Verify that the value for `lpfc_sg_seg_cnt` is 256:

```
cat /sys/module/lpfc/parameters/lpfc_sg_seg_cnt
```

Configure NVMe/TCP

NVMe/TCP does not have auto-connect functionality. Therefore, if a path goes down and is not reinstated within the default time out period of 10 minutes, NVMe/TCP cannot automatically reconnect. To prevent a time out, you should set the retry period for failover events to at least 30 minutes.

Steps

1. Verify that the initiator port can fetch the discovery log page data across the supported NVMe/TCP LIFs:

   ```
   nvme discover -t tcp -w host-traddr -a traddr
   ```

   Example output:

   ```
   # nvme discover -t tcp -w 192.168.2.31 -a 192.168.2.25

   Discovery Log Number of Records 8, Generation counter 18
   =====Discovery Log Entry 0======
   trtype:  tcp
   adrfam:  ipv4
   subtype: unrecognized
   treq:    not specified.
   portid:  0
   trsvcid: 8009
   subnqn:  nqn.1992-
   08.com.netapp:sn.a1b2b785b9de11ee8e7fd039ea9e8ae9:discovery: discovery
   traddr:  192.168.1.25
   sectype: none
   =====Discovery Log Entry 1======
   trtype:  tcp
   adrfam:  ipv4
   subtype: unrecognized
   treq:    not specified.
   portid:  1
   trsvcid: 8009
   subnqn:  nqn.1992-
   08.com.netapp:sn.a1b2b785b9de11ee8e7fd039ea9e8ae9:discovery
   traddr:  192.168.2.26
   sectype: none ..........
   ```

2. Verify that the other NVMe/TCP initiator-target LIF combinations can successfully fetch discovery log page data:

   ```
   nvme discover -t tcp -w host-traddr -a traddr
   ```

   Example output:

```
# nvme  discover  -t  tcp  -w 192.168.2.31  -a 192.168.2.25
# nvme  discover  -t  tcp  -w 192.168.1.31  -a 192.168.1.24
# nvme  discover  -t  tcp  -w 192.168.2.31  -a 192.168.2.26
# nvme  discover  -t  tcp  -w 192.168.1.31  -a 192.168.1.25
```

3. Run the `nvme connect-all` command across all the supported NVMe/TCP initiator-target LIFs across the nodes, and set the controller loss timeout period for at least 30 minutes or 1800 seconds:

```
nvme connect-all -t tcp -w host-traddr -a traddr -l 1800
```

Example output:

```
# nvme  connect-all -t  tcp -w  192.168.2.31    -a  192.168.2.25    -l
1800
# nvme  connect-all -t  tcp -w  192.168.1.31    -a  192.168.1.24    -l
1800
# nvme  connect-all -t  tcp -w  192.168.2.31    -a  192.168.2.26    -l
1800
# nvme  connect-all -t  tcp -w  192.168.1.31    -a  192.168.1.25    -l
1800
```

Validate NVMe-oF

You can use the following procedure to validate NVMe-oF.

Steps

1. Verify that the in-kernel NVMe multipath is enabled:

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

2. Verify that the appropriate NVMe-oF settings (such as, `model` set to `NetApp ONTAP Controller` and load balancing `iopolicy` set to `round-robin`) for the respective ONTAP namespaces correctly reflect on the host:

```
# cat /sys/class/nvme-subsystem/nvme-subsys*/model
NetApp ONTAP Controller
NetApp ONTAP Controller
```

```
# cat /sys/class/nvme-subsystem/nvme-subsys*/iopolicy
round-robin
round-robin
```

3. Verify that the namespaces are created and correctly discovered on the host:

```
# nvme list
```

Example output:

```
Node         SN                   Model
---------------------------------------------------------
/dev/nvme0n1 81K1ABVnkwbNAAAAAAAB   NetApp ONTAP Controller


Namespace Usage    Format             FW            Rev
-----------------------------------------------------------
1                 21.47 GB / 21.47 GB   4 KiB + 0 B   FFFFFFFF
```

4. Verify that the controller state of each path is live and has the correct ANA status:

NVMe/FC

```
# nvme list-subsys /dev/nvme0n1
```

Example output:

```
nvme-subsys0 - NQN=nqn.1992-
08.com.netapp:sn.0cd9ee0dc0ec11ee8e7fd039ea9e8ae9:subsystem.nvme
\
 +- nvme1 fc traddr=nn-0x2005d039eaa7dfc8:pn-0x2086d039eaa7dfc8
host_traddr=nn-0x20000024ff752e6d:pn-0x21000024ff752e6d live non-
optimized
 +- nvme2 fc traddr=nn-0x2005d039eaa7dfc8:pn-0x2016d039eaa7dfc8
host_traddr=nn-0x20000024ff752e6c:pn-0x21000024ff752e6c live
optimized
 +- nvme3 fc traddr=nn-0x2005d039eaa7dfc8:pn-0x2081d039eaa7dfc8
host_traddr=nn-0x20000024ff752e6c:pn-0x21000024ff752e6c live non-
optimized
 +- nvme4 fc traddr=nn-0x2005d039eaa7dfc8:pn-0x2087d039eaa7dfc8
host_traddr=nn-0x20000024ff752e6d:pn-0x21000024ff752e6d live
optimized
```

NVMe/TCP

```
# nvme list-subsys /dev/nvme0n1
```

Example output:

```
nvme-subsys0 - NQN=nqn.1992-
08.com.netapp:sn.a1b2b785b9de11ee8e7fd039ea9e8ae9:subsystem.nvme_tcp
_1
\
 +- nvme0 tcp traddr=192.168.2.26 trsvcid=4420
host_traddr=192.168.2.31 live non-optimized
 +- nvme1 tcp traddr=192.168.2.25 trsvcid=4420
host_traddr=192.168.2.31 live optimized
 +- nvme2 tcp traddr=192.168.1.25 trsvcid=4420
host_traddr=192.168.1.31 live non-optimized
 +- nvme3 tcp traddr=192.168.1.24 trsvcid=4420
host_traddr=192.168.1.31 live optimized
```

5. Verify that the NetApp plug-in displays the correct values for each ONTAP namespace device:

Column

```
# nvme netapp ontapdevices -o column
```

Example output:

```
Device        Vserver       Namespace Path
-----------------------------------------------------
/dev/nvme0n1  tcpiscsi_129  /vol/tcpnvme_1_0_0/tcpnvme_ns



NSID       UUID                                   Size
------------------------------------------------------------
1          05c2c351-5d7f-41d7-9bd8-1a56c            21.47GB
```

JSON

```
# nvme netapp ontapdevices -o json
```

Example output

```
{
  "ONTAPdevices": [
    {
      "Device": "/dev/nvme0n1",
      "Vserver": "tcpiscsi_129",
      "Namespace Path": /vol/tcpnvme_1_0_0/tcpnvme_ns ",
      "NSID": 1,
      "UUID": " 05c2c351-5d7f-41d7-9bd8-1a56c160c80b ",
      "Size2: "21.47GB",
      "LBA_Data_Size": 4096,
      "Namespace Size" : 5242880
    },
  ]
}
```

Known issues

The NVMe-oF host configuration for RHEL 8.10 with ONTAP has the following known issue:

NetApp Bug ID	Title	Description
1479047	RHEL 8.10 NVMe-oF hosts create duplicate persistent discovery controllers	On NVMe over Fabrics (NVMe-oF) hosts, you can use the "nvme discover -p" command to create Persistent Discovery Controllers (PDCs). When this command is used, only one PDC should be created per initiator-target combination. However, if you are running Red Hat Enterprise Linux (RHEL) 8.10 on an NVMe-oF host, a duplicate PDC is created each time "nvme discover -p" is executed. This leads to unnecessary usage of resources on both the host and the target.

NVMe-oF host configuration for RHEL 8.9 with ONTAP

NVMe over Fabrics (NVMe-oF), including NVMe over Fibre Channel (NVMe/FC) and other transports, is supported with Red Hat Enterprise Linux (RHEL) 8.9 with Asymmetric Namespace Access (ANA). In NVMe-oF environments, ANA is the equivalent of ALUA multipathing in iSCSI and FC environments and is implemented with in-kernel NVMe multipath.

The following support is available for NVMe-oF host configuration for RHEL 8.9 with ONTAP:

- Support for NVMe over TCP (NVMe/TCP) in addition to NVMe/FC. The NetApp plug-in in the native nvme-cli package displays ONTAP details for both NVMe/FC and NVMe/TCP namespaces.

For additional details on supported configurations, see the Interoperability Matrix Tool.

Known limitations

- In-kernel NVMe multipath is disabled by default for RHEL 8.9 NVMe-oF hosts. Therefore, you need to enable it manually.
- On RHEL 8.9 hosts, NVMe/TCP is a technology preview feature due to open issues.
- SAN booting using the NVMe-oF protocol is currently not supported.

Enable SAN booting

You can configure your host to use SAN booting to simplify deployment and improve scalability.

Before you begin

Use the Interoperability Matrix Tool to verify that your Linux OS, host bus adapter (HBA), HBA firmware, HBA boot BIOS, and ONTAP version support SAN booting.

Steps

1. Create a SAN boot namespace and map it to the host.

   See Provision NVMe storage.

2. Enable SAN booting in the server BIOS for the ports to which the SAN boot LUN is mapped.

For information on how to enable the HBA BIOS, see your vendor-specific documentation.

3. Verify that the configuration was successful by rebooting the host and verifying that the OS is up and running.

Enable in-kernel multipath

You can use the following procedure to enable in-kernel multipath.

Steps

1. Install RHEL 8.9 on the host server.
2. After the installation is complete, verify that you are running the specified RHEL 8.9 kernel:

```
# uname -r
```

Example output

```
4.18.0-513.5.1.el8_9.x86_64
```

3. Install the nvme-cli package:

```
rpm -qa|grep nvme-cli
```

Example output

```
nvme-cli-1.16-9.el8.x86_64
```

4. Enable in -kernel NVMe multipath:

```
# grubby --args=nvme_core.multipath=Y --update-kernel /boot/vmlinuz-
4.18.0-513.5.1.el8_9.x86_64
```

5. On the host, check the host NQN string at `/etc/nvme/hostnqn`:

```
# cat /etc/nvme/hostnqn
```

Example output

```
nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0032-3410-8035-b8c04f4c5132
```

6. Verify that the `hostnqn` string matches the `hostnqn` string for the corresponding subsystem on the

ONTAP array:

```
::> vserver nvme subsystem host show -vserver vs_fcnvme_141
```

Example output

```
Vserver     Subsystem       Host NQN
----------- ---------------
----------------------------------------------------------
vs_nvme101 rhel_101_QLe2772    nqn.2014-08.org.nvmexpress:
uuid:4c4c4544-0032-3410-8035-b8c04f4c5132
```

> If the host NQN strings do not match, you can use the `vserver modify` command to update the host NQN string on your corresponding ONTAP NVMe subsystem to match the host NQN string `/etc/nvme/hostnqn` on the host.

7. Reboot the host.

> If you intend to run both NVMe and SCSI co-existent traffic on the same host, NetApp recommends using the in-kernel NVMe multipath for ONTAP namespaces and dm-multipath for ONTAP LUNs respectively. This should exclude the ONTAP namespaces from dm-multipath and prevent dm-multipath from claiming these namespace devices. You can do this by adding the `enable_foreign` setting to the `/etc/multipath.conf` file:
>
> ```
> # cat /etc/multipath.conf
> defaults {
>   enable_foreign  NONE
> }
> ```

Configure NVMe/FC

You can configure NVMe/FC for Broadcom/Emulex or Marvell/Qlogic adapters.

Broadcom/Emulex

Steps

1. Verify that you are using the supported adapter model:

   ```
   # cat /sys/class/scsi_host/host*/modelname
   ```

   Example output:

   ```
   LPe32002-M2
   LPe32002-M2
   ```

   ```
   # cat /sys/class/scsi_host/host*/modeldesc
   ```

   Example output:

   ```
   Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
   Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
   ```

2. Verify that you are using the recommended Broadcom `lpfc` firmware and inbox driver:

   ```
   # cat /sys/class/scsi_host/host*/fwrev
   14.2.539.16, sli-4:2:c
   14.2.539.16, sli-4:2:c
   ```

   ```
   # cat /sys/module/lpfc/version
   0:14.0.0.21
   ```

   For the current list of supported adapter driver and firmware versions, see the Interoperability Matrix Tool.

3. Verify that `lpfc_enable_fc4_type` is set to `3`:

   ```
   # cat /sys/module/lpfc/parameters/lpfc_enable_fc4_type
   3
   ```

4. Verify that the initiator ports are up and running and that you can see the target LIFs:

```
# cat /sys/class/fc_host/host*/port_name
0x10000090fae0ec88
0x10000090fae0ec89
```

```
# cat /sys/class/fc_host/host*/port_state
Online
Online
```

```
# cat /sys/class/scsi_host/host*/nvme_info
NVME Initiator Enabled
XRI Dist lpfc0 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc0 WWPN x10000090fae0ec88 WWNN x20000090fae0ec88 DID
x0a1300 ONLINE
NVME RPORT       WWPN x2049d039ea36a105 WWNN x2048d039ea36a105 DID
x0a0c0a TARGET DISCSRVC ONLINE
NVME Statistics
LS: Xmt 0000000024 Cmpl 0000000024 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 00000000000001aa Issue 00000000000001ab OutIO
0000000000000001
        abort 00000002 noxri 00000000 nondlp 00000000 qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 00000002 Err 00000003
NVME Initiator Enabled
XRI Dist lpfc1 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc1 WWPN x10000090fae0ec89 WWNN x20000090fae0ec89 DID
x0a1200 ONLINE
NVME RPORT       WWPN x204ad039ea36a105 WWNN x2048d039ea36a105 DID
x0a080a TARGET DISCSRVC ONLINE
NVME Statistics
LS: Xmt 0000000024 Cmpl 0000000024 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 00000000000001ac Issue 00000000000001ad OutIO
0000000000000001
        abort 00000002 noxri 00000000 nondlp 00000000 qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 00000002 Err 00000003
```

Marvell/QLogic FC Adapter for NVMe/FC

The native inbox qla2xxx driver included in the RHEL 8.9 GA kernel has the latest upstream fixes. These fixes are essential for ONTAP support.

Steps

1. Verify that you are running the supported adapter driver and firmware versions:

```
# cat /sys/class/fc_host/host*/symbolic_name
```

Example output

```
QLE2742 FW: v9.10.11 DVR: v10.02.08.200-k
QLE2742 FW: v9.10.11 DVR: v10.02.08.200-k
```

2. Verify that `ql2xnvmeenable` is set. This enables the Marvell adapter to function as an NVMe/FC initiator:

```
# cat /sys/module/qla2xxx/parameters/ql2xnvmeenable
1
```

Enable 1MB I/O (Optional)

ONTAP reports an MDTS (Max Data Transfer Size) of 8 in the Identify Controller data. This means the maximum I/O request size can be up to 1MB. To issue I/O requests of size 1MB for a Broadcom NVMe/FC host, you should increase the `lpfc` value of the `lpfc_sg_seg_cnt` parameter to 256 from the default value of 64.

> These steps don't apply to Qlogic NVMe/FC hosts.

Steps

1. Set the `lpfc_sg_seg_cnt` parameter to 256:

```
cat /etc/modprobe.d/lpfc.conf
```

You should see an output similar to the following example:

```
options lpfc lpfc_sg_seg_cnt=256
```

2. Run the `dracut -f` command, and reboot the host.
3. Verify that the value for `lpfc_sg_seg_cnt` is 256:

```
cat /sys/module/lpfc/parameters/lpfc_sg_seg_cnt
```

Configure NVMe/TCP

NVMe/TCP does not have auto-connect functionality. Therefore, if a path goes down and is not reinstated within the default time out period of 10 minutes, NVMe/TCP cannot automatically reconnect. To prevent a time out, you should set the retry period for failover events to at least 30 minutes.

Steps

1. Verify that the initiator port can fetch the discovery log page data across the supported NVMe/TCP LIFs:

   ```
   nvme discover -t tcp -w host-traddr -a traddr
   ```

   Example output:

   ```
   # nvme discover -t tcp -w 192.168.111.79 -a 192.168.111.14 -l 1800

   Discovery Log Number of Records 8, Generation counter 18
   =====Discovery Log Entry 0======
   trtype:  tcp
   adrfam:  ipv4
   subtype: unrecognized
   treq:    not specified.
   portid:  0
   trsvcid: 8009
   subnqn:  nqn.1992-08.com.netapp:sn.154a5833c78c11ecb069d039ea359e4b:
   discovery
   traddr:  192.168.211.15
   sectype: none
   =====Discovery Log Entry 1======
   trtype:  tcp
   adrfam:  ipv4
   subtype: unrecognized
   treq:    not specified.
   portid:  1
   trsvcid: 8009
   subnqn:  nqn.1992-08.com.netapp:sn.154a5833c78c11ecb069d039ea359e4b:
   discovery
   traddr:  192.168.111.15
   sectype: none ..........
   ```

2. Verify that the other NVMe/TCP initiator-target LIF combinations can successfully fetch discovery log page data:

   ```
   nvme discover -t tcp -w host-traddr -a traddr
   ```

   Example output:

```
# nvme  discover    -t   tcp    -w  192.168.111.79   -a 192.168.111.14
# nvme  discover    -t   tcp    -w  192.168.111.79   -a 192.168.111.15
# nvme  discover    -t   tcp    -w  192.168.211.79   -a 192.168.211.14
# nvme  discover    -t   tcp    -w  192.168.211.79   -a 192.168.211.15
```

3. Run the `nvme connect-all` command across all the supported NVMe/TCP initiator-target LIFs across the nodes, and set the controller loss timeout period for at least 30 minutes or 1800 seconds:

```
nvme connect-all -t tcp -w host-traddr -a traddr -l 1800
```

Example output:

```
# nvme  connect-all -t  tcp -w  192.168.111.79  -a  192.168.111.14  -l
1800
# nvme  connect-all -t  tcp -w  192.168.111.79  -a  192.168.111.15  -l
1800
# nvme  connect-all -t  tcp -w  192.168.211.79  -a  192.168.211.14  -l
1800
# nvme  connect-all -t  tcp -w  192.168.211.79  -a  192.168.211.15  -l
1800
```

Validate NVMe-oF

You can use the following procedure to validate NVMe-oF.

Steps

1. Verify that the in-kernel NVMe multipath is enabled:

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

2. Verify that the appropriate NVMe-oF settings (such as, `model` set to `NetApp ONTAP Controller` and load balancing `iopolicy` set to `round-robin`) for the respective ONTAP namespaces correctly reflect on the host:

```
# cat /sys/class/nvme-subsystem/nvme-subsys*/model
NetApp ONTAP Controller
NetApp ONTAP Controller
```

```
# cat /sys/class/nvme-subsystem/nvme-subsys*/iopolicy
round-robin
round-robin
```

3. Verify that the namespaces are created and correctly discovered on the host:

```
# nvme list
```

Example output:

```
Node         SN                   Model
---------------------------------------------------------
/dev/nvme0n1 81Gx7NSiKSQqAAAAAAAB   NetApp ONTAP Controller


Namespace Usage    Format             FW             Rev
-----------------------------------------------------------
1                 21.47 GB / 21.47 GB   4 KiB + 0 B   FFFFFFFF
```

4. Verify that the controller state of each path is live and has the correct ANA status:

NVMe/FC

```
# nvme list-subsys /dev/nvme3n1
```

Example output:

```
nvme-subsys0 - NQN=nqn.1992-
08.com.netapp:sn.8e501f8ebafa11ec9b99d039ea359e4b:subsystem.rhel_163
_Qle2742
+- nvme0 fc traddr=nn-0x204dd039ea36a105:pn-0x2050d039ea36a105
host_traddr=nn-0x20000024ff7f4994:pn-0x21000024ff7f4994 live non-
optimized
+- nvme1 fc traddr=nn-0x204dd039ea36a105:pn-0x2050d039ea36a105
host_traddr=nn-0x20000024ff7f4994:pn-0x21000024ff7f4994 live non-
optimized
+- nvme2 fc traddr=nn-0x204dd039ea36a105:pn-0x204fd039ea36a105
host_traddr=nn-0x20000024ff7f4995:pn-0x21000024ff7f4995 live
optimized
+- nvme3 fc traddr=nn-0x204dd039ea36a105:pn-0x204ed039ea36a105
host_traddr=nn-0x20000024ff7f4994:pn-0x21000024ff7f4994 live
optimized
```

NVMe/TCP

```
# nvme list-subsys /dev/nvme0n1
```

Example output:

```
nvme-subsys0 - NQN=nqn.1992-
08.com.netapp:sn.154a5833c78c11ecb069d039ea359e4b:subsystem.rhel_tcp
_165\
+- nvme0 tcp traddr=192.168.111.15 trsvcid=4420
host_traddr=192.168.111.79 live non-optimized
+- nvme1 tcp traddr=192.168.111.14 trsvcid=4420
host_traddr=192.168.111.79 live optimized
+- nvme2 tcp traddr=192.168.211.15 trsvcid=4420
host_traddr=192.168.211.79 live non-optimized
+- nvme3 tcp traddr=192.168.211.14 trsvcid=4420
host_traddr=192.168.211.79 live optimized
```

5. Verify that the NetApp plug-in displays the correct values for each ONTAP namespace device:

Column

```
# nvme netapp ontapdevices -o column
```

Example output:

```
Device        Vserver   Namespace Path
----------------------- ------------------------------
/dev/nvme0n1 vs_tcp79           /vol/vol1/ns


NSID       UUID                                   Size
------------------------------------------------------------
1          aa197984-3f62-4a80-97de-e89436360cec 21.47GB
```

JSON

```
# nvme netapp ontapdevices -o json
```

Example output

```
{
  "ONTAPdevices": [
    {
      "Device": "/dev/nvme0n1",
      "Vserver": "vs_tcp79",
      "Namespace Path": "/vol/vol1/ns",
      "NSID": 1,
      "UUID": "aa197984-3f62-4a80-97de-e89436360cec",
      "Size": "21.47GB",
      "LBA_Data_Size": 4096,
      "Namespace Size" : 5242880
    },
]

}
```

Known issues

The NVMe-oF host configuration for RHEL 8.9 with ONTAP release has the following known issue:

NetApp Bug ID	Title	Description
1479047	RHEL 8.9 NVMe-oF hosts create duplicate persistent discovery controllers	On NVMe over Fabrics (NVMe-oF) hosts, you can use the "nvme discover -p" command to create Persistent Discovery Controllers (PDCs). When this command is used, only one PDC should be created per initiator-target combination. However, if you are running Red Hat Enterprise Linux (RHEL) 8.9 on an NVMe-oF host, a duplicate PDC is created each time "nvme discover -p" is executed. This leads to unnecessary usage of resources on both the host and the target.

NVMe-oF host configuration for RHEL 8.8 with ONTAP

NVMe over Fabrics (NVMe-oF), including NVMe over Fibre Channel (NVMe/FC) and other transports, is supported with Red Hat Enterprise Linux (RHEL) 8.8 with Asymmetric Namespace Access (ANA). In NVMe-oF environments, ANA is the equivalent of ALUA multipathing in iSCSI and FC environments and is implemented with in-kernel NVMe multipath.

The following support is available for the NVMe-oF host configuration for RHEL 8.8 with ONTAP:

- Support for NVMe over TCP (NVMe/TCP) in addition to NVMe/FC. The NetApp plug-in in the native nvme-cli package displays ONTAP details for both NVMe/FC and NVMe/TCP namespaces.

For additional details on supported configurations, see the Interoperability Matrix Tool.

Known limitations

- In-kernel NVMe multipath is disabled by default for RHEL 8.8 NVMe-oF hosts. Therefore, you need to enable it manually.
- On RHEL 8.8 hosts, NVMe/TCP is a technology preview feature due to open issues.
- SAN booting using the NVMe-oF protocol is currently not supported.

Enable SAN booting

You can configure your host to use SAN booting to simplify deployment and improve scalability.

Before you begin

Use the Interoperability Matrix Tool to verify that your Linux OS, host bus adapter (HBA), HBA firmware, HBA boot BIOS, and ONTAP version support SAN booting.

Steps

1. Create a SAN boot namespace and map it to the host.

   See Provision NVMe storage.

2. Enable SAN booting in the server BIOS for the ports to which the SAN boot LUN is mapped.

For information on how to enable the HBA BIOS, see your vendor-specific documentation.

3. Verify that the configuration was successful by rebooting the host and verifying that the OS is up and running.

Enable in-kernel multipath

You can use the following procedure to enable in-kernal multipath.

Steps

1. Install RHEL 8.8 on the host server.
2. After the installation is complete, verify that you are running the specified RHEL 8.8 kernel.

```
# uname -r
```

Example output

```
4.18.0-477.10.1.el8_8.x86_64
```

3. Install the nvme-cli package:

```
rpm -qa|grep nvme-cli
```

Example output

```
nvme-cli-1.16-7.el8.x86_64
```

4. Enable in -kernel NVMe multipath:

```
# grubby --args=nvme_core.multipath=Y --update-kernel /boot/vmlinuz-
4.18.0-477.10.1.el8_8.x86_64
```

5. On the host, check the host NQN string at `/etc/nvme/hostnqn`:

```
# cat /etc/nvme/hostnqn
```

Example output

```
nqn.2014-08.org.nvmexpress:uuid:f6517cae-3133-11e8-bbff-7ed30aef123f
```

6. Verify that the `hostnqn` string matches the `hostnqn` string for the corresponding subsystem on the

ONTAP array:

```
::> vserver nvme subsystem host show -vserver vs_fcnvme_141
```

Example output

```
Vserver     Subsystem       Host NQN
----------- ---------------
----------------------------------------------------------
vs_nvme161 rhel_161_LPe32002    nqn.2014-
08.org.nvmexpress:uuid:f6517cae-3133-11e8-bbff-7ed30aef123f
```

> If the host NQN strings do not match, you can use the `vserver modify` command to update the host NQN string on your corresponding ONTAP NVMe subsystem to match the host NQN string `/etc/nvme/hostnqn` on the host.

7. Reboot the host.

> If you intend to run both NVMe and SCSI co-existent traffic on the same host, NetApp recommends using the in-kernel NVMe multipath for ONTAP namespaces and dm-multipath for ONTAP LUNs respectively. This means that the ONTAP namespaces should be excluded from dm-multipath to prevent dm-multipath from claiming these namespace devices. This can be done by adding the `enable_foreign` setting to the `/etc/multipath.conf` file:

```
# cat /etc/multipath.conf
defaults {
  enable_foreign  NONE
}
```

Configure NVMe/FC

You can configure NVMe/FC for Broadcom/Emulex or Marvell/Qlogic adapters.

Broadcom/Emulex

Steps

1. Verify that you are using the supported adapter model:

   ```
   # cat /sys/class/scsi_host/host*/modelname
   ```

   Example output:

   ```
   LPe32002-M2
   LPe32002-M2
   ```

   ```
   # cat /sys/class/scsi_host/host*/modeldesc
   ```

   Example output:

   ```
   Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
   Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
   ```

2. Verify that you are using the recommended Broadcom `lpfc` firmware and inbox driver:

   ```
   # cat /sys/class/scsi_host/host*/fwrev
   14.0.639.18, sli-4:2:c
   14.0.639.18, sli-4:2:c
   ```

   ```
   # cat /sys/module/lpfc/version
   0:14.0.0.18
   ```

   For the current list of supported adapter driver and firmware versions, see the Interoperability Matrix Tool.

3. Verify that `lpfc_enable_fc4_type` is set to `3`:

   ```
   # cat /sys/module/lpfc/parameters/lpfc_enable_fc4_type
   3
   ```

4. Verify that the initiator ports are up and running and that you can see the target LIFs:

```
# cat /sys/class/fc_host/host*/port_name
0x100000109b1c1204
0x100000109b1c1205
```

```
# cat /sys/class/fc_host/host*/port_state
Online
Online
```

```
# cat /sys/class/scsi_host/host*/nvme_info NVME Initiator Enabled
XRI Dist lpfc0 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc0 WWPN x10000090fae0ec88 WWNN x20000090fae0ec88 DID
x0a1300 ONLINE
NVME RPORT       WWPN x2049d039ea36a105 WWNN x2048d039ea36a105 DID
x0a0c0a TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x204bd039ea36a105 WWNN x2048d039ea36a105 DID
x0a100a TARGET DISCSRVC ONLINE
NVME Statistics
LS: Xmt 0000000134 Cmpl 0000000134 Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 00000000825e567 Issue 00000000825d7ed OutIO
fffffffffffff286
abort 0000027c noxri 00000000 nondlp 00000a02 qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 00000782 Err 000130fa

NVME Initiator Enabled
XRI Dist lpfc1 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc1 WWPN x10000090fae0ec89 WWNN x20000090fae0ec89 DID
x0a1200 ONLINE
NVME RPORT       WWPN x204ad039ea36a105 WWNN x2048d039ea36a105 DID
x0a080a TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x204cd039ea36a105 WWNN x2048d039ea36a105 DID
x0a090a TARGET DISCSRVC ONLINE
NVME Statistics
LS: Xmt 0000000134 Cmpl 0000000134 Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 00000000826ced5 Issue 00000000826c226 OutIO
fffffffffffff351
        abort 0000029d noxri 00000000 nondlp 000008df qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 00000821 Err 00012fcd
```

Marvell/QLogic FC Adapter for NVMe/FC

The native inbox qla2xxx driver included in the RHEL 8.8 GA kernel has the latest upstream fixes. These fixes are essential for ONTAP support.

Steps

Verify that you are running the supported adapter driver and firmware versions:

+

```
# cat /sys/class/fc_host/host*/symbolic_name
```

+
Example output

+

```
QLE2772 FW:v9.10.11 DVR:v10.02.07.900-k-debug
QLE2772 FW:v9.10.11 DVR:v10.02.07.900-k-debug
```

1. Verify that `ql2xnvmeenable` is set. This enables the Marvell adapter to function as an NVMe/FC initiator:

   ```
   # cat /sys/module/qla2xxx/parameters/ql2xnvmeenable
   1
   ```

Enable 1MB I/O (Optional)

ONTAP reports an MDTS (Max Data Transfer Size) of 8 in the Identify Controller data. This means the maximum I/O request size can be up to 1MB. To issue I/O requests of size 1MB for a Broadcom NVMe/FC host, you should increase the `lpfc` value of the `lpfc_sg_seg_cnt` parameter to 256 from the default value of 64.

> These steps don't apply to Qlogic NVMe/FC hosts.

Steps

1. Set the `lpfc_sg_seg_cnt` parameter to 256:

   ```
   cat /etc/modprobe.d/lpfc.conf
   ```

   You should see an output similar to the following example:

   ```
   options lpfc lpfc_sg_seg_cnt=256
   ```

2. Run the `dracut -f` command, and reboot the host.
3. Verify that the value for `lpfc_sg_seg_cnt` is 256:

```
cat /sys/module/lpfc/parameters/lpfc_sg_seg_cnt
```

Configure NVMe/TCP

NVMe/TCP does not have auto-connect functionality. Therefore, if a path goes down and is not reinstated within the default time out period of 10 minutes, NVMe/TCP cannot automatically reconnect. To prevent a time out, you should set the retry period for failover events to at least 30 minutes.

Steps

1. Verify that the initiator port can fetch the discovery log page data across the supported NVMe/TCP LIFs:

```
nvme discover -t tcp -w host-traddr -a traddr
```

Example output:

```
# nvme discover -t tcp -w 192.168.111.79 -a 192.168.111.14

Discovery Log Number of Records 8, Generation counter 10
=====Discovery Log Entry 0======
trtype:  tcp
adrfam:  ipv4
subtype: unrecognized
treq:    not specified
portid:  0
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.154a5833c78c11ecb069d039ea359e4b:discovery
traddr:  192.168.211.15
sectype: none
=====Discovery Log Entry 1======
trtype:  tcp
adrfam:  ipv4
subtype: unrecognized
treq:    not specified
portid:  1
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.154a5833c78c11ecb069d039ea359e4b:discovery
traddr:  192.168.111.15
sectype: none
=====Discovery Log Entry 2======
trtype:  tcp
adrfam:  ipv4
subtype: unrecognized
treq:    not specified
portid:  2
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.154a5833c78c11ecb069d039ea359e4b:discovery
traddr:  192.168.211.14
sectype: none
..........
```

2. Verify that the other NVMe/TCP initiator-target LIF combinations can successfully fetch discovery log page data:

```
nvme discover -t tcp -w host-traddr -a traddr
```

Example output:

```
# nvme  discover    -t   tcp    -w  192.168.111.79   -a 192.168.111.14
# nvme  discover    -t   tcp    -w  192.168.111.79   -a 192.168.111.15
# nvme  discover    -t   tcp    -w  192.168.211.79   -a 192.168.211.14
# nvme  discover    -t   tcp    -w  192.168.211.79   -a 192.168.211.15
```

3. Run the `nvme connect-all` command across all the supported NVMe/TCP initiator-target LIFs across the nodes, and set the controller loss timeout period for at least 30 minutes or 1800 seconds:

```
nvme connect-all -t tcp -w host-traddr -a traddr -l 1800
```

Example output:

```
# nvme  connect-all -t  tcp -w  192.168.111.79  -a  192.168.111.14  -l
1800
# nvme  connect-all -t  tcp -w  192.168.111.79  -a  192.168.111.15  -l
1800
# nvme  connect-all -t  tcp -w  192.168.211.79  -a  192.168.211.14  -l
1800
# nvme  connect-all -t  tcp -w  192.168.211.79  -a  192.168.211.15  -l
1800
```

Validate NVMe-oF

You can use the following procedure to validate NVME-oF.

Steps

1. Verify that the in-kernel NVMe multipath is enabled:

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

2. Verify that the appropriate NVMe-oF settings (such as, `model` set to `NetApp ONTAP Controller` and load balancing `iopolicy` set to `round-robin`) for the respective ONTAP namespaces correctly reflect on the host:

```
# cat /sys/class/nvme-subsystem/nvme-subsys*/model
NetApp ONTAP Controller
NetApp ONTAP Controller
```

```
# cat /sys/class/nvme-subsystem/nvme-subsys*/iopolicy
round-robin
round-robin
```

3. Verify that the namespaces are created and correctly discovered on the host:

```
# nvme list
```

Example output:

```
Node         SN                   Model
---------------------------------------------------------
/dev/nvme3n1 81Gx7NSiKSQeAAAAAAAB   NetApp ONTAP Controller


Namespace Usage    Format             FW            Rev
-----------------------------------------------------------
1                 21.47 GB / 21.47 GB   4 KiB + 0 B   FFFFFFFF
```

4. Verify that the controller state of each path is live and has the correct ANA status:

NVMe/FC

```
# nvme list-subsys /dev/nvme3n1
```

Example output:

```
nvme-subsys3 - NQN=nqn.1992-
08.com.netapp:sn.ab4fa6a5ba8b11ecbe3dd039ea359e4b:subsystem.rhel_161
_Lpe32002
\
 +- nvme0 fc traddr=nn-0x2048d039ea36a105:pn-0x204cd039ea36a105
host_traddr=nn-0x20000090fae0ec89:pn-0x10000090fae0ec89 live non-
optimized
 +- nvme1 fc traddr=nn-0x2048d039ea36a105:pn-0x204ad039ea36a105
host_traddr=nn-0x20000090fae0ec89:pn-0x10000090fae0ec89 live
optimized
 +- nvme2 fc traddr=nn-0x2048d039ea36a105:pn-0x204bd039ea36a105
host_traddr=nn-0x20000090fae0ec88:pn-0x10000090fae0ec88 live non-
optimized
 +- nvme4 fc traddr=nn-0x2048d039ea36a105:pn-0x2049d039ea36a105
host_traddr=nn-0x20000090fae0ec88:pn-0x10000090fae0ec88 live
optimized
```

NVMe/TCP

```
# nvme list-subsys /dev/nvme0n1
```

Example output:

```
nvme-subsys0 - NQN=nqn.1992-
08.com.netapp:sn.154a5833c78c11ecb069d039ea359e4b:subsystem.rhel_tcp
_165
\
 +- nvme0 tcp traddr=192.168.111.15 trsvcid=4420
host_traddr=192.168.111.79 live non-optimized
 +- nvme1 tcp traddr=192.168.111.14 trsvcid=4420
host_traddr=192.168.111.79 live optimized
 +- nvme2 tcp traddr=192.168.211.15 trsvcid=4420
host_traddr=192.168.211.79 live non-optimized
```

5. Verify that the NetApp plug-in displays the correct values for each ONTAP namespace device:

Column

```
# nvme netapp ontapdevices -o column
```

Example output:

```
Device        Vserver   Namespace Path
----------------------- ------------------------------
/dev/nvme0n1 vs_tcp           /vol/vol1/ns1



NSID       UUID                                   Size
------------------------------------------------------------
1          338d73ce-b5a8-4847-9cc9-b127c75d8855 21.47GB
```

JSON

```
# nvme netapp ontapdevices -o json
```

Example output

```
{
  "ONTAPdevices" : [
    {
      "Device" : "/dev/nvme0n1",
      "Vserver" : "vs_tcp79",
      "Namespace_Path" : "/vol/vol1/ns1",
      "NSID" : 1,
      "UUID" : "338d73ce-b5a8-4847-9cc9-b127c75d8855",
      "Size" : "21.47GB",
      "LBA_Data_Size" : 4096,
      "Namespace_Size" : 5242880
    },
]

}
```

Known issues

The NVMe-oF host configuration for RHEL 8.8 with ONTAP release has the following known issues:

NetApp Bug ID	Title	Description
1479047	RHEL 8.8 NVMe-oF hosts create duplicate persistent discovery controllers	On NVMe over Fabrics (NVMe-oF) hosts, you can use the "nvme discover -p" command to create Persistent Discovery Controllers (PDCs). When this command is used, only one PDC should be created per initiator-target combination. However, if you are running Red Hat Enterprise Linux (RHEL) 8.8 on an NVMe-oF host, a duplicate PDC is created each time "nvme discover -p" is executed. This leads to unnecessary usage of resources on both the host and the target.

NVMe-oF host configuration for RHEL 8.7 with ONTAP

NVMe over Fabrics or NVMe-oF (including NVMe/FC and other transports) is supported with Red Hat Enterprise Linux (RHEL) 8.7 with ANA (Asymmetric Namespace Access). ANA is the asymmetric logical unit access (ALUA) equivalent in the NVMe-oF environment, and is currently implemented with in-kernel NVMe Multipath. During this procedure, you enable NVMe-oF with in-kernel NVMe Multipath using ANA on RHEL 8.7 and ONTAP as the target.

See the Interoperability Matrix Tool for accurate details regarding supported configurations.

Features

RHEL 8.7 includes support for NVMe/TCP (as a Technology Preview feature) in addition to NVMe/FC. The NetApp plugin in the native nvme-cli package is capable of displaying ONTAP details for both NVMe/FC and NVMe/TCP namespaces.

Known limitations

- For RHEL 8.7, in-kernel NVMe multipath remains disabled by default. Therefore, you need to enable it manually.
- NVMe/TCP on RHEL 8.7 remains a Technology Preview feature due to open issues. Refer to the RHEL 8.7 release notes for details.
- SAN booting using the NVMe-oF protocol is currently not supported.

Enable SAN booting

You can configure your host to use SAN booting to simplify deployment and improve scalability.

Before you begin

Use the Interoperability Matrix Tool to verify that your Linux OS, host bus adapter (HBA), HBA firmware, HBA boot BIOS, and ONTAP version support SAN booting.

Steps

1. Create a SAN boot namespace and map it to the host.

   See Provision NVMe storage.

2. Enable SAN booting in the server BIOS for the ports to which the SAN boot LUN is mapped.

   For information on how to enable the HBA BIOS, see your vendor-specific documentation.

3. Verify that the configuration was successful by rebooting the host and verifying that the OS is up and running.

Enable in-kernel NVMe Multipath

You can use the following procedure to enable in-kernel NVMe multipath.

Steps

1. Install RHEL 8.7 on the server.
2. After the installation is complete, verify that you are running the specified RHEL 8.7 kernel. See the Interoperability Matrix Tool for the current list of supported versions.

   Example:

   ```
   # uname -r
   4.18.0-425.3.1.el8.x86_64
   ```

3. Install the `nvme-cli` package:

   Example:

   ```
   # rpm -qa|grep nvme-cli
   nvme-cli-1.16-5.el8.x86_64
   ```

4. Enable in-kernel NVMe multipath:

   Example

   ```
   # grubby --args=nvme_core.multipath=Y --update-kernel
   /boot/vmlinuz-4.18.0-425.3.1.el8.x86_64
   ```

5. On the host, check the host NQN string at `/etc/nvme/hostnqn` and verify that it matches the host NQN string for the corresponding subsystem on the ONTAP array. Example:

```
# cat /etc/nvme/hostnqn

          nqn.2014-08.org.nvmexpress:uuid:a7f7a1d4-311a-11e8-b634-
7ed30aef10b7

::> vserver nvme subsystem host show -vserver vs_nvme167
Vserver     Subsystem       Host NQN
----------- --------------- ----------------
vs_nvme167 rhel_167_LPe35002  nqn.2014-08.org.nvmexpress:uuid: a7f7a1d4-
311a-11e8-b634-7ed30aef10b7
```

> If the host NQN strings do not match, you should use the `vserver modify` command to update the host NQN string on your corresponding ONTAP NVMe subsystem to match the host NQN string `/etc/nvme/hostnqn` on the host.

6. Reboot the host.

> If you intend to run both NVMe and SCSI co-existent traffic on the same host, NetApp recommends using in-kernel NVMe multipath for ONTAP namespaces and dm-multipath for ONTAP LUNs respectively. This means that the ONTAP namespaces should be excluded from dm-multipath to prevent dm-multipath from claiming these namespace devices. You can do this by adding the enable_foreign setting to the `/etc/multipath.conf` file:
>
> ```
> # cat /etc/multipath.conf
> defaults {
>         enable_foreign     NONE
> }
> ```
>
> Restart the multipathd daemon by running a `systemctl restart multipathd` command to allow the new setting to take effect.

Configure NVMe/FC

You can configure NVMe/FC for Broadcom/Emulex or Marvell/Qlogic adapters.

Broadcom/Emulex

Steps

1. Verify that you are using the supported adapter. See the Interoperability Matrix Tool for the current list of supported adapters.

```
# cat /sys/class/scsi_host/host*/modelname
LPe35002-M2
LPe35002-M2
# cat /sys/class/scsi_host/host*/modeldesc
Emulex LightPulse LPe35002-M2 2-Port 32Gb Fibre Channel Adapter
Emulex LightPulse LPe35002-M2 2-Port 32Gb Fibre Channel Adapter
```

2. Verify that you are using the recommended Broadcom lpfc firmware and inbox driver. See the Interoperability Matrix Tool for the current list of supported adapter driver and firmware versions.

```
# cat /sys/class/scsi_host/host*/fwrev
14.0.505.12, sli-4:6:d
14.0.505.12, sli-4:6:d
# cat /sys/module/lpfc/version
0:14.0.0.15
```

3. Verify that `lpfc_enable_fc4_type` is set to 3

```
# cat /sys/module/lpfc/parameters/lpfc_enable_fc4_type
3
```

4. Verify that the initiator ports are up and running, and that you can see the target LIFs.

```
# cat /sys/class/fc_host/host*/port_name
0x100000109b95467c
0x100000109b95467b
# cat /sys/class/fc_host/host*/port_state
Online
Online
# cat /sys/class/scsi_host/host*/nvme_info
NVME Initiator Enabled
XRI Dist lpfc1 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc1 WWPN x100000109b95467c WWNN x200000109b95467c DID
x0a1500 ONLINE
NVME RPORT       WWPN x2071d039ea36a105 WWNN x206ed039ea36a105 DID
x0a0907 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2072d039ea36a105 WWNN x206ed039ea36a105 DID
x0a0805 TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 00000001c7 Cmpl 00000001c7 Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 0000000004909837 Issue 0000000004908cfc OutIO
fffffffffffff4c5
abort 0000004a noxri 00000000 nondlp 00000458 qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 00000061 Err 00017f43

NVME Initiator Enabled
XRI Dist lpfc0 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc0 WWPN x100000109b95467b WWNN x200000109b95467b DID
x0a1100 ONLINE
NVME RPORT       WWPN x2070d039ea36a105 WWNN x206ed039ea36a105 DID
x0a1007 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x206fd039ea36a105 WWNN x206ed039ea36a105 DID
x0a0c05 TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 00000001c7 Cmpl 00000001c7 Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 0000000004909464 Issue 0000000004908531 OutIO
fffffffffffff0cd
abort 0000004f noxri 00000000 nondlp 00000361 qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 0000006b Err 00017f99
```

Marvell/QLogic FC adapter for NVMe/FC

The native inbox `qla2xxx` driver included in the RHEL 8.7 kernel has the latest fixes. These fixes are essential for ONTAP support.

Steps

1. Verify that you are running the supported adapter driver and firmware versions using the following command:

   ```
   # cat /sys/class/fc_host/host*/symbolic_name
   QLE2772 FW:v9.08.02 DVR:v10.02.07.400-k-debug
   QLE2772 FW:v9.08.02 DVR:v10.02.07.400-k-debug
   ```

2. Verify `ql2xnvmeenable` is set, which enables the Marvell adapter to function as a NVMe/FC initiator using the following command:

   ```
   # cat /sys/module/qla2xxx/parameters/ql2xnvmeenable
   1
   ```

Enable 1MB I/O (Optional)

ONTAP reports an MDTS (Max Data Transfer Size) of 8 in the Identify Controller data. This means the maximum I/O request size can be up to 1MB. To issue I/O requests of size 1MB for a Broadcom NVMe/FC host, you should increase the `lpfc` value of the `lpfc_sg_seg_cnt` parameter to 256 from the default value of 64.

These steps don't apply to Qlogic NVMe/FC hosts.

Steps

1. Set the `lpfc_sg_seg_cnt` parameter to 256:

   ```
   cat /etc/modprobe.d/lpfc.conf
   ```

   You should see an output similar to the following example:

   ```
   options lpfc lpfc_sg_seg_cnt=256
   ```

2. Run the `dracut -f` command, and reboot the host.

3. Verify that the value for `lpfc_sg_seg_cnt` is 256:

   ```
   cat /sys/module/lpfc/parameters/lpfc_sg_seg_cnt
   ```

Configure NVMe/TCP

NVMe/TCP does not have auto-connect functionality. Therefore, if a path goes down and is not reinstated within the default time out period of 10 minutes, NVMe/TCP cannot automatically reconnect. To prevent a time out, you should set the retry period for failover events to at least 30 minutes.

Steps

1. Verify whether the initiator port can fetch the discovery log page data across the supported NVMe/TCP LIFs:

```
# nvme discover -t tcp -w 192.168.211.5 -a 192.168.211.14

Discovery Log Number of Records 8, Generation counter 10

=====Discovery Log Entry 0======
trtype:  tcp
adrfam:  ipv4
subtype: unrecognized
treq:    not specified
portid:  0
trsvcid: 8009
subnqn:
nqn.199208.com.netapp:sn.154a5833c78c11ecb069d039ea359e4b:discovery
traddr:  192.168.211.15
sectype: none
=====Discovery Log Entry 1======
trtype:  tcp
adrfam:  ipv4
subtype: unrecognized
treq:    not specified
portid:  1
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.154a5833c78c11ecb069d039ea359e4b:discovery
traddr:  192.168.111.15
sectype: none
=====Discovery Log Entry 2======
trtype:  tcp
adrfam:  ipv4
subtype: unrecognized
treq:    not specified
portid:  2
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.154a5833c78c11ecb069d039ea359e4b:discovery
traddr:  192.168.211.14
sectype: none
```

```
=====Discovery Log Entry 3======
trtype:  tcp
adrfam:  ipv4
subtype: unrecognized
treq:    not specified
portid:  3
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.154a5833c78c11ecb069d039ea359e4b:discovery
traddr:  192.168.111.14
sectype: none
=====Discovery Log Entry 4======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  0
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.154a5833c78c11ecb069d039ea359e4b:subsystem.rhel_tcp_165
traddr:  192.168.211.15
sectype: none
=====Discovery Log Entry 5======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.154a5833c78c11ecb069d039ea359e4b:subsystem.rhel_tcp_165
traddr:  192.168.111.15
sectype: none
=====Discovery Log Entry 6======

trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.154a5833c78c11ecb069d039ea359e4b:subsystem.rhel_tcp_165
traddr:  192.168.211.14
sectype: none
```

```
=====Discovery Log Entry 7======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified

   portid:  3

trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.154a5833c78c11ecb069d039ea359e4b:subsystem.rhel_tcp_165
traddr:  192.168.111.14
sectype: none
[root@R650-13-79 ~]#
```

2. Verify that other NVMe/TCP initiator-target LIF combos can successfully fetch discovery log page data. For example:

```
# nvme discover -t tcp -w 192.168.211.5 -a 192.168.211.14
# nvme discover -t tcp -w 192.168.211.5 -a 192.168.211.15
# nvme discover -t tcp -w 192.168.111.5 -a 192.168.111.14
# nvme discover -t tcp -w 192.168.111.5 -a 192.168.111.15
```

3. Run `nvme connect-all` command across all the supported NVMe/TCP initiator-target LIFs across the nodes. Ensure you set a longer `ctrl_loss_tmo` timer retry period (for example, 30 minutes, which can be set through `-l 1800`) during the connect-all so that it would retry for a longer period of time in the event of a path loss. For example:

```
# nvme connect-all -t tcp -w 192.168.211.5-a 192.168.211.14 -l 1800
# nvme connect-all -t tcp -w 192.168.211.5 -a 192.168.211.15 -l 1800
# nvme connect-all -t tcp -w 192.168.111.5 -a 192.168.111.14 -l 1800
# nvme connect-all -t tcp -w 192.168.111.5 -a 192.168.111.15 -l 1800
```

Validate NVMe-oF

You can use the following procedure to validate NVMe-oF.

Steps

1. Verify that in-kernel NVMe multipath is indeed enabled by checking:

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

2. Verify that the appropriate NVMe-oF settings (such as, `model` set to `NetApp ONTAP Controller` and

load balancing `iopolicy` set to `round-robin`) for the respective ONTAP namespaces properly reflect on the host:

```
# cat /sys/class/nvme-subsystem/nvme-subsys*/model
NetApp ONTAP Controller
NetApp ONTAP Controller

# cat /sys/class/nvme-subsystem/nvme-subsys*/iopolicy
round-robin
round-robin
```

3. Verify that the ONTAP namespaces properly reflect on the host. For example:

```
# nvme list
Node           SN                   Model                        Namespace
------------   -------------------- ---------------------------------
/dev/nvme0n1   81Gx7NSiKSRNAAAAAAAB   NetApp ONTAP Controller   1

Usage                Format         FW Rev
-------------------  -----------    --------
21.47  GB /  21.47  GB  4 KiB + 0 B    FFFFFFFF
```

4. Verify that the controller state of each path is live and has proper ANA status. For example:

```
# nvme list-subsys /dev/nvme1n1

nvme-subsys0 - NQN=nqn.1992-
08.com.netapp:sn.154a5833c78c11ecb069d039ea359e4b:subsystem.rhel_tcp_165

\

 +- nvme0 tcp traddr=192.168.211.15 trsvcid=4420
host_traddr=192.168.211.5 live non-optimized

 +- nvme1 tcp traddr=192.168.211.14 trsvcid=4420
host_traddr=192.168.211.5 live optimized

 +- nvme2 tcp traddr=192.168.111.15 trsvcid=4420
host_traddr=192.168.111.5 live non-optimized

 +- nvme3 tcp traddr=192.168.111.14 trsvcid=4420
host_traddr=192.168.111.5 live optimized
```

5. Verify that the NetApp plug-in displays proper values for each ONTAP namespace device. For example:

```
# nvme netapp ontapdevices -o column
Device        Vserver          Namespace Path
---------     -------
--------------------------------------------------
/dev/nvme0n1 vs_tcp79     /vol/vol1/ns1

NSID  UUID                                   Size
----  ------------------------------         ------
1     79c2c569-b7fa-42d5-b870-d9d6d7e5fa84  21.47GB


# nvme netapp ontapdevices -o json
{

  "ONTAPdevices" : [
  {

      "Device" : "/dev/nvme0n1",
      "Vserver" : "vs_tcp79",
      "Namespace_Path" : "/vol/vol1/ns1",
      "NSID" : 1,
      "UUID" : "79c2c569-b7fa-42d5-b870-d9d6d7e5fa84",
      "Size" : "21.47GB",
      "LBA_Data_Size" : 4096,
      "Namespace_Size" : 5242880
    },

]

}
```

Known issues

The NVMe-oF host configuration for RHEL 8.7 with ONTAP has the following known issues:

NetApp Bug ID	Title	Description
1479047	RHEL 8.7 NVMe-oF hosts create duplicate Persistent Discovery Controllers	On NVMe over Fabrics (NVMe-oF) hosts, you can use the "nvme discover -p" command to create Persistent Discovery Controllers (PDCs). When this command is used, only one PDC should be created per initiator-target combination. However, if you are running ONTAP 9.10.1 and Red Hat Enterprise Linux (RHEL) 8.7 with an NVMe-oF host, a duplicate PDC is created each time "nvme discover -p" is executed. This leads to unnecessary usage of resources on both the host and the target.

NVMe-oF host configuration for RHEL 8.6 with ONTAP

NVMe over Fabrics or NVMe-oF (including NVMe/FC and other transports) is supported with Red Hat Enterprise Linux (RHEL) 8.6 with ANA (Asymmetric Namespace Access). ANA is the asymmetric logical unit access (ALUA) equivalent in the NVMe-oF environment, and is currently implemented with in-kernel NVMe Multipath. During this procedure, you enable NVMe-oF with in-kernel NVMe Multipath using ANA on RHEL 8.6 and ONTAP as the target

See the Interoperability Matrix Tool for accurate details regarding supported configurations.

Features

- RHEL 8.6 includes support for NVMe/TCP (as a Technology Preview feature) in addition to NVMe/FC. The NetApp plugin in the native nvme-cli package is capable of displaying ONTAP details for both NVMe/FC and NVMe/TCP namespaces.

Known limitations

- For RHEL 8.6, in-kernel NVMe multipath remains disabled by default. Therefore, you need to enable it manually.
- NVMe/TCP on RHEL 8.6 remains a Technology Preview feature due to open issues. Refer to the RHEL 8.6 Release Notes for details.
- SAN booting using the NVMe-oF protocol is currently not supported.

Enable SAN booting

You can configure your host to use SAN booting to simplify deployment and improve scalability.

Before you begin

Use the Interoperability Matrix Tool to verify that your Linux OS, host bus adapter (HBA), HBA firmware, HBA boot BIOS, and ONTAP version support SAN booting.

Steps

1. Create a SAN boot namespace and map it to the host.

   See Provision NVMe storage.

2. Enable SAN booting in the server BIOS for the ports to which the SAN boot LUN is mapped.

   For information on how to enable the HBA BIOS, see your vendor-specific documentation.

3. Verify that the configuration was successful by rebooting the host and verifying that the OS is up and running.

Enable in-kernel NVMe Multipath

You can use the following procedure to enable in-kernel NVMe multipath.

Steps

1. Install RHEL 8.6 on the server. After the installation is complete, verify that you are running the specified RHEL 8.6 kernel. See the Interoperability Matrix Tool for the current list of supported versions.
2. After the installation is complete, verify that you are running the specified RHEL 8.6 kernel. See the Interoperability Matrix Tool for the current list of supported versions.

   Example:

```
# uname -r
4.18.0-372.9.1.el8.x86_64
```

3. Install the `nvme-cli` package:

   Example:

```
# rpm -qa|grep nvme-cli
nvme-cli-1.16-3.el8.x86_64
```

4. Enable in-kernel NVMe multipath:

```
# grubby --args=nvme_core.multipath=Y --update-kernel /boot/vmlinuz-
4.18.0-372.9.1.el8.x86_64
```

5. On the host, check the host NQN string at `/etc/nvme/hostnqn` and verify that it matches the host NQN string for the corresponding subsystem on the ONTAP array. Example:

```
# cat /etc/nvme/hostnqn
nqn.2014-08.org.nvmexpress:uuid:9ed5b327-b9fc-4cf5-97b3-1b5d986345d1
::> vserver nvme subsystem host show -vserver vs_fcnvme_141
Vserver     Subsystem       Host NQN
----------- ---------------
----------------------------------------------------------
vs_fcnvme_14 nvme_141_1     nqn.2014-08.org.nvmexpress:uuid:9ed5b327-
b9fc-4cf5-97b3-1b5d986345d1
```

> If the host NQN strings do not match, you should use the `vserver modify` command to update the host NQN string on your corresponding ONTAP NVMe subsystem to match the host NQN string `/etc/nvme/hostnqn` on the host.

6. Reboot the host.

> If you intend to run both NVMe and SCSI co-existent traffic on the same host, NetApp recommends using in-kernel NVMe multipath for ONTAP namespaces and dm-multipath for ONTAP LUNs respectively. This means that the ONTAP namespaces should be excluded from dm-multipath to prevent dm-multipath from claiming these namespace devices. This can be done by adding the enable_foreign setting to the `/etc/multipath.conf` file:
>
> ```
> # cat /etc/multipath.conf
> defaults {
>         enable_foreign     NONE
> }
> ```
>
> Restart the multipathd daemon by running a `systemctl restart multipathd` command to allow the new setting to take effect.

Configure NVMe/FC

You can configure NVMe/FC for Broadcom/Emulex or Marvell/Qlogic adapters.

Broadcom/Emulex

Steps

1. Verify that you are using the supported adapter. See the Interoperability Matrix Tool for the current list of supported adapters.

```
# cat /sys/class/scsi_host/host*/modelname
LPe32002-M2
LPe32002-M2
# cat /sys/class/scsi_host/host*/modeldesc
Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
```

2. Verify that you are using the recommended Broadcom lpfc firmware and inbox driver. See the Interoperability Matrix Tool for the current list of supported adapter driver and firmware versions.

```
# cat /sys/class/scsi_host/host*/fwrev
12.8.351.47, sli-4:2:c
12.8.351.47, sli-4:2:c
# cat /sys/module/lpfc/version
0:14.0.0.4
```

3. Verify that `lpfc_enable_fc4_type` is set to 3

```
# cat /sys/module/lpfc/parameters/lpfc_enable_fc4_type
3
```

4. Verify that the initiator ports are up and running, and that you can see the target LIFs.

```
# cat /sys/class/fc_host/host*/port_name
0x100000109b1c1204
0x100000109b1c1205
# cat /sys/class/fc_host/host*/port_state
Online
Online
# cat /sys/class/scsi_host/host*/nvme_info
NVME Initiator Enabled
XRI Dist lpfc0 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc0 WWPN x100000109b1c1204 WWNN x200000109b1c1204 DID
x011d00 ONLINE
NVME RPORT WWPN x203800a098dfdd91 WWNN x203700a098dfdd91 DID x010c07
TARGET DISCSRVC ONLINE
NVME RPORT WWPN x203900a098dfdd91 WWNN x203700a098dfdd91 DID x011507
TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 0000000f78 Cmpl 0000000f78 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000002fe29bba Issue 000000002fe29bc4 OutIO
000000000000000a
abort 00001bc7 noxri 00000000 nondlp 00000000 qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 00001e15 Err 0000d906

NVME Initiator Enabled
XRI Dist lpfc1 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc1 WWPN x100000109b1c1205 WWNN x200000109b1c1205 DID
x011900 ONLINE
NVME RPORT WWPN x203d00a098dfdd91 WWNN x203700a098dfdd91 DID x010007
TARGET DISCSRVC ONLINE
NVME RPORT WWPN x203a00a098dfdd91 WWNN x203700a098dfdd91 DID x012a07
TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 0000000fa8 Cmpl 0000000fa8 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000002e14f170 Issue 000000002e14f17a OutIO
000000000000000a
abort 000016bb noxri 00000000 nondlp 00000000 qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 00001f50 Err 0000d9f8
```

Marvell/QLogic FC adapter for NVMe/FC

The native inbox qla2xxx driver included in the RHEL 8.6 kernel has the latest upstream fixes. These fixes are essential for ONTAP support.

Steps

1. Verify that you are running the supported adapter driver and firmware versions:

```
# cat /sys/class/fc_host/host*/symbolic_name
QLE2742 FW:v9.06.02 DVR:v10.02.00.200-k
QLE2742 FW:v9.06.02 DVR:v10.02.00.200-k
```

2. Verify ql2xnvmeenable is set which enables the Marvell adapter to function as a NVMe/FC initiator using the following command:

```
# cat /sys/module/qla2xxx/parameters/ql2xnvmeenable
1
```

Enable 1MB I/O (Optional)

ONTAP reports an MDTS (Max Data Transfer Size) of 8 in the Identify Controller data. This means the maximum I/O request size can be up to 1MB. To issue I/O requests of size 1MB for a Broadcom NVMe/FC host, you should increase the lpfc value of the lpfc_sg_seg_cnt parameter to 256 from the default value of 64.

> These steps don't apply to Qlogic NVMe/FC hosts.

Steps

1. Set the lpfc_sg_seg_cnt parameter to 256:

```
cat /etc/modprobe.d/lpfc.conf
```

You should see an output similar to the following example:

```
options lpfc lpfc_sg_seg_cnt=256
```

2. Run the dracut -f command, and reboot the host.
3. Verify that the value for lpfc_sg_seg_cnt is 256:

```
cat /sys/module/lpfc/parameters/lpfc_sg_seg_cnt
```

Configure NVMe/TCP

NVMe/TCP does not have auto-connect functionality. Therefore, if a path goes down and is not reinstated within the default time out period of 10 minutes, NVMe/TCP cannot automatically reconnect. To prevent a time out, you should set the retry period for failover events to at least 30 minutes.

Steps

1. Verify whether the initiator port can fetch the discovery log page data across the supported NVMe/TCP LIFs:

```
# nvme discover -t tcp -w 192.168.1.8 -a 192.168.1.51
Discovery Log Number of Records 10, Generation counter 119
=====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.56e362e9bb4f11ebbaded039ea165abc:subsystem.nvme_118_tcp
_1
traddr: 192.168.2.56
sectype: none
=====Discovery Log Entry 1======
trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 1
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.56e362e9bb4f11ebbaded039ea165abc:subsystem.nvme_118_tcp
_1
traddr: 192.168.1.51
sectype: none
=====Discovery Log Entry 2======
trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 0
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.56e362e9bb4f11ebbaded039ea165abc:subsystem.nvme_118_tcp
_2
traddr: 192.168.2.56
sectype: none
...
```

2. Verify that other NVMe/TCP initiator-target LIF combos can successfully fetch discovery log page data. For example:

```
# nvme discover -t tcp -w 192.168.1.8 -a 192.168.1.51
# nvme discover -t tcp -w 192.168.1.8 -a 192.168.1.52
# nvme discover -t tcp -w 192.168.2.9 -a 192.168.2.56
# nvme discover -t tcp -w 192.168.2.9 -a 192.168.2.57
```

3. Run `nvme connect-all` command across all the supported NVMe/TCP initiator-target LIFs across the nodes. Ensure you set a longer `ctrl_loss_tmo` timer retry period (for example, 30 minutes, which can be set through `-l 1800`) during the connect-all so that it would retry for a longer period of time in the event of a path loss. For example:

```
# nvme connect-all -t tcp -w 192.168.1.8 -a 192.168.1.51 -l 1800
# nvme connect-all -t tcp -w 192.168.1.8 -a 192.168.1.52 -l 1800
# nvme connect-all -t tcp -w 192.168.2.9 -a 192.168.2.56 -l 1800
# nvme connect-all -t tcp -w 192.168.2.9 -a 192.168.2.57 -l 1800
```

Validate NVMe-oF

You can use the following procedure to validate NVMe-oF.

Steps

1. Verify that in-kernel NVMe multipath is enabled:

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

2. Verify that the appropriate NVMe-oF settings (such as, `model` set to `NetApp ONTAP Controller` and load balancing `iopolicy` set to `round-robin`) for the respective ONTAP namespaces properly reflect on the host:

```
# cat /sys/class/nvme-subsystem/nvme-subsys*/model
NetApp ONTAP Controller
NetApp ONTAP Controller

# cat /sys/class/nvme-subsystem/nvme-subsys*/iopolicy
round-robin
round-robin
```

3. Verify that the ONTAP namespaces properly reflect on the host. For example:

```
# nvme list
Node           SN                    Model                       Namespace
------------   --------------------- ----------------------------------
/dev/nvme0n1   814vWBNRwf9HAAAAAAAB   NetApp ONTAP Controller   1

Usage                Format         FW Rev
-------------------  -----------    --------
85.90 GB / 85.90 GB  4 KiB + 0 B    FFFFFFFF
```

4. Verify that the controller state of each path is live and has proper ANA status. For example:

```
# nvme list-subsys /dev/nvme1n1
nvme-subsys1 - nvme-subsys0 - NQN=nqn.1992-
08.com.netapp:sn.5f5f2c4aa73b11e9967e00a098df41bd:subsystem.nvme_141_1
\
+- nvme0 fc traddr=nn-0x203700a098dfdd91:pn-0x203800a098dfdd91
host_traddr=nn-0x200000109b1c1204:pn-0x100000109b1c1204 live
inaccessible
+- nvme1 fc traddr=nn-0x203700a098dfdd91:pn-0x203900a098dfdd91
host_traddr=nn-0x200000109b1c1204:pn-0x100000109b1c1204 live
inaccessible
+- nvme2 fc traddr=nn-0x203700a098dfdd91:pn-0x203a00a098dfdd91
host_traddr=nn-0x200000109b1c1205:pn-0x100000109b1c1205 live optimized
+- nvme3 fc traddr=nn-0x203700a098dfdd91:pn-0x203d00a098dfdd91
host_traddr=nn-0x200000109b1c1205:pn-0x100000109b1c1205 live optimized
```

5. Verify that the NetApp plug-in displays proper values for each ONTAP namespace device. For example:

```
# nvme netapp ontapdevices -o column
Device       Vserver          Namespace Path
---------    -------
--------------------------------------------------
/dev/nvme0n1 vs_fcnvme_141    /vol/fcnvme_141_vol_1_1_0/fcnvme_141_ns

NSID  UUID                                  Size
----  ------------------------------         ------
1     72b887b1-5fb6-47b8-be0b-33326e2542e2  85.90GB


# nvme netapp ontapdevices -o json
{
"ONTAPdevices" : [
    {
        "Device" : "/dev/nvme0n1",
        "Vserver" : "vs_fcnvme_141",
        "Namespace_Path" : "/vol/fcnvme_141_vol_1_1_0/fcnvme_141_ns",
        "NSID" : 1,
        "UUID" : "72b887b1-5fb6-47b8-be0b-33326e2542e2",
        "Size" : "85.90GB",
        "LBA_Data_Size" : 4096,
        "Namespace_Size" : 20971520
    }
  ]
}
```

Known issues

The NVMe-oF host configuration for RHEL 8.6 with ONTAP has the following known issues:

NetApp Bug ID	Title	Description
1479047	RHEL 8.6 NVMe-oF hosts create duplicate Persistent Discovery Controllers	On NVMe over Fabrics (NVMe-oF) hosts, you can use the "nvme discover -p" command to create Persistent Discovery Controllers (PDCs). When this command is used, only one PDC should be created per initiator-target combination. However, if you are running ONTAP 9.10.1 and Red Hat Enterprise Linux (RHEL) 8.6 with an NVMe-oF host, a duplicate PDC is created each time "nvme discover -p" is executed. This leads to unnecessary usage of resources on both the host and the target.

NVMe-oF host configuration for RHEL 8.5 with ONTAP

NVMe over Fabrics or NVMe-oF (including NVMe/FC and other transports) is supported with Red Hat Enterprise Linux (RHEL) 8.5 with ANA (Asymmetric Namespace Access). ANA is the asymmetric logical unit access (ALUA) equivalent in the NVMe-oF environment, and is currently implemented with in-kernel NVMe Multipath. During this procedure, you enable NVMe-oF with in-kernel NVMe Multipath using ANA on RHEL 8.5 and ONTAP as the target.

See the Interoperability Matrix Tool for accurate details regarding supported configurations.

Features

RHEL 8.5 includes support for NVMe/TCP (as a Technology Preview feature) in addition to NVMe/FC. The NetApp plugin in the native nvme-cli package can display ONTAP details for both NVMe/FC and NVMe/TCP namespaces.

Known limitations

- For RHEL 8.5, in-kernel NVMe multipath remains disabled by default. Therefore, you need to enable it manually.
- NVMe/TCP on RHEL 8.5 remains a Technology Preview feature due to open issues. Refer to the RHEL 8.5 Release Notes for details.
- SAN booting using the NVMe-oF protocol is currently not supported.

Enable SAN booting

You can configure your host to use SAN booting to simplify deployment and improve scalability.

Before you begin

Use the Interoperability Matrix Tool to verify that your Linux OS, host bus adapter (HBA), HBA firmware, HBA boot BIOS, and ONTAP version support SAN booting.

Steps

1. Create a SAN boot namespace and map it to the host.

   See Provision NVMe storage.

2. Enable SAN booting in the server BIOS for the ports to which the SAN boot LUN is mapped.

   For information on how to enable the HBA BIOS, see your vendor-specific documentation.

3. Verify that the configuration was successful by rebooting the host and verifying that the OS is up and running.

Enable in-kernel NVMe Multipath

You can use the following procedure to enable in-kernel NVMe multipath.

Steps

1. Install RHEL 8.5 GA on the server. After the installation is complete, verify that you are running the specified RHEL 8.5 GA kernel. See the Interoperability Matrix Tool for the current list of supported versions.

Example:

```
# uname -r
4.18.0-348.el8.x86_64
```

2. Install the `nvme-cli` package:

   Example:

```
# rpm -qa|grep nvme-cli
nvme-cli-1.14-3.el8.x86_64
```

3. Enable in-kernel NVMe multipath:

```
# grubby --args=nvme_core.multipath=Y --update-kernel /boot/vmlinuz-
4.18.0-348.el8.x86_64
```

4. On the host, check the host NQN string at `/etc/nvme/hostnqn` and verify that it matches the host NQN string for the corresponding subsystem on the ONTAP array. Example:

```
# cat /etc/nvme/hostnqn
nqn.2014-08.org.nvmexpress:uuid:9ed5b327-b9fc-4cf5-97b3-1b5d986345d1
::> vserver nvme subsystem host show -vserver vs_fcnvme_141
Vserver     Subsystem       Host NQN
----------- ---------------
----------------------------------------------------------
vs_fcnvme_14 nvme_141_1     nqn.2014-08.org.nvmexpress:uuid:9ed5b327-
b9fc-4cf5-97b3-1b5d986345d1
```

   If the host NQN strings do not match, you should use the `vserver modify` command to update the host NQN string on your corresponding ONTAP NVMe subsystem to match the host NQN string `/etc/nvme/hostnqn` on the host.

5. Reboot the host.

If you intend to run both NVMe and SCSI co-existent traffic on the same host, NetApp recommends using in-kernel NVMe multipath for ONTAP namespaces and dm-multipath for ONTAP LUNs respectively. This means that the ONTAP namespaces should be excluded from dm-multipath to prevent dm-multipath from claiming these namespace devices. You can do this by adding the enable_foreign setting to the `/etc/multipath.conf` file:

```
# cat /etc/multipath.conf
defaults {
        enable_foreign     NONE
}
```

Restart the multipathd daemon by running a `systemctl restart multipathd` command to allow the new setting to take effect.

Configure NVMe/FC

You can configure NVMe/FC for Broadcom/Emulex or Marvell/Qlogic adapters.

Broadcom/Emulex

Steps

1. Verify that you are using the supported adapter. See the Interoperability Matrix Tool for the current list of supported adapters.

```
# cat /sys/class/scsi_host/host*/modelname
LPe32002-M2
LPe32002-M2
# cat /sys/class/scsi_host/host*/modeldesc
Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
```

2. Verify that you are using the recommended Broadcom lpfc firmware and inbox driver. See the Interoperability Matrix Tool for the current list of supported adapter driver and firmware versions.

```
# cat /sys/class/scsi_host/host*/fwrev
12.8.351.47, sli-4:2:c
12.8.351.47, sli-4:2:c
# cat /sys/module/lpfc/version
0:12.8.0.10
```

3. Verify that `lpfc_enable_fc4_type` is set to 3

```
# cat /sys/module/lpfc/parameters/lpfc_enable_fc4_type
3
```

4. Verify that the initiator ports are up and running, and that you can see the target LIFs.

```
# cat /sys/class/fc_host/host*/port_name
0x100000109b1c1204
0x100000109b1c1205

# cat /sys/class/fc_host/host*/port_state
Online
Online

# cat /sys/class/scsi_host/host*/nvme_info

NVME Initiator Enabled
XRI Dist lpfc0 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc0 WWPN x100000109b1c1204 WWNN x200000109b1c1204 DID
x011d00 ONLINE
NVME RPORT WWPN x203800a098dfdd91 WWNN x203700a098dfdd91 DID x010c07
TARGET DISCSRVC ONLINE
NVME RPORT WWPN x203900a098dfdd91 WWNN x203700a098dfdd91 DID x011507
TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 0000000f78 Cmpl 0000000f78 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000002fe29bba Issue 000000002fe29bc4 OutIO
000000000000000a
abort 00001bc7 noxri 00000000 nondlp 00000000 qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 00001e15 Err 0000d906

NVME Initiator Enabled
XRI Dist lpfc1 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc1 WWPN x100000109b1c1205 WWNN x200000109b1c1205 DID
x011900 ONLINE
NVME RPORT WWPN x203d00a098dfdd91 WWNN x203700a098dfdd91 DID x010007
TARGET DISCSRVC ONLINE
NVME RPORT WWPN x203a00a098dfdd91 WWNN x203700a098dfdd91 DID x012a07
TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 0000000fa8 Cmpl 0000000fa8 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000002e14f170 Issue 000000002e14f17a OutIO
000000000000000a
abort 000016bb noxri 00000000 nondlp 00000000 qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 00001f50 Err 0000d9f8
```

Marvell/QLogic

The native inbox `qla2xxx` driver included in the RHEL 8.5 GA kernel has the latest fixes. These fixes are essential for ONTAP support.

Steps

1. Verify that you are running the supported adapter driver and firmware versions:

```
# cat /sys/class/fc_host/host*/symbolic_name
QLE2742 FW:v9.06.02 DVR:v10.02.00.106-k
QLE2742 FW:v9.06.02 DVR:v10.02.00.106-k
```

2. Verify `ql2xnvmeenable` is set which enables the Marvell adapter to function as a NVMe/FC initiator:

```
# cat /sys/module/qla2xxx/parameters/ql2xnvmeenable
1
```

Enable 1MB I/O (Optional)

ONTAP reports an MDTS (Max Data Transfer Size) of 8 in the Identify Controller data. This means the maximum I/O request size can be up to 1MB. To issue I/O requests of size 1MB for a Broadcom NVMe/FC host, you should increase the `lpfc` value of the `lpfc_sg_seg_cnt` parameter to 256 from the default value of 64.

> These steps don't apply to Qlogic NVMe/FC hosts.

Steps

1. Set the `lpfc_sg_seg_cnt` parameter to 256:

```
cat /etc/modprobe.d/lpfc.conf
```

You should see an output similar to the following example:

```
options lpfc lpfc_sg_seg_cnt=256
```

2. Run the `dracut -f` command, and reboot the host.
3. Verify that the value for `lpfc_sg_seg_cnt` is 256:

```
cat /sys/module/lpfc/parameters/lpfc_sg_seg_cnt
```

Configure NVMe/TCP

NVMe/TCP does not have auto-connect functionality. Therefore, if a path goes down and is not reinstated within the default time out period of 10 minutes, NVMe/TCP cannot automatically reconnect. To prevent a time out, you should set the retry period for failover events to at least 30 minutes.

Steps

1. Verify whether the initiator port can fetch the discovery log page data across the supported NVMe/TCP LIFs:

```
# nvme discover -t tcp -w 192.168.1.8 -a 192.168.1.51
Discovery Log Number of Records 10, Generation counter 119
=====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.56e362e9bb4f11ebbaded039ea165abc:subsystem.nvme_118_tcp
_1
traddr: 192.168.2.56
sectype: none
=====Discovery Log Entry 1======
trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 1
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.56e362e9bb4f11ebbaded039ea165abc:subsystem.nvme_118_tcp
_1
traddr: 192.168.1.51
sectype: none
=====Discovery Log Entry 2======
trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 0
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.56e362e9bb4f11ebbaded039ea165abc:subsystem.nvme_118_tcp
_2
traddr: 192.168.2.56
sectype: none
...
```

2. Verify that other NVMe/TCP initiator-target LIF combos can successfully fetch discovery log page data. For example:

```
# nvme discover -t tcp -w 192.168.1.8 -a 192.168.1.51
# nvme discover -t tcp -w 192.168.1.8 -a 192.168.1.52
# nvme discover -t tcp -w 192.168.2.9 -a 192.168.2.56
# nvme discover -t tcp -w 192.168.2.9 -a 192.168.2.57
```

3. Run the `nvme connect-all` command across all the supported NVMe/TCP initiator-target LIFs across the nodes. Ensure you set a longer `ctrl_loss_tmo` timer retry period (for example, 30 minutes, which can be set through `-l 1800`) during the connect-all so that it retries for a longer period of time in the event of a path loss. For example:

```
# nvme connect-all -t tcp -w 192.168.1.8 -a 192.168.1.51 -l 1800
# nvme connect-all -t tcp -w 192.168.1.8 -a 192.168.1.52 -l 1800
# nvme connect-all -t tcp -w 192.168.2.9 -a 192.168.2.56 -l 1800
# nvme connect-all -t tcp -w 192.168.2.9 -a 192.168.2.57 -l 1800
```

Validate NVMe-oF

You can use the following procedure to validate NVMe-oF.

Steps

1. Verify that in-kernel NVMe multipath is enabled:

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

2. Verify that the appropriate NVMe-oF settings (such as, `model` set to `NetApp ONTAP Controller` and `load balancing iopolicy` set to `round-robin`) for the respective ONTAP namespaces properly reflect on the host:

```
# cat /sys/class/nvme-subsystem/nvme-subsys*/model
NetApp ONTAP Controller
NetApp ONTAP Controller

# cat /sys/class/nvme-subsystem/nvme-subsys*/iopolicy
round-robin
round-robin
```

3. Verify that the ONTAP namespaces properly reflect on the host. For example:

```
# nvme list
Node           SN                     Model                    Namespace
------------   ---------------------- ----------------------------------
/dev/nvme0n1    814vWBNRwf9HAAAAAAB  NetApp ONTAP Controller   1

Usage                 Format         FW Rev
-------------------  -----------    --------
85.90 GB / 85.90 GB  4 KiB + 0 B    FFFFFFFF
```

4. Verify that the controller state of each path is live and has proper ANA status. For example:

```
# nvme list-subsys /dev/nvme0n1
nvme-subsys0 - NQN=nqn.1992-
08.com.netapp:sn.5f5f2c4aa73b11e9967e00a098df41bd:subsystem.nvme_141_1
\
+- nvme0 fc traddr=nn-0x203700a098dfdd91:pn-0x203800a098dfdd91
host_traddr=nn-0x200000109b1c1204:pn-0x100000109b1c1204 live
inaccessible
+- nvme1 fc traddr=nn-0x203700a098dfdd91:pn-0x203900a098dfdd91
host_traddr=nn-0x200000109b1c1204:pn-0x100000109b1c1204 live
inaccessible
+- nvme2 fc traddr=nn-0x203700a098dfdd91:pn-0x203a00a098dfdd91
host_traddr=nn-0x200000109b1c1205:pn-0x100000109b1c1205 live optimized
+- nvme3 fc traddr=nn-0x203700a098dfdd91:pn-0x203d00a098dfdd91
host_traddr=nn-0x200000109b1c1205:pn-0x100000109b1c1205 live optimized
```

5. Verify that the NetApp plug-in displays proper values for each ONTAP namespace device. For example:

```
# nvme netapp ontapdevices -o column
Device       Vserver          Namespace Path
---------    -------
--------------------------------------------------
/dev/nvme0n1 vs_fcnvme_141  vol/fcnvme_141_vol_1_1_0/fcnvme_141_ns

NSID  UUID                                   Size
----  ------------------------------         ------
1     72b887b1-5fb6-47b8-be0b-33326e2542e2  85.90GB


# nvme netapp ontapdevices -o json
{
"ONTAPdevices" : [
    {
        "Device" : "/dev/nvme0n1",
        "Vserver" : "vs_fcnvme_141",
        "Namespace_Path" : "/vol/fcnvme_141_vol_1_1_0/fcnvme_141_ns",
        "NSID" : 1,
        "UUID" : "72b887b1-5fb6-47b8-be0b-33326e2542e2",
        "Size" : "85.90GB",
        "LBA_Data_Size" : 4096,
        "Namespace_Size" : 20971520
    }
  ]
}
```

Known issues

There are no known issues.

NVMe-oF Host Configuration for RHEL 8.4 with ONTAP

NVMe over Fabrics or NVMe-oF (including NVMe/FC and other transports) is supported with Red Hat Enterprise Linux (RHEL) 8.4 with ANA (Asymmetric Namespace Access). ANA is the asymmetric logical unit access (ALUA) equivalent in the NVMe-oF environment, and is currently implemented with in-kernel NVMe Multipath. You can enable NVMe-oF with in-kernel NVMe Multipath using ANA on RHEL 8.4 and ONTAP as the target.

Features

There are no new features in this release.

Known limitations

- For RHEL 8.4, in-kernel NVMe multipath is disabled by default. Therefore, you need to enable it manually.
- NVMe/TCP on RHEL 8.4 remains a Technology Preview feature due to open issues. Refer to the RHEL 8.4 Release Notes for details.
- SAN booting using the NVMe-oF protocol is currently not supported.

Enable SAN booting

You can configure your host to use SAN booting to simplify deployment and improve scalability.

Before you begin

Use the Interoperability Matrix Tool to verify that your Linux OS, host bus adapter (HBA), HBA firmware, HBA boot BIOS, and ONTAP version support SAN booting.

Steps

1. Create a SAN boot namespace and map it to the host.

   See Provision NVMe storage.

2. Enable SAN booting in the server BIOS for the ports to which the SAN boot LUN is mapped.

   For information on how to enable the HBA BIOS, see your vendor-specific documentation.

3. Verify that the configuration was successful by rebooting the host and verifying that the OS is up and running.

Enable in-kernel NVMe multipath

You can use the following procedure to enable in-kernel NVMe multipath.

Steps

1. Install RHEL 8.4 GA on the server.
2. After the installation is complete, verify that you are running the specified RHEL 8.4 kernel. See the Interoperability Matrix Tool for the current list of supported versions.

   Example:

   ```
   # uname -r
   4.18.0-305.el8.x86_64
   ```

3. Install the `nvme-cli` package:

   Example:

   ```
   # rpm -qa|grep nvme-cli
   nvme-cli-1.12-3.el8.x86_64
   ```

4. Enable in-kernel NVMe multipath:

```
# grubby --args=nvme_core.multipath=Y --update-kernel /boot/vmlinuz-
4.18.0-305.el8.x86_64
```

5. On the host, check the host NQN string at `/etc/nvme/hostnqn` and verify that it matches the host NQN string for the corresponding subsystem on the ONTAP array. Example:

```
# cat /etc/nvme/hostnqn
nqn.2014-08.org.nvmexpress:uuid:9ed5b327-b9fc-4cf5-97b3-1b5d986345d1
::> vserver nvme subsystem host show -vserver vs_fcnvme_141
Vserver     Subsystem       Host NQN
----------- ---------------
----------------------------------------------------------
vs_fcnvme_14 nvme_141_1     nqn.2014-08.org.nvmexpress:uuid:9ed5b327-
b9fc-4cf5-97b3-1b5d986345d1
```

> If the host NQN strings do not match, you should use the `vserver modify` command to update the host NQN string on your corresponding ONTAP NVMe subsystem to match the host NQN string `/etc/nvme/hostnqn` on the host.

6. Reboot the host.

> If you intend to run both NVMe and SCSI co-existent traffic on the same host, it is recommended to use in-kernel NVMe multipath for ONTAP namespaces and dm-multipath for ONTAP LUNs respectively. This means that the ONTAP namespaces should be excluded from dm-multipath to prevent dm-multipath from claiming these namespace devices. This can be done by adding the enable_foreign setting to the `/etc/multipath.conf` file:
>
> ```
> # cat /etc/multipath.conf
> defaults {
>         enable_foreign     NONE
> }
> ```
>
> Restart the multipathd daemon by running a `systemctl restart multipathd` command to allow the new setting to take effect.

Configure NVMe/FC

You can configure NVMe/FC for Broadcom/Emulex or Marvell/Qlogic adapters.

Broadcom/Emulex

Steps

1. Verify that you are using the supported adapter. See the Interoperability Matrix Tool for the current list of supported adapters.

```
# cat /sys/class/scsi_host/host*/modelname
LPe32002-M2
LPe32002-M2
# cat /sys/class/scsi_host/host*/modeldesc
Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
```

2. Verify that you are using the recommended Broadcom lpfc firmware and inbox driver. See the Interoperability Matrix Tool for the current list of supported adapter driver and firmware versions.

```
# cat /sys/class/scsi_host/host*/fwrev
12.8.340.8, sli-4:2:c
12.8.340.8, sli-4:2:c
# cat /sys/module/lpfc/version
0:12.8.0.5
```

3. Verify that `lpfc_enable_fc4_type` is set to 3.

```
# cat /sys/module/lpfc/parameters/lpfc_enable_fc4_type
3
```

4. Verify that the initiator ports are up and running, and you are able to see the target LIFs.

```
# cat /sys/class/fc_host/host*/port_name
0x100000109b1c1204
0x100000109b1c1205
# cat /sys/class/fc_host/host*/port_state
Online
Online
# cat /sys/class/scsi_host/host*/nvme_info
NVME Initiator Enabled
XRI Dist lpfc0 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc0 WWPN x100000109b1c1204 WWNN x200000109b1c1204 DID
x011d00 ONLINE
NVME RPORT WWPN x203800a098dfdd91 WWNN x203700a098dfdd91 DID x010c07
TARGET DISCSRVC ONLINE
NVME RPORT WWPN x203900a098dfdd91 WWNN x203700a098dfdd91 DID x011507
TARGET DISCSRVC ONLINE
NVME Statistics
LS: Xmt 0000000f78 Cmpl 0000000f78 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000002fe29bba Issue 000000002fe29bc4 OutIO
000000000000000a
abort 00001bc7 noxri 00000000 nondlp 00000000 qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 00001e15 Err 0000d906
NVME Initiator Enabled
XRI Dist lpfc1 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc1 WWPN x100000109b1c1205 WWNN x200000109b1c1205 DID
x011900 ONLINE
NVME RPORT WWPN x203d00a098dfdd91 WWNN x203700a098dfdd91 DID x010007
TARGET DISCSRVC ONLINE
NVME RPORT WWPN x203a00a098dfdd91 WWNN x203700a098dfdd91 DID x012a07
TARGET DISCSRVC ONLINE
NVME Statistics
LS: Xmt 0000000fa8 Cmpl 0000000fa8 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000002e14f170 Issue 000000002e14f17a OutIO
000000000000000a
abort 000016bb noxri 00000000 nondlp 00000000 qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 00001f50 Err 0000d9f8
```

Marvell/QLogic FC adapter for NVMe/FC

The native inbox `qla2xxx` driver included in the RHEL 8.4 GA kernel has the latest fixes. These fixes are essential for ONTAP support.

Steps

1. Verify that you are running the supported adapter driver and firmware versions using the following command:

```
# cat /sys/class/fc_host/host*/symbolic_name
QLE2742 FW:v9.06.02 DVR:v10.02.00.104-k
QLE2742 FW:v9.06.02 DVR:v10.02.00.104-k
```

2. Verify `ql2xnvmeenable` is set which enables the Marvell adapter to function as a NVMe/FC initiator using the following command:

```
# cat /sys/module/qla2xxx/parameters/ql2xnvmeenable
1
```

Enable 1MB I/O (Optional)

ONTAP reports an MDTS (Max Data Transfer Size) of 8 in the Identify Controller data. This means the maximum I/O request size can be up to 1MB. To issue I/O requests of size 1MB for a Broadcom NVMe/FC host, you should increase the `lpfc` value of the `lpfc_sg_seg_cnt` parameter to 256 from the default value of 64.

> These steps don't apply to Qlogic NVMe/FC hosts.

Steps

1. Set the `lpfc_sg_seg_cnt` parameter to 256:

```
cat /etc/modprobe.d/lpfc.conf
```

You should see an output similar to the following example:

```
options lpfc lpfc_sg_seg_cnt=256
```

2. Run the `dracut -f` command, and reboot the host.
3. Verify that the value for `lpfc_sg_seg_cnt` is 256:

```
cat /sys/module/lpfc/parameters/lpfc_sg_seg_cnt
```

Configure NVMe/TCP

NVMe/TCP does not have auto-connect functionality. Therefore, if a path goes down and is not reinstated within the default time out period of 10 minutes, NVMe/TCP cannot automatically reconnect. To prevent a time out, you should set the retry period for failover events to at least 30 minutes.

Steps

1. Verify whether the initiator port can fetch the discovery log page data across the supported NVMe/TCP LIFs:

```
# nvme discover -t tcp -w 192.168.1.8 -a 192.168.1.51
Discovery Log Number of Records 10, Generation counter 119
=====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.56e362e9bb4f11ebbaded039ea165abc:subsystem.nvme_118_tcp
_1
traddr: 192.168.2.56
sectype: none
=====Discovery Log Entry 1======
trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 1
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.56e362e9bb4f11ebbaded039ea165abc:subsystem.nvme_118_tcp
_1
traddr: 192.168.1.51
sectype: none
=====Discovery Log Entry 2======
trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 0
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.56e362e9bb4f11ebbaded039ea165abc:subsystem.nvme_118_tcp
_2
traddr: 192.168.2.56
sectype: none
...
```

2. Verify that other NVMe/TCP initiator-target LIF combos are able to successfully fetch discovery log page data. For example,

```
# nvme discover -t tcp -w 192.168.1.8 -a 192.168.1.52
# nvme discover -t tcp -w 192.168.2.9 -a 192.168.2.56
# nvme discover -t tcp -w 192.168.2.9 -a 192.168.2.57
```

3. Run `nvme connect-all` command across all the supported NVMe/TCP initiator-target LIFs across the nodes. Ensure you set a longer `ctrl_loss_tmo` timer retry period (for example, 30 minutes, which can be set through `-l 1800`) during the connect-all so that it would retry for a longer period of time in the event of a path loss. For example,

```
# nvme connect-all -t tcp -w 192.168.1.8 -a 192.168.1.51 -l 1800
# nvme connect-all -t tcp -w 192.168.1.8 -a 192.168.1.52 -l 1800
# nvme connect-all -t tcp -w 192.168.2.9 -a 192.168.2.56 -l 1800
# nvme connect-all -t tcp -w 192.168.2.9 -a 192.168.2.57 -l 1800
```

Validate NVMe-oF

You can use the following procedure to validate NVMe-oF.

Steps

1. Verify that in-kernel NVMe multipath is enabled:

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

2. Verify that the appropriate NVMe-oF settings (such as, `model` set to `NetApp ONTAP Controller` and load balancing `iopolicy` set to `round-robin`) for the respective ONTAP namespaces properly reflect on the host:

```
# cat /sys/class/nvme-subsystem/nvme-subsys*/model
NetApp ONTAP Controller
NetApp ONTAP Controller

# cat /sys/class/nvme-subsystem/nvme-subsys*/iopolicy
round-robin
round-robin
```

3. Verify that the ONTAP namespaces properly reflect on the host. For example,

   Example (a):

```
# nvme list
Node           SN                    Model                          Namespace
------------   --------------------- ---------------------------------
/dev/nvme0n1   81CZ5BQuUNfGAAAAAAAB  NetApp ONTAP Controller   1

Usage                Format         FW Rev
-------------------  -----------    --------
85.90 GB / 85.90 GB  4 KiB + 0 B    FFFFFFFF
```

Example (b):

```
# nvme list
Node           SN                    Model                          Namespace
------------   --------------------- ---------------------------------
/dev/nvme0n1   81CYrBQuTHQFAAAAAAAC  NetApp ONTAP Controller   1

Usage                Format         FW Rev
-------------------  -----------    --------
85.90 GB / 85.90 GB  4 KiB + 0 B    FFFFFFFF
```

4. Verify that the controller state of each path is live and has proper ANA status. For example,

Example (a):

```
# nvme list-subsys /dev/nvme1n1
nvme-subsys1 - NQN=nqn.1992-
08.com.netapp:sn.04ba0732530911ea8e8300a098dfdd91:subsystem.nvme_145_1
\
+- nvme2 fc traddr=nn-0x208100a098dfdd91:pn-0x208200a098dfdd91
host_traddr=nn-0x200000109b579d5f:pn-0x100000109b579d5f live non-
optimized
+- nvme3 fc traddr=nn-0x208100a098dfdd91:pn-0x208500a098dfdd91
host_traddr=nn-0x200000109b579d5e:pn-0x100000109b579d5e live non-
optimized
+- nvme4 fc traddr=nn-0x208100a098dfdd91:pn-0x208400a098dfdd91
host_traddr=nn-0x200000109b579d5e:pn-0x100000109b579d5e live optimized
+- nvme6 fc traddr=nn-0x208100a098dfdd91:pn-0x208300a098dfdd91
host_traddr=nn-0x200000109b579d5f:pn-0x100000109b579d5f live optimized
```

Example (b):

```
#nvme list-subsys /dev/nvme0n1
nvme-subsys0 - NQN=nqn.1992-
08.com.netapp:sn.37ba7d9cbfba11eba35dd039ea165514:subsystem.nvme_114_tcp
_1
\
+- nvme0 tcp traddr=192.168.2.36 trsvcid=4420 host_traddr=192.168.1.4
live optimized
+- nvme1 tcp traddr=192.168.1.31 trsvcid=4420 host_traddr=192.168.1.4
live optimized
+- nvme10 tcp traddr=192.168.2.37 trsvcid=4420 host_traddr=192.168.1.4
live non-optimized
+- nvme11 tcp traddr=192.168.1.32 trsvcid=4420 host_traddr=192.168.1.4
live non-optimized
+- nvme20 tcp traddr=192.168.2.36 trsvcid=4420 host_traddr=192.168.2.5
live optimized
+- nvme21 tcp traddr=192.168.1.31 trsvcid=4420 host_traddr=192.168.2.5
live optimized
+- nvme30 tcp traddr=192.168.2.37 trsvcid=4420 host_traddr=192.168.2.5
live non-optimized
+- nvme31 tcp traddr=192.168.1.32 trsvcid=4420 host_traddr=192.168.2.5
live non-optimized
```

5. Verify that the NetApp plug-in displays proper values for each ONTAP namespace device. For example,

   Example (a):

```
# nvme netapp ontapdevices -o column
Device       Vserver          Namespace Path
---------    -------
--------------------------------------------------
/dev/nvme1n1 vserver_fcnvme_145 /vol/fcnvme_145_vol_1_0_0/fcnvme_145_ns

NSID  UUID                                   Size
-------  ------------------------------     --------------------
1      23766b68-e261-444e-b378-2e84dbe0e5e1  85.90GB


# nvme netapp ontapdevices -o json
{
"ONTAPdevices" : [
     {
       "Device" : "/dev/nvme1n1",
       "Vserver" : "vserver_fcnvme_145",
       "Namespace_Path" : "/vol/fcnvme_145_vol_1_0_0/fcnvme_145_ns",
       "NSID" : 1,
       "UUID" : "23766b68-e261-444e-b378-2e84dbe0e5e1",
       "Size" : "85.90GB",
       "LBA_Data_Size" : 4096,
       "Namespace_Size" : 20971520
     }
  ]
}
```

Example (b):

```
# nvme netapp ontapdevices -o column
Device       Vserver          Namespace Path
---------    -------
--------------------------------------------------
/dev/nvme0n1 vs_tcp_114       /vol/tcpnvme_114_1_0_1/tcpnvme_114_ns

NSID  UUID                                  Size
-------  ------------------------------         ---------------
1      a6aee036-e12f-4b07-8e79-4d38a9165686  85.90GB


# nvme netapp ontapdevices -o json
{
     "ONTAPdevices" : [
     {
          "Device" : "/dev/nvme0n1",
           "Vserver" : "vs_tcp_114",
          "Namespace_Path" : "/vol/tcpnvme_114_1_0_1/tcpnvme_114_ns",
          "NSID" : 1,
          "UUID" : "a6aee036-e12f-4b07-8e79-4d38a9165686",
          "Size" : "85.90GB",
          "LBA_Data_Size" : 4096,
          "Namespace_Size" : 20971520
       }
  ]

}
```

Known issues

There are no known issues.

NVMe/FC host configuration for RHEL 8.3 with ONTAP

Beginning with ONTAP 9.6, NVMe/FC is supported for Red Hat Enterprise Linux (RHEL) 8.3. The RHEL 8.3 host runs both NVMe and SCSI traffic through the same FC initiator adapter ports. See the Hardware Universe for a list of supported FC adapters and controllers.

See the Interoperability Matrix Tool for the current list of supported configurations.

Features

There are no new features in this release.

Known limitations

- For RHEL 8.3, in-kernel NVMe multipath is disabled by default. You can enable it manually.
- SAN booting using the NVMe-oF protocol is currently not supported.

Enable SAN booting

You can configure your host to use SAN booting to simplify deployment and improve scalability.

Before you begin

Use the Interoperability Matrix Tool to verify that your Linux OS, host bus adapter (HBA), HBA firmware, HBA boot BIOS, and ONTAP version support SAN booting.

Steps

1. Create a SAN boot namespace and map it to the host.

   See Provision NVMe storage.

2. Enable SAN booting in the server BIOS for the ports to which the SAN boot LUN is mapped.

   For information on how to enable the HBA BIOS, see your vendor-specific documentation.

3. Verify that the configuration was successful by rebooting the host and verifying that the OS is up and running.

Enable NVMe/FC on RHEL 8.3

You can use the following procedure to enable NVMe/FC.

Steps

1. Install Red Hat Enterprise Linux 8.3 GA on the server.
2. If you are upgrading from RHEL 8.2 to RHEL 8.3 using the `yum update/upgrade` command, your `/etc/nvme/host*` files might be lost. To avoid file loss, use the following procedure:

Show example output

a. Backup your `/etc/nvme/host*` files.

b. If you have a manually edited `udev` rule, remove it:

```
/lib/udev/rules.d/71-nvme-iopolicy-netapp-ONTAP.rules
```

c. Perform the upgrade.

d. After the upgrade is complete, run the following command:

```
yum remove nvme-cli
```

e. Restore the host files at `/etc/nvme/`.

```
yum install nvmecli
```

f. Copy the original `/etc/nvme/host*` contents from the backup to the actual host files at `/etc/nvme/`.

3. After the installation is complete, verify that you're running the specified RHEL kernel:

```
# uname -r
4.18.0-240.el8.x86_64
```

See the Interoperability Matrix Tool for the current list of supported versions.

4. Install the nvme-cli package:

```
# rpm -qa|grep nvme-cli
nvme-cli-1.12-2.el8.x86_64
```

5. Enable in-kernel NVMe multipath.

```
# grubby --args=nvme_core.multipath=Y --update-kernel /boot/vmlinuz-
4.18.0-240.el8.x86_64
```

6. On the RHEL 8.3 host, check the host NQN string at `/etc/nvme/hostnqn` verify that it matches the host NQN string for the corresponding subsystem on the ONTAP array:

```
# cat /etc/nvme/hostnqn
```

Example output:

```
# cat /etc/nvme/hostnqn
nqn.2014-08.org.nvmexpress:uuid:9ed5b327-b9fc-4cf5-97b3-1b5d986345d1
```

7. Verify that the `hostnqn` string matches the hostnqn string for the corresponding subsystem on the ONTAP array:

```
vserver nvme subsystem host show -vserver vs_fcnvme_141
```

Example output

```
::> vserver nvme subsystem host show -vserver vs_fcnvme_141
Vserver          Subsystem        Host            NQN
-----------      --------------- ----------- ---------------
vs_fcnvme_141     nvme_141_1                  nqn.2014-
08.org.nvmexpress:uuid:9ed5b327-b9fc-4cf5-97b3-1b5d986345d1
```

If the host NQN strings do not match, use the `vserver modify` command to update the host NQN string on the corresponding ONTAP array subsystem to match to host NQN string from `/etc/nvme/hostnqn` on the host.

8. Reboot the host.
9. Optionally, update the `enable_foreign` setting.

If you intend to run both NVMe and SCSI traffic on the same RHEL 8.3 host, NetApp recommends that you use in-kernel NVMe multipath for ONTAP namespaces and dm-multipath for ONTAP LUNs, respectively. You should also blacklist the ONTAP namespaces in dm-multipath to prevent dm-multipath from claiming these namespace devices. You can do this by adding the `enable_foreign` setting to the /etc/multipath.conf, as shown below:

```
# cat /etc/multipath.conf
defaults {
   enable_foreign NONE
}
```

Restart the multipathd daemon by running a `systemctl restart multipathd`.

Validate NVMe/FC

You can use the following procedure to validate NVMe/FC.

Steps

1. Verify the following NVMe/FC settings:

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

```
# cat /sys/class/nvme-subsystem/nvme-subsys*/model
NetApp ONTAP Controller
NetApp ONTAP Controller
```

```
# cat /sys/class/nvme-subsystem/nvme-subsys*/iopolicy
round-robin
round-robin
```

2. Verify that the namespaces are created and properly discovered on the host.

```
/dev/nvme0n1     814vWBNRwf9HAAAAAAAB  NetApp ONTAP Controller
1                  85.90 GB / 85.90 GB     4 KiB + 0 B   FFFFFFFF
/dev/nvme0n2     814vWBNRwf9HAAAAAAAB  NetApp ONTAP Controller
2                  85.90 GB / 85.90 GB     4 KiB + 0 B   FFFFFFFF
/dev/nvme0n3     814vWBNRwf9HAAAAAAAB  NetApp ONTAP Controller
3                  85.90 GB / 85.90 GB     4 KiB + 0 B   FFFFFFFF
```

3. Verify the status of the ANA paths.

```
# nvme list-subsys /dev/nvme0n1
nvme-subsys0 - NQN=nqn.1992-
08.com.netapp:sn.5f5f2c4aa73b11e9967e00a098df41bd:subsystem.nvme_141_1
\
+- nvme0 fc traddr=nn-0x203700a098dfdd91:pn-0x203800a098dfdd91
host_traddr=nn-0x200000109b1c1204:pn-0x100000109b1c1204 live
inaccessible
+- nvme1 fc traddr=nn-0x203700a098dfdd91:pn-0x203900a098dfdd91
host_traddr=nn-0x200000109b1c1204:pn-0x100000109b1c1204 live
inaccessible
+- nvme2 fc traddr=nn-0x203700a098dfdd91:pn-0x203a00a098dfdd91
host_traddr=nn-0x200000109b1c1205:pn-0x100000109b1c1205 live optimized
+- nvme3 fc traddr=nn-0x203700a098dfdd91:pn-0x203d00a098dfdd91
host_traddr=nn-0x200000109b1c1205:pn-0x100000109b1c1205 live optimized
```

4. Verify the NetApp plug-in for ONTAP devices:

Column

```
# nvme netapp ontapdevices -o column
```

Example output

```
Device                  Vserver             Namespace Path
NSID                         UUID                         Size
--------------- ---------------
---------------------------------------------  --------
--------------------------------------  ---------
/dev/nvme0n1      vs_fcnvme_141
/vol/fcnvme_141_vol_1_1_0/fcnvme_141_ns        1      72b887b1-5fb6-
47b8-be0b-33326e2542e2    85.90GB
/dev/nvme0n2      vs_fcnvme_141
/vol/fcnvme_141_vol_1_0_0/fcnvme_141_ns        2      04bf9f6e-9031-
40ea-99c7-a1a61b2d7d08    85.90GB
/dev/nvme0n3      vs_fcnvme_141
/vol/fcnvme_141_vol_1_1_1/fcnvme_141_ns        3      264823b1-8e03-
4155-80dd-e904237014a4    85.90GB
```

JSON

```
# nvme netapp ontapdevices -o json
```

Example output

```
{
"ONTAPdevices" : [
    {
        "Device" : "/dev/nvme0n1",
        "Vserver" : "vs_fcnvme_141",
        "Namespace_Path" :
"/vol/fcnvme_141_vol_1_1_0/fcnvme_141_ns",
        "NSID" : 1,
        "UUID" : "72b887b1-5fb6-47b8-be0b-33326e2542e2",
        "Size" : "85.90GB",
        "LBA_Data_Size" : 4096,
        "Namespace_Size" : 20971520
    },
    {
        "Device" : "/dev/nvme0n2",
        "Vserver" : "vs_fcnvme_141",
        "Namespace_Path" :
"/vol/fcnvme_141_vol_1_0_0/fcnvme_141_ns",
        "NSID" : 2,
        "UUID" : "04bf9f6e-9031-40ea-99c7-a1a61b2d7d08",
        "Size" : "85.90GB",
        "LBA_Data_Size" : 4096,
        "Namespace_Size" : 20971520
      },
      {
         "Device" : "/dev/nvme0n3",
         "Vserver" : "vs_fcnvme_141",
         "Namespace_Path" :
"/vol/fcnvme_141_vol_1_1_1/fcnvme_141_ns",
         "NSID" : 3,
         "UUID" : "264823b1-8e03-4155-80dd-e904237014a4",
         "Size" : "85.90GB",
         "LBA_Data_Size" : 4096,
         "Namespace_Size" : 20971520
       },
  ]
```

Configure the Broadcom FC adapter for NVMe/FC

You can use the following procedure to configure a Broadcom FC adapter.

For the current list of supported adapters, see the Interoperability Matrix Tool.

Steps

1. Verify that you are using the supported adapter.

```
# cat /sys/class/scsi_host/host*/modelname
LPe32002-M2
LPe32002-M2
```

```
# cat /sys/class/scsi_host/host*/modeldesc
Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
```

2. Verify that `lpfc_enable_fc4_type` is set to "**3**".

```
# cat /sys/module/lpfc/parameters/lpfc_enable_fc4_type
3
```

3. Verify that the initiator ports are up and running and can see the target LIFs.

```
# cat /sys/class/fc_host/host*/port_name
0x100000109b1c1204
0x100000109b1c1205
```

```
# cat /sys/class/fc_host/host*/port_state
Online
Online
```

```
# cat /sys/class/scsi_host/host*/nvme_info
NVME Initiator Enabled
XRI Dist lpfc0 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc0 WWPN x100000109b1c1204 WWNN x200000109b1c1204 DID
x011d00 ONLINE
NVME RPORT WWPN x203800a098dfdd91 WWNN x203700a098dfdd91 DID x010c07
TARGET DISCSRVC ONLINE
NVME RPORT WWPN x203900a098dfdd91 WWNN x203700a098dfdd91 DID x011507
TARGET DISCSRVC ONLINE
NVME Statistics
LS: Xmt 0000000f78 Cmpl 0000000f78 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000002fe29bba Issue 000000002fe29bc4 OutIO
000000000000000a
abort 00001bc7 noxri 00000000 nondlp 00000000 qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 00001e15 Err 0000d906
NVME Initiator Enabled
XRI Dist lpfc1 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc1 WWPN x100000109b1c1205 WWNN x200000109b1c1205 DID
x011900 ONLINE
NVME RPORT WWPN x203d00a098dfdd91 WWNN x203700a098dfdd91 DID x010007
TARGET DISCSRVC ONLINE
NVME RPORT WWPN x203a00a098dfdd91 WWNN x203700a098dfdd91 DID x012a07
TARGET DISCSRVC ONLINE
NVME Statistics
LS: Xmt 0000000fa8 Cmpl 0000000fa8 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000002e14f170 Issue 000000002e14f17a OutIO
000000000000000a
abort 000016bb noxri 00000000 nondlp 00000000 qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 00001f50 Err 0000d9f8
```

4. Enable 1MB I/O size *(optional)*.

   The `lpfc_sg_seg_cnt` parameter needs to be set to 256 for the lpfc driver to issue I/O requests up to 1MB in size.

```
# cat /etc/modprobe.d/lpfc.conf
options lpfc lpfc_sg_seg_cnt=256
```

5. Run the `dracut -f` command and then reboot the host.
6. After the host boots up, verify that lpfc_sg_seg_cnt is set to 256.

```
# cat /sys/module/lpfc/parameters/lpfc_sg_seg_cnt
256
```

7. Verify that you are using the recommended Broadcom lpfc firmware as well as the inbox driver:

```
# cat /sys/class/scsi_host/host*/fwrev
12.8.340.8, sli-4:2:c
12.8.340.8, sli-4:2:c
```

```
# cat /sys/module/lpfc/version
0:12.8.0.1
```

NVMe/FC host configuration for RHEL 8.2 with ONTAP

Beginning with ONTAP 9.6, NVMe/FC is supported for Red Hat Enterprise Linux (RHEL) 8.2. The RHEL 8.2 host runs both NVMe and SCSI traffic through the same fibre channel (FC) initiator adapter ports. See the Hardware Universe for a list of supported FC adapters and controllers.

See the Interoperability Matrix Tool for the current list of supported configurations.

Features

- Beginning with RHEL 8.2, `nvme-fc auto-connect` scripts are included in the native `nvme-cli` package. You can use these native auto-connect scripts instead of having to install the external vendor provided outbox auto-connect scripts.
- Beginning with RHEL 8.2, a native `udev` rule is already provided as part of the `nvme-cli` package which enables round-robin load balancing for NVMe multipath. You need not manually create this rule any more (as was done in RHEL 8.1).
- Beginning with RHEL 8.2, both NVMe and SCSI traffic can be run on the same host. In fact, this is the expected deployed host configuration. Therefore, for SCSI, you can configure `dm-multipath` as usual for SCSI LUNs resulting in `mpath` devices, whereas NVMe multipath can be used to configure NVMe-oF multipath devices on the host.
- Beginning with RHEL 8.2, the NetApp plug-in in the native `nvme-cli` package is capable of displaying ONTAP details for ONTAP namespaces.

Known limitations

- For RHEL 8.2, in-kernel NVMe multipath is disabled by default. Therefore, you need to enable it manually.
- SAN booting using the NVMe-oF protocol is currently not supported.

Enable SAN booting

You can configure your host to use SAN booting to simplify deployment and improve scalability.

Before you begin

Use the Interoperability Matrix Tool to verify that your Linux OS, host bus adapter (HBA), HBA firmware, HBA boot BIOS, and ONTAP version support SAN booting.

Steps

1. Create a SAN boot namespace and map it to the host.

   See Provision NVMe storage.

2. Enable SAN booting in the server BIOS for the ports to which the SAN boot LUN is mapped.

   For information on how to enable the HBA BIOS, see your vendor-specific documentation.

3. Verify that the configuration was successful by rebooting the host and verifying that the OS is up and running.

Enable NVMe/FC

You can use the following procedure to enable NVMe/FC.

Steps

1. Install Red Hat Enterprise Linux 8.2 GA on the server.
2. If you are upgrading from RHEL 8.1 to RHEL 8.2 using `yum update/upgrade`, your `/etc/nvme/host*` files might be lost. To avoid the file loss, do the following:
   a. Backup your `/etc/nvme/host*` files.
   b. If you have a manually edited `udev` rule, remove it:

      ```
      /lib/udev/rules.d/71-nvme-iopolicy-netapp-ONTAP.rules
      ```

   c. Perform the upgrade.
   d. After the upgrade is complete, run the following command:

      ```
      yum remove nvme-cli
      ```

   e. Restore the host files at `/etc/nvme/`.

      ```
      yum install nvmecli
      ```

   f. Copy the original `/etc/nvme/host*` contents from the backup to the actual host files at `/etc/nvme/`.
3. After the installation is complete, verify that you are running the specified Red Hat Enterprise Linux kernel.

```
# uname -r
4.18.0-193.el8.x86_64
```

See the Interoperability Matrix Tool for the current list of supported versions.

4. Install the nvme-cli package.

```
# rpm -qa|grep nvme-cli
nvme-cli-1.9.5.el8.x86_64
```

5. Enable in-kernel NVMe multipath.

```
# grubby –args=nvme_core.multipath=Y –update-kernel /boot/vmlinuz-
4.18.0-193.el8.x86_64
```

6. On the RHEL 8.2 host, check the host NQN string at `/etc/nvme/hostnqn` and verify that it matches the host NQN string for the corresponding subsystem on the ONTAP array.

```
# cat /etc/nvme/hostnqn
nqn.2014-08.org.nvmexpress:uuid:9ed5b327-b9fc-4cf5-97b3-1b5d986345d1



::> vserver nvme subsystem host show -vserver vs_fcnvme_141
Vserver      Subsystem        Host           NQN
----------- --------------- ----------- ---------------
  vs_fcnvme_141
    nvme_141_1
        nqn.2014-08.org.nvmexpress:uuid:9ed5b327-b9fc-4cf5-97b3-
1b5d986345d1
```

If the host NQN strings do not match, use the `vserver modify` command to update the host NQN string on the corresponding ONTAP array subsystem to match to host NQN string from `/etc/nvme/hostnqn` on the host.

7. Reboot the host.

8. Update the `enable_foreign` setting *(optional)*.

If you intend to run both NVMe and SCSI traffic on the same RHEL 8.2 host, NetApp recommends using in-kernel NVMe multipath for ONTAP namespaces and dm-multipath for ONTAP LUNs respectively. You should also blacklist the ONTAP namespaces in dm-multipath to prevent dm-multipath from claiming these namespace devices. You can do this by adding the `enable_foreign` setting to the `/etc/multipath.conf`, as shown below.

```
# cat /etc/multipath.conf
defaults {
   enable_foreign NONE
}
```

9. Restart the multipathd daemon by running a `systemctl restart multipathd`.

Configure the Broadcom FC adapter for NVMe/FC

You can use the following procedure to configure a Broadcom FC adapter.

For the current list of supported adapters, see the Interoperability Matrix Tool.

Steps

1. Verify that you are using the supported adapter.

```
# cat /sys/class/scsi_host/host*/modelname
LPe32002-M2
LPe32002-M2
```

```
# cat /sys/class/scsi_host/host*/modeldesc
Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
```

2. Verify that `lpfc_enable_fc4_type` is set to "**3**".

```
# cat /sys/module/lpfc/parameters/lpfc_enable_fc4_type
3
```

3. Verify that the initiator ports are up and running and can see the target LIFs.

```
# cat /sys/class/fc_host/host*/port_name
0x100000109b1c1204
0x100000109b1c1205
```

```
# cat /sys/class/fc_host/host*/port_state
Online
Online
```

```
# cat /sys/class/scsi_host/host*/nvme_info
NVME Initiator Enabled
XRI Dist lpfc0 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc0 WWPN x100000109b1c1204 WWNN x200000109b1c1204 DID
x011d00 ONLINE
NVME RPORT WWPN x203800a098dfdd91 WWNN x203700a098dfdd91 DID x010c07
TARGET DISCSRVC ONLINE
NVME RPORT WWPN x203900a098dfdd91 WWNN x203700a098dfdd91 DID x011507
TARGET DISCSRVC ONLINE
NVME Statistics
LS: Xmt 0000000f78 Cmpl 0000000f78 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000002fe29bba Issue 000000002fe29bc4 OutIO
000000000000000a
abort 00001bc7 noxri 00000000 nondlp 00000000 qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 00001e15 Err 0000d906
NVME Initiator Enabled
XRI Dist lpfc1 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc1 WWPN x100000109b1c1205 WWNN x200000109b1c1205 DID
x011900 ONLINE
NVME RPORT WWPN x203d00a098dfdd91 WWNN x203700a098dfdd91 DID x010007
TARGET DISCSRVC ONLINE
NVME RPORT WWPN x203a00a098dfdd91 WWNN x203700a098dfdd91 DID x012a07
TARGET DISCSRVC ONLINE
NVME Statistics
LS: Xmt 0000000fa8 Cmpl 0000000fa8 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000002e14f170 Issue 000000002e14f17a OutIO
000000000000000a
abort 000016bb noxri 00000000 nondlp 00000000 qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 00001f50 Err 0000d9f8
```

4. Enable 1MB I/O size *(optional)*.

   The `lpfc_sg_seg_cnt` parameter needs to be set to 256 for the lpfc driver to issue I/O requests up to 1MB in size.

   ```
   # cat /etc/modprobe.d/lpfc.conf
   options lpfc lpfc_sg_seg_cnt=256
   ```

5. Run the `dracut -f` command and then reboot the host.
6. After the host boots up, verify that lpfc_sg_seg_cnt is set to 256.

```
# cat /sys/module/lpfc/parameters/lpfc_sg_seg_cnt
256
```

7. Verify that you are using the recommended Broadcom lpfc firmware as well as the inbox driver.

```
# cat /sys/class/scsi_host/host*/fwrev
12.6.182.8, sli-4:2:c
12.6.182.8, sli-4:2:c
```

```
# cat /sys/module/lpfc/version
0:12.6.0.2
```

8. Verify that `lpfc_enable_fc4_type` is set to "**3**".

```
# cat /sys/module/lpfc/parameters/lpfc_enable_fc4_type
3
```

9. Verify that the initiator ports are up and running and can see the target LIFs.

```
# cat /sys/class/fc_host/host*/port_name
0x100000109b1c1204
0x100000109b1c1205
```

```
# cat /sys/class/fc_host/host*/port_state
Online
Online
```

```
# cat /sys/class/scsi_host/host*/nvme_info
NVME Initiator Enabled
XRI Dist lpfc0 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc0 WWPN x100000109b1c1204 WWNN x200000109b1c1204 DID
x011d00 ONLINE
NVME RPORT WWPN x203800a098dfdd91 WWNN x203700a098dfdd91 DID x010c07
TARGET DISCSRVC ONLINE
NVME RPORT WWPN x203900a098dfdd91 WWNN x203700a098dfdd91 DID x011507
TARGET DISCSRVC ONLINE
NVME Statistics
LS: Xmt 0000000f78 Cmpl 0000000f78 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000002fe29bba Issue 000000002fe29bc4 OutIO
000000000000000a
abort 00001bc7 noxri 00000000 nondlp 00000000 qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 00001e15 Err 0000d906
NVME Initiator Enabled
XRI Dist lpfc1 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc1 WWPN x100000109b1c1205 WWNN x200000109b1c1205 DID
x011900 ONLINE
NVME RPORT WWPN x203d00a098dfdd91 WWNN x203700a098dfdd91 DID x010007
TARGET DISCSRVC ONLINE
NVME RPORT WWPN x203a00a098dfdd91 WWNN x203700a098dfdd91 DID x012a07
TARGET DISCSRVC ONLINE
NVME Statistics
LS: Xmt 0000000fa8 Cmpl 0000000fa8 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000002e14f170 Issue 000000002e14f17a OutIO
000000000000000a
abort 000016bb noxri 00000000 nondlp 00000000 qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 00001f50 Err 0000d9f8
```

10. Enable 1MB I/O size *(optional)*.

    The `lpfc_sg_seg_cnt` parameter needs to be set to 256 for the lpfc driver to issue I/O requests up to 1MB in size.

    ```
    # cat /etc/modprobe.d/lpfc.conf
    options lpfc lpfc_sg_seg_cnt=256
    ```

11. Run the `dracut -f` command and then reboot the host.
12. After the host boots up, verify that lpfc_sg_seg_cnt is set to 256.

```
# cat /sys/module/lpfc/parameters/lpfc_sg_seg_cnt
256
```

Validate NVMe/FC

You can use the following procedure to validate NVMe/FC.

Steps

1. Verify the following NVMe/FC settings.

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

   ```
   # cat /sys/class/nvme-subsystem/nvme-subsys*/model
   NetApp ONTAP Controller
   NetApp ONTAP Controller
   ```

   ```
   # cat /sys/class/nvme-subsystem/nvme-subsys*/iopolicy
   round-robin
   round-robin
   ```

2. Verify that the namespaces are created.

   ```
   # nvme list
   Node SN Model Namespace Usage Format FW Rev
   ---------------- -------------------- -----------------------
   /dev/nvme0n1 80BADBKnB/JvAAAAAAAC NetApp ONTAP Controller 1 53.69 GB /
   53.69 GB 4 KiB + 0 B FFFFFFFF
   ```

3. Verify the status of the ANA paths.

```
# nvme list-subsys/dev/nvme0n1
Nvme-subsysf0 - NQN=nqn.1992-
08.com.netapp:sn.341541339b9511e8a9b500a098c80f09:subsystem.rhel_141_nvm
e_ss_10_0
\
+- nvme0 fc traddr=nn-0x202c00a098c80f09:pn-0x202d00a098c80f09
host_traddr=nn-0x20000090fae0ec61:pn-0x10000090fae0ec61 live optimized
+- nvme1 fc traddr=nn-0x207300a098dfdd91:pn-0x207600a098dfdd91
host_traddr=nn-0x200000109b1c1204:pn-0x100000109b1c1204 live
inaccessible
+- nvme2 fc traddr=nn-0x207300a098dfdd91:pn-0x207500a098dfdd91
host_traddr=nn-0x200000109b1c1205:pn-0x100000109b1c1205 live optimized
+- nvme3 fc traddr=nn-0x207300a098dfdd91:pn-0x207700a098dfdd91 host
traddr=nn-0x200000109b1c1205:pn-0x100000109b1c1205 live inaccessible
```

4. Verify the NetApp plug-in for ONTAP devices.

```
# nvme netapp ontapdevices -o column
Device    Vserver  Namespace Path             NSID   UUID   Size
-------   --------- -------------------------  ------ ----- -----
/dev/nvme0n1   vs_nvme_10       /vol/rhel_141_vol_10_0/rhel_141_ns_10_0
1        55baf453-f629-4a18-9364-b6aee3f50dad   53.69GB

# nvme netapp ontapdevices -o json
{
   "ONTAPdevices" : [
   {
        Device" : "/dev/nvme0n1",
        "Vserver" : "vs_nvme_10",
        "Namespace_Path" : "/vol/rhel_141_vol_10_0/rhel_141_ns_10_0",
         "NSID" : 1,
         "UUID" : "55baf453-f629-4a18-9364-b6aee3f50dad",
         "Size" : "53.69GB",
         "LBA_Data_Size" : 4096,
         "Namespace_Size" : 13107200
    }
]
```

NVMe/FC host configuration for RHEL 8.1 with ONTAP

Beginning with ONTAP 9.6, NVMe/FC is supported for Red Hat Enterprise Linux (RHEL) 8.1. A RHEL 8.1 host can run both NVMe and SCSI traffic through the same FC initiator adapter ports. See the Hardware Universe for a list of supported FC adapters and controllers.

See the Interoperability Matrix Tool for the current list of supported configurations.

Known limitations

- Native NVMe/FC auto-connect scripts are not available in the `nvme-cli` package. You can use the host bus adapter (HBA) vendor-provided external auto-connect script.
- NVMe multipath is disabled by default. Therefore, you need to enable it manually.
- By default, round-robin load balancing is not enabled. You can enable this functionality by writing a `udev` rule.
- SAN booting using the NVMe-oF protocol is currently not supported.

Enable SAN booting

You can configure your host to use SAN booting to simplify deployment and improve scalability.

Before you begin

Use the Interoperability Matrix Tool to verify that your Linux OS, host bus adapter (HBA), HBA firmware, HBA boot BIOS, and ONTAP version support SAN booting.

Steps

1. Create a SAN boot namespace and map it to the host.

   See Provision NVMe storage.

2. Enable SAN booting in the server BIOS for the ports to which the SAN boot LUN is mapped.

   For information on how to enable the HBA BIOS, see your vendor-specific documentation.

3. Verify that the configuration was successful by rebooting the host and verifying that the OS is up and running.

Enable NVMe/FC

You can use the following procedure to enable NVMe/FC.

Steps

1. Install Red Hat Enterprise Linux 8.1 on the server.
2. After the installation is complete, verify that you are running the specified RHEL kernel:

   ```
   # uname -r
   4.18.0-147.el8.x86_64
   ```

   See the Interoperability Matrix Tool for the current list of supported versions.

3. Install the `nvme-cli-1.8.1-3.el8` package:

   ```
   # rpm -qa|grep nvme-cli
   nvme-cli-1.8.1-3.el8.x86_64
   ```

4. Enable in-kernel NVMe multipath:

```
# grubby -args=nvme_core.multipath=Y -update-kernel /boot/vmlinuz-
4.18.0-147.el8.x86_64
```

5. Add the following string as a separate udev rule at `/lib/udev/rules.d/71-nvme-iopolicy-netapp-ONTAP.rules`. This enables round-robin load balancing for NVMe multipath:

```
# Enable round-robin for NetApp ONTAP
ACTION==”add”, SUBSYSTEM==”nvme-subsystem”, ATTR{model}==”NetApp ONTAP
Controller”, ATTR{iopolicy}=”round-robin
```

6. On the RHEL 8.1 host, check the host NQN string at `/etc/nvme/hostnqn` and verify that it matches the host NQN string for the corresponding subsystem on the ONTAP array:

```
# cat /etc/nvme/hostnqn
nqn.2014-08.org.nvmexpress:uuid:75953f3b-77fe-4e03-bf3c-09d5a156fbcd
```

```
*> vserver nvme subsystem host show -vserver vs_nvme_10
Vserver Subsystem Host NQN
------- --------- -------------------------------------- -----------
rhel_141_nvme_ss_10_0
nqn.2014-08.org.nvmexpress:uuid:75953f3b-77fe-4e03-bf3c-09d5a156fbcd
```

If the host NQN strings do not match, use the `vserver modify` command to update the host NQN string on your corresponding ONTAP array subsystem to match with the host NQN string from `/etc/nvme/hostnqn` on the host.

7. Reboot the host.

Configure the Broadcom FC adapter for NVMe/FC

You can use the following procedure to configure a Broadcom FC adapter.

Steps

1. Verify that you are using the supported adapter. See the Interoperability Matrix Tool for the current list of supported adapters.

```
# cat /sys/class/scsi_host/host*/modelname
LPe32002-M2
LPe32002-M2
```

```
# cat /sys/class/scsi_host/host*/modeldesc
Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
```

2. Copy and install the Broadcom lpfc outbox driver and auto-connect scripts:

```
# tar -xvzf elx-lpfc-dd-rhel8-12.4.243.20-ds-1.tar.gz
# cd elx-lpfc-dd-rhel8-12.4.2453.20-ds-1
# ./elx_lpfc_install-sh -i -n
```

> The native drivers that are bundled with the OS are called the inbox drivers. If you download the outbox drivers (drivers that are not included with an OS release), an auto-connect script is included in the download and should be installed as part of the driver installation process.

3. Reboot the host.
4. Verify that you are using the recommended Broadcom lpfc firmware, outbox driver, and auto-connect package versions:

```
# cat /sys/class/scsi_host/host*/fwrev
12.4.243.20, sil-4.2.c
12.4.243.20, sil-4.2.c
```

```
# cat /sys/module/lpfc/version
0:12.4.243.20
```

```
# rpm -qa | grep nvmefc
nvmefc-connect-12.6.61.0-1.noarch
```

5. Verify that `lpfc_enable_fc4_type` is set to 3:

```
# cat /sys/module/lpfc/parameters/lpfc_enable_fc4_type
3
```

6. Verify that the initiator ports are up and running:

```
# cat /sys/class/fc_host/host*/port_name
0x10000090fae0ec61
0x10000090fae0ec62
```

```
# cat /sys/class/fc_host/host*/port_state
Online
Online
```

7. Verify that the NVMe/FC initiator ports are enabled, running, and you can see the target LIFs:

```
# cat /sys/class/scsi_host/host*/nvme_info
NVME Initiator Enabled
XRI Dist lpfc0 Total 6144 NVME 2947 SCSI 2977 ELS 250
NVME LPORT lpfc0 WWPN x10000090fae0ec61 WWNN x20000090fae0ec61 DID
x012000 ONLINE
NVME RPORT WWPN x202d00a098c80f09 WWNN x202c00a098c80f09 DID x010201
TARGET DISCSRVC ONLINE
NVME RPORT WWPN x203100a098c80f09 WWNN x202c00a098c80f09 DID x010601
TARGET DISCSRVC ONLINE
NVME Statistics
…
```

Enable 1MB I/O Size for Broadcom NVMe/FC

ONTAP reports an MDTS (Max Data Transfer Size) of 8 in the Identify Controller data. This means the maximum I/O request size can be up to 1MB. To issue I/O requests of size 1MB for a Broadcom NVMe/FC host, you should increase the `lpfc` value of the `lpfc_sg_seg_cnt` parameter to 256 from the default value of 64.

> These steps don't apply to Qlogic NVMe/FC hosts.

Steps

1. Set the `lpfc_sg_seg_cnt` parameter to 256:

```
cat /etc/modprobe.d/lpfc.conf
```

You should see an output similar to the following example:

```
options lpfc lpfc_sg_seg_cnt=256
```

2. Run the `dracut -f` command, and reboot the host.
3. Verify that the value for `lpfc_sg_seg_cnt` is 256:

```
cat /sys/module/lpfc/parameters/lpfc_sg_seg_cnt
```

Validate NVMe/FC

You can use the following procedure to validate NVMe/FC.

Steps

1. Verify the following NVMe/FC settings.

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

```
# cat /sys/class/nvme-subsystem/nvme-subsys*/model
NetApp ONTAP Controller
NetApp ONTAP Controller
```

```
# cat /sys/class/nvme-subsystem/nvme-subsys*/iopolicy
round-robin
round-robin
```

2. Verify that the namespaces are created.

```
# nvme list
Node SN Model Namespace Usage Format FW Rev
---------------- -------------------- -----------------------
/dev/nvme0n1 80BADBKnB/JvAAAAAAAC NetApp ONTAP Controller 1 53.69 GB /
53.69 GB 4 KiB + 0 B FFFFFFFF
```

3. Verify the status of the ANA paths.

```
# nvme list-subsys/dev/nvme0n1
Nvme-subsysf0 – NQN=nqn.1992-
08.com.netapp:sn.341541339b9511e8a9b500a098c80f09:subsystem.rhel_141_nvm
e_ss_10_0
\
+- nvme0 fc traddr=nn-0x202c00a098c80f09:pn-0x202d00a098c80f09
host_traddr=nn-0x20000090fae0ec61:pn-0x10000090fae0ec61 live optimized
+- nvme1 fc traddr=nn-0x207300a098dfdd91:pn-0x207600a098dfdd91
host_traddr=nn-0x200000109b1c1204:pn-0x100000109b1c1204 live
inaccessible
+- nvme2 fc traddr=nn-0x207300a098dfdd91:pn-0x207500a098dfdd91
host_traddr=nn-0x200000109b1c1205:pn-0x100000109b1c1205 live optimized
+- nvme3 fc traddr=nn-0x207300a098dfdd91:pn-0x207700a098dfdd91 host
traddr=nn-0x200000109b1c1205:pn-0x100000109b1c1205 live inaccessible
```

4. Verify the NetApp plug-in for ONTAP devices.

```
# nvme netapp ontapdevices -o column
Device   Vserver  Namespace Path             NSID   UUID   Size
-------  -------- ------------------------  ------ ----- -----
/dev/nvme0n1   vs_nvme_10       /vol/rhel_141_vol_10_0/rhel_141_ns_10_0
1        55baf453-f629-4a18-9364-b6aee3f50dad   53.69GB

# nvme netapp ontapdevices -o json
{
   "ONTAPdevices" : [
   {
        Device" : "/dev/nvme0n1",
        "Vserver" : "vs_nvme_10",
        "Namespace_Path" : "/vol/rhel_141_vol_10_0/rhel_141_ns_10_0",
         "NSID" : 1,
         "UUID" : "55baf453-f629-4a18-9364-b6aee3f50dad",
         "Size" : "53.69GB",
         "LBA_Data_Size" : 4096,
         "Namespace_Size" : 13107200
    }
]
```

Rocky Linux

Rocky Linux 10

Configure Rocky Linux 10 for NVMe-oF with ONTAP storage

NetApp SAN host configurations support the NVMe over Fabrics (NVMe-oF) protocol with Asymmetric Namespace Access (ANA). In NVMe-oF environments, ANA is equivalent to asymmetric logical unit access (ALUA) multipathing in iSCSI and FCP environments. ANA is implemented using the in-kernel NVMe multipath feature.

About this task

You can use the following support and features with the NVMe-oF host configuration for Rocky Linux 10. You should also review the known limitations before starting the configuration process.

- Support available:
  - Support for NVMe over TCP (NVMe/TCP) in addition to NVMe over Fibre Channel (NVMe/FC). The NetApp plug-in in the native `nvme-cli` package displays ONTAP details for both NVMe/FC and NVMe/TCP namespaces.
  - Running both NVMe and SCSI traffic on the same host. For example, you can configure dm-multipath for SCSI mpath devices on SCSI LUNs and use NVMe multipath to configure NVMe-oF namespace devices on the host.

    ◦ Beginning with ONTAP 9.12.1, support for secure in-band authentication is introduced for NVMe/TCP. You can use secure in-band authentication for NVMe/TCP with Rocky Linux 10.

  For additional details on supported configurations, see the Interoperability Matrix Tool.

- Features available:
    ◦ Beginning with Rocky Linux 10, native NVMe multipathing is always enabled, and DM multipath support for NVMe-oF is not supported.
- Known limitations:
    ◦ Avoid issuing the `nvme disconnect-all` command on systems booting from SAN over NVMe-TCP or NVMe-FC namespaces because it disconnects both root and data filesystems and might lead to system instability.

Step 1: Optionally, enable SAN booting

You can configure your host to use SAN booting to simplify deployment and improve scalability.

Before you begin

Use the Interoperability Matrix Tool to verify that your Linux OS, host bus adapter (HBA), HBA firmware, HBA boot BIOS, and ONTAP version support SAN booting.

Steps

1. Create a SAN boot namespace and map it to the host.
2. Enable SAN booting in the server BIOS for the ports to which the SAN boot namespace is mapped.

   For information on how to enable the HBA BIOS, see your vendor-specific documentation.

3. Verify that the configuration was successful by rebooting the host and verifying that the OS is up and running.

Step 2: Validate software versions

Use the following procedure to validate the minimum supported Rocky Linux 10 software versions.

Steps

1. Install Rocky Linux 10 on the server. After the installation is complete, verify that you are running the specified Rocky Linux 10 kernel:

   ```
   uname -r
   ```

   The following example shows a Rocky Linux kernel version:

   ```
   6.12.0-55.9.1.el10_0.x86_64
   ```

2. Install the `nvme-cli` package:

   ```
   rpm -qa|grep nvme-cli
   ```

The following example shows an `nvme-cli` package version:

```
nvme-cli-2.11-5.el10.x86_64
```

3. Install the `libnvme` package:

```
rpm -qa|grep libnvme
```

The following example shows an `libnvme` package version:

```
libnvme-1.11.1-1.el10.x86_64
```

4. On the host, check the hostnqn string at `/etc/nvme/hostnqn`:

```
cat /etc/nvme/hostnqn
```

The following example shows an `hostnqn` version:

```
nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
```

5. Verify that the `hostnqn` string matches the `hostnqn` string for the corresponding subsystem on the ONTAP array:

```
::> vserver nvme subsystem host show -vserver vs_nvme_194_rockylinux10
```

Show example

```
Vserver Subsystem Priority  Host NQN
------- --------- --------
------------------------------------------------
vs_ nvme_194_rockylinux10
        nvme4
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048- c7c04f425633
        nvme_1
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048- c7c04f425633
        nvme_2
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048- c7c04f425633
        nvme_3
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048- c7c04f425633
4 entries were displayed.
```

If the `hostnqn` strings do not match, use the `vserver modify` command to update the `hostnqn` string on your corresponding ONTAP array subsystem to match the `hostnqn` string from `/etc/nvme/hostnqn` on the host.

Step 3: Configure NVMe/FC

You can configure NVMe/FC with Broadcom/Emulex FC or Marvell/Qlogic FC adapters. For NVMe/FC configured with a Broadcom adapter, you can enable I/O requests of size 1MB.

Broadcom/Emulex

Configure NVMe/FC for a Broadcom/Emulex adapter.

Steps

1. Verify that you are using the supported adapter model:
   a. Display the model names:

      ```
      cat /sys/class/scsi_host/host*/modelname
      ```

      You should see the following output:

      ```
      LPe36002-M64
      LPe36002-M64
      ```

   b. Display the model descriptions:

      ```
      cat /sys/class/scsi_host/host*/modeldesc
      ```

      You should see an output similar to the following example:

      ```
      Emulex LightPulse LPe36002-M64 2-Port 64Gb Fibre Channel Adapter
      Emulex LightPulse LPe36002-M64 2-Port 64Gb Fibre Channel Adapter
      ```

2. Verify that you are using the recommended Broadcom `lpfc` firmware and inbox driver:
   a. Display the firmware version:

      ```
      cat /sys/class/scsi_host/host*/fwrev
      ```

      The following example shows firmware versions:

      ```
      14.0.539.16, sli-4:6:d
      14.0.539.16, sli-4:6:d
      ```

   b. Display the inbox driver version:

      ```
      cat /sys/module/lpfc/version
      ```

      The following example shows a driver version:

```
0:14.4.0.6
```

For the current list of supported adapter driver and firmware versions, see the Interoperability Matrix Tool.

3. Verify that the expected output of `lpfc_enable_fc4_type` is set to `3`:

```
cat /sys/module/lpfc/parameters/lpfc_enable_fc4_type
```

4. Verify that you can view your initiator ports:

```
cat /sys/class/fc_host/host*/port_name
```

The following example shows port identities:

```
0x2100f4c7aa0cd7c2
0x2100f4c7aa0cd7c3
```

5. Verify that your initiator ports are online:

```
cat /sys/class/fc_host/host*/port_state
```

You should see the following output:

```
Online
Online
```

6. Verify that the NVMe/FC initiator ports are enabled and that the target ports are visible:

```
cat /sys/class/scsi_host/host*/nvme_info
```

Show example

```
NVME Initiator Enabled
XRI Dist lpfc2 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc2 WWPN x100000109bf044b1 WWNN x200000109bf044b1
DID x022a00 ONLINE
NVME RPORT       WWPN x202fd039eaa7dfc8 WWNN x202cd039eaa7dfc8
DID x021310 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x202dd039eaa7dfc8 WWNN x202cd039eaa7dfc8
DID x020b10 TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 0000000810 Cmpl 0000000810 Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000007b098f07 Issue 000000007aee27c4 OutIO
fffffffffe498bd
        abort 000013b4 noxri 00000000 nondlp 00000058 qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 000013b4 Err 00021443

NVME Initiator Enabled
XRI Dist lpfc3 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc3 WWPN x100000109bf044b2 WWNN x200000109bf044b2
DID x021b00 ONLINE
NVME RPORT       WWPN x2033d039eaa7dfc8 WWNN x202cd039eaa7dfc8
DID x020110 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2032d039eaa7dfc8 WWNN x202cd039eaa7dfc8
DID x022910 TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 0000000840 Cmpl 0000000840 Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000007afd4434 Issue 000000007ae31b83 OutIO
fffffffffe5d74f
        abort 000014a5 noxri 00000000 nondlp 0000006a qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 000014a5 Err 0002149a
```

Marvell/QLogic

Configure NVMe/FC for a Marvell/QLogic adapter.

Steps

1. Verify that you are running the supported adapter driver and firmware versions:

```
cat /sys/class/fc_host/host*/symbolic_name
```

The follow example shows driver and firmware versions:

```
QLE2872 FW:v9.15.00 DVR:v10.02.09.300-k
QLE2872 FW:v9.15.00 DVR:v10.02.09.300-k
```

2. Verify that `ql2xnvmeenable` is set. This enables the Marvell adapter to function as an NVMe/FC initiator:

```
cat /sys/module/qla2xxx/parameters/ql2xnvmeenable
```

The expected ouptut is 1.

Step 4: Optionally, enable 1MB I/O

ONTAP reports an MDTS (Max Data Transfer Size) of 8 in the Identify Controller data. This means the maximum I/O request size can be up to 1MB. To issue I/O requests of size 1MB for a Broadcom NVMe/FC host, you should increase the `lpfc` value of the `lpfc_sg_seg_cnt` parameter to 256 from the default value of 64.

> These steps don't apply to Qlogic NVMe/FC hosts.

Steps

1. Set the `lpfc_sg_seg_cnt` parameter to 256:

```
cat /etc/modprobe.d/lpfc.conf
```

You should see an output similar to the following example:

```
options lpfc lpfc_sg_seg_cnt=256
```

2. Run the `dracut -f` command, and reboot the host.
3. Verify that the value for `lpfc_sg_seg_cnt` is 256:

```
cat /sys/module/lpfc/parameters/lpfc_sg_seg_cnt
```

Step 5: Verify NVMe boot services

With Rocky Linux 10, the `nvmefc-boot-connections.service` and `nvmf-autoconnect.service` boot services included in the NVMe/FC `nvme-cli` package are automatically enabled when the system boots.

After booting completes, verify that the `nvmefc-boot-connections.service` and `nvmf-autoconnect.service` boot services are enabled.

Steps

1. Verify that `nvmf-autoconnect.service` is enabled:

```
systemctl status nvmf-autoconnect.service
```

Show example output

```
nvmf-autoconnect.service - Connect NVMe-oF subsystems automatically
during boot
     Loaded: loaded (/usr/lib/systemd/system/nvmf-
autoconnect.service; enabled; preset: disabled)
     Active: inactive (dead)

Jun 10 04:06:26 SR630-13-201.lab.eng.btc.netapp.in systemd[1]:
Starting Connect NVMe-oF subsystems automatically during boot...
Jun 10 04:06:26 SR630-13-201.lab.eng.btc.netapp.in systemd[1]: nvmf-
autoconnect.service: Deactivated successfully.
Jun 10 04:06:26 SR630-13-201.lab.eng.btc.netapp.in systemd[1]:
Finished Connect NVMe-oF subsystems automatically during boot.
```

2. Verify that `nvmefc-boot-connections.service` is enabled:

```
systemctl status nvmefc-boot-connections.service
```

Show example output

```
nvmefc-boot-connections.service - Auto-connect to subsystems on FC-
NVME devices found during boot
     Loaded: loaded (/usr/lib/systemd/system/nvmefc-boot-
connections.service; enabled; preset: enabled)
     Active: inactive (dead) since Tue 2025-06-10 01:08:36 EDT; 2h
59min ago
   Main PID: 7090 (code=exited, status=0/SUCCESS)
        CPU: 30ms

Jun 10 01:08:36 localhost systemd[1]: Starting Auto-connect to
subsystems on FC-NVME devices found during boot...
Jun 10 01:08:36 localhost systemd[1]: nvmefc-boot-
connections.service: Deactivated successfully.
Jun 10 01:08:36 localhost systemd[1]: Finished Auto-connect to
subsystems on FC-NVME devices found during boot.
```

Step 6: Configure NVMe/TCP

The NVMe/TCP protocol doesn't support the `auto-connect` operation. Instead, you can discover the NVMe/TCP subsystems and namespaces by performing the NVMe/TCP `connect` or `connect-all` operations manually.

Steps

1. Verify that the initiator port can fetch the discovery log page data across the supported NVMe/TCP LIFs:

```
nvme discover -t tcp -w host-traddr -a traddr
```

Show example

```
nvme discover -t tcp -w 192.168.20.1 -a 192.168.20.20

Discovery Log Number of Records 8, Generation counter 18
=====Discovery Log Entry 0======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  4
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.64e65e6caae711ef9668d039ea951c46:discovery
traddr:  192.168.21.21
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 1======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  2
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.64e65e6caae711ef9668d039ea951c46:discovery
traddr:  192.168.20.21
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 2======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  3
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.64e65e6caae711ef9668d039ea951c46:discovery
traddr:  192.168.21.20
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 3======
trtype:  tcp
```

```
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  1
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.64e65e6caae711ef9668d039ea951c46:discovery
traddr:  192.168.20.20
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 4======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.64e65e6caae711ef9668d039ea951c46:subsystem.rockylin
ux10_tcp_subsystem
traddr:  192.168.21.21
eflags:  none
sectype: none
=====Discovery Log Entry 5======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.64e65e6caae711ef9668d039ea951c46:subsystem.rockylin
ux10_tcp_subsystem
traddr:  192.168.20.21
eflags:  none
sectype: none
=====Discovery Log Entry 6======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  3
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.64e65e6caae711ef9668d039ea951c46:subsystem.rockylin
```

```
ux10_tcp_subsystem
traddr:  192.168.21.20
eflags:  none
sectype: none
=====Discovery Log Entry 7======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.64e65e6caae711ef9668d039ea951c46:subsystem.rockylin
ux10_tcp_subsystem
traddr:  192.168.20.20
eflags:  none
sectype: none
```

2. Verify that the other NVMe/TCP initiator-target LIF combinations can successfully fetch discovery log page data:

```
nvme discover -t tcp -w host-traddr -a traddr
```

Show example

```
nvme discover -t tcp -w 192.168.20.1 -a 192.168.20.20
nvme discover -t tcp -w 192.168.21.1 -a 192.168.21.20
nvme discover -t tcp -w 192.168.20.1 -a 192.168.20.21
nvme discover -t tcp -w 192.168.21.1 -a 192.168.21.21
```

3. Run the `nvme connect-all` command across all the supported NVMe/TCP initiator-target LIFs across the nodes:

```
nvme connect-all -t tcp -w host-traddr -a traddr
```

Show example

```
nvme    connect-all -t  tcp -w  192.168.20.1    -a  192.168.20.20
nvme    connect-all -t  tcp -w  192.168.21.1    -a  192.168.21.20
nvme    connect-all -t  tcp -w  192.168.20.1    -a  192.168.20.21
nvme    connect-all -t  tcp -w  192.168.21.1    -a  192.168.21.21
```

Beginning with Rocky Linux 9.4, the setting for the NVMe/TCP `ctrl_loss_tmo timeout` is automatically set to "off". As a result:

- There are no limits on the number of retries (indefinite retry).
- You don’t need to manually configure a specific `ctrl_loss_tmo timeout` duration when using the `nvme connect` or `nvme connect-all` commands (option -l).
- The NVMe/TCP controllers don’t experience timeouts in the event of a path failure and remain connected indefinitely.

Step 7: Validate NVMe-oF

Verify that the in-kernel NVMe multipath status, ANA status, and ONTAP namespaces are correct for the NVMe-oF configuration.

Steps

1. Verify that the in-kernel NVMe multipath is enabled:

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

   You should see the following output:

```
Y
```

2. Verify that the appropriate NVMe-oF settings (such as, model set to NetApp ONTAP Controller and load balancing iopolicy set to round-robin) for the respective ONTAP namespaces correctly reflect on the host:

   a. Display the subsystems:

```
cat /sys/class/nvme-subsystem/nvme-subsys*/model
```

   You should see the following output:

```
NetApp ONTAP Controller
NetApp ONTAP Controller
```

b. Display the policy:

```
cat /sys/class/nvme-subsystem/nvme-subsys*/iopolicy
```

You should see the following output:

```
round-robin
round-robin
```

3. Verify that the namespaces are created and correctly discovered on the host:

```
nvme list
```

Show example

```
Node         SN                   Model
---------------------------------------------------------
/dev/nvme4n1 81Ix2BVuekWcAAAAAAAB   NetApp ONTAP Controller


Namespace Usage    Format             FW             Rev
-----------------------------------------------------------
1                 21.47 GB / 21.47 GB   4 KiB + 0 B   FFFFFFFF
```

4. Verify that the controller state of each path is live and has the correct ANA status:

NVMe/FC

```
nvme list-subsys /dev/nvme5n1
```

Show example

```
nvme-subsys5 - NQN=nqn.1992-
08.com.netapp:sn.f7565b15a66911ef9668d039ea951c46:subsystem.nvme
1
               hostnqn=nqn.2014-08.org.nvmexpress:uuid:4c4c4544-
0056-5410-8048-c7c04f425633
\
 +- nvme126 fc traddr=nn-0x2036d039ea951c45:pn-
0x2038d039ea951c45,host_traddr=nn-0x2000f4c7aa0cd7c3:pn-
0x2100f4c7aa0cd7c3 live optimized
 +- nvme176 fc traddr=nn-0x2036d039ea951c45:pn-
0x2037d039ea951c45,host_traddr=nn-0x2000f4c7aa0cd7c2:pn-
0x2100f4c7aa0cd7c2 live optimized
 +- nvme5 fc traddr=nn-0x2036d039ea951c45:pn-
0x2039d039ea951c45,host_traddr=nn-0x2000f4c7aa0cd7c2:pn-
0x2100f4c7aa0cd7c2 live non-optimized
 +- nvme71 fc traddr=nn-0x2036d039ea951c45:pn-
0x203ad039ea951c45,host_traddr=nn-0x2000f4c7aa0cd7c3:pn-
0x2100f4c7aa0cd7c3 live non-optimized
```

NVMe/TCP

```
nvme list-subsys /dev/nvme4n2
```

Show example

```
nvme-subsys4 - NQN=nqn.1992-
08.com.netapp:sn.64e65e6caae711ef9668d039ea951c46:subsystem.nvme
4
               hostnqn=nqn.2014-08.org.nvmexpress:uuid:4c4c4544-
0035-5910-804b-c2c04f444d33
\
+- nvme102 tcp
traddr=192.168.21.20,trsvcid=4420,host_traddr=192.168.21.1,src_a
ddr=192.168.21.1 live non-optimized
+- nvme151 tcp
traddr=192.168.21.21,trsvcid=4420,host_traddr=192.168.21.1,src_a
ddr=192.168.21.1 live optimized
+- nvme4 tcp
traddr=192.168.20.20,trsvcid=4420,host_traddr=192.168.20.1,src_a
ddr=192.168.20.1 live non-optimized
+- nvme53 tcp
traddr=192.168.20.21,trsvcid=4420,host_traddr=192.168.20.1,src_a
ddr=192.168.20.1 live optimized
```

5. Verify that the NetApp plug-in displays the correct values for each ONTAP namespace device:

Column

```
nvme netapp ontapdevices -o column
```

Show example

```
Device        Vserver   Namespace Path
----------------------- ------------------------------
/dev/nvme10n1     vs_tcp_rockylinux10       /vol/vol10/ns10

NSID       UUID                                   Size
------------------------------ ------------------------------
1    bbf51146-fc64-4197-b8cf-8a24f6f359b3   21.47GB
```

JSON

```
nvme netapp ontapdevices -o json
```

Show example

```
{
  "ONTAPdevices":[
    {
      "Device":"/dev/nvme10n1",
      "Vserver":"vs_tcp_rockylinux10",
      "Namespace_Path":"/vol/vol10/ns10",
      "NSID":1,
      "UUID":"bbf51146-fc64-4197-b8cf-8a24f6f359b3",
      "Size":"21.47GB",
      "LBA_Data_Size":4096,
      "Namespace_Size":5242880
}
]
    }
```

Step 8: Set up secure in-band authentication

Beginning with ONTAP 9.12.1, secure in-band authentication is supported over NVMe/TCP between a Rocky Linux 10 host and an ONTAP controller.

Each host or controller must be associated with a `DH-HMAC-CHAP` key to set up secure authentication. A `DH-`

`HMAC-CHAP` key is a combination of the NQN of the NVMe host or controller and an authentication secret configured by the administrator. To authenticate its peer, an NVMe host or controller must recognize the key associated with the peer.

Set up secure in-band authentication using the CLI or a config JSON file. If you need to specify different dhchap keys for different subsystems, you must use a config JSON file.

CLI

Set up secure in-band authentication using the CLI.

Steps

1. Obtain the host NQN:

```
cat /etc/nvme/hostnqn
```

2. Generate the dhchap key for the Rocky Linux 10 host.

   The following output describes the `gen-dhchap-key` command paramters:

```
nvme gen-dhchap-key -s optional_secret -l key_length {32|48|64} -m
HMAC_function {0|1|2|3} -n host_nqn
•   -s secret key in hexadecimal characters to be used to initialize
the host key
•   -l length of the resulting key in bytes
•   -m HMAC function to use for key transformation
0 = none, 1- SHA-256, 2 = SHA-384, 3=SHA-512
•   -n host NQN to use for key transformation
```

   In the following example, a random dhchap key with HMAC set to 3 (SHA-512) is generated.

```
nvme gen-dhchap-key -m 3 -n nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0035-5910-804b-c2c04f444d33
DHHC-
1:03:7zf8I9gaRcDWH3tCH5vLGaoyjzPIvwNWusBfKdpJa+hia1aKDKJQ2o53pX3wYM9
xdv5DtKNNhJInZ7X8wU2RQpQIngc=:
```

3. On the ONTAP controller, add the host and specify both dhchap keys:

```
vserver nvme subsystem host add -vserver <svm_name> -subsystem
<subsystem> -host-nqn <host_nqn> -dhchap-host-secret
<authentication_host_secret> -dhchap-controller-secret
<authentication_controller_secret> -dhchap-hash-function {sha-
256|sha-512} -dhchap-group {none|2048-bit|3072-bit|4096-bit|6144-
bit|8192-bit}
```

4. A host supports two types of authentication methods, unidirectional and bidirectional. On the host, connect to the ONTAP controller and specify dhchap keys based on the chosen authentication method:

```
nvme connect -t tcp -w <host-traddr> -a <tr-addr> -n <host_nqn> -S
<authentication_host_secret> -C <authentication_controller_secret>
```

5. Validate the `nvme connect authentication` command by verifying the host and controller dhchap keys:

   a. Verify the host dhchap keys:

   ```
   cat /sys/class/nvme-subsystem/<nvme-subsysX>/nvme*/dhchap_secret
   ```

   Show example output for a unidirectional configuration

   ```
   cat /sys/class/nvme-subsystem/nvme-subsys1/nvme*/dhchap_secret
   DHHC-
   1:03:fMCrJharXUOqRoIsOEaG6m2PH1yYvu5+z3jTmzEKUbcWu26I33b93b
   il2WR09XDho/ld3L45J+0FeCsStBEAfhYgkQU=:
   DHHC-
   1:03:fMCrJharXUOqRoIsOEaG6m2PH1yYvu5+z3jTmzEKUbcWu26I33b93b
   il2WR09XDho/ld3L45J+0FeCsStBEAfhYgkQU=:
   DHHC-
   1:03:fMCrJharXUOqRoIsOEaG6m2PH1yYvu5+z3jTmzEKUbcWu26I33b93b
   il2WR09XDho/ld3L45J+0FeCsStBEAfhYgkQU=:
   DHHC-
   1:03:fMCrJharXUOqRoIsOEaG6m2PH1yYvu5+z3jTmzEKUbcWu26I33b93b
   il2WR09XDho/ld3L45J+0FeCsStBEAfhYgkQU=:
   ```

   b. Verify the controller dhchap keys:

   ```
   cat /sys/class/nvme-subsystem/<nvme-
   subsysX>/nvme*/dhchap_ctrl_secret
   ```

Show example output for a bidirectional configuration

```
cat /sys/class/nvme-subsystem/nvme-
subsys6/nvme*/dhchap_ctrl_secret
DHHC- 1:03:7zf8I9gaRcDWH3tCH5vLGaoyjzPIvwNWusBfKdpJa+hia
1aKDKJQ2o53pX3wYM9xdv5DtKNNhJInZ7X8wU2RQpQIngc=:

DHHC- 1:03:7zf8I9gaRcDWH3tCH5vLGaoyjzPIvwNWusBfKdpJa+hia
1aKDKJQ2o53pX3wYM9xdv5DtKNNhJInZ7X8wU2RQpQIngc=:

DHHC- 1:03:7zf8I9gaRcDWH3tCH5vLGaoyjzPIvwNWusBfKdpJa+hia
1aKDKJQ2o53pX3wYM9xdv5DtKNNhJInZ7X8wU2RQpQIngc=:

DHHC- 1:03:7zf8I9gaRcDWH3tCH5vLGaoyjzPIvwNWusBfKdpJa+hia
1aKDKJQ2o53pX3wYM9xdv5DtKNNhJInZ7X8wU2RQpQIngc=:
```

JSON file

When multiple NVMe subsystems are available on the ONTAP controller configuration, you can use the `/etc/nvme/config.json` file with the `nvme connect-all` command.

Use the `-o` option to generate the JSON file. See the NVMe connect-all manual pages for more syntax options.

Steps

1. Configure the JSON file.

   > In the following example, `dhchap_key` corresponds to `dhchap_secret` and `dhchap_ctrl_key` corresponds to `dhchap_ctrl_secret`.

Show example

```
cat /etc/nvme/config.json
[
{
"hostnqn":"nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0035-5910-
804b-c2c04f444d33",
"hostid":"4c4c4544-0035-5910-804b-c2c04f444d33",
"dhchap_key":"DHHC-
1:03:7zf8I9gaRcDWH3tCH5vLGaoyjzPIvwNWusBfKdpJa+hia1aKDKJQ2o53pX3
wYM9xdv5DtKNNhJInZ7X8wU2RQpQIngc=:",
"subsystems":[
{
"nqn":"nqn.1992-
08.com.netapp:sn.127ade26168811f0a50ed039eab69ad3:subsystem.inba
nd_unidirectional",
"ports":[
{
"transport":"tcp",
"traddr":"192.168.20.17",
"host_traddr":"192.168.20.1",
"trsvcid":"4420"
},
{
"transport":"tcp",
"traddr":"192.168.20.18",
"host_traddr":"192.168.20.1",
"trsvcid":"4420"
},
{
"transport":"tcp",
"traddr":"192.168.21.18",
"host_traddr":"192.168.21.1",
"trsvcid":"4420"
},
{
"transport":"tcp",
"traddr":"192.168.21.17",
"host_traddr":"192.168.21.1",
"trsvcid":"4420"
}]
```

2. Connect to the ONTAP controller using the config JSON file:

```
nvme connect-all -J /etc/nvme/config.json
```

Show example

```
traddr=192.168.20.20 is already connected
traddr=192.168.20.20 is already connected
traddr=192.168.20.20 is already connected
traddr=192.168.20.20 is already connected
traddr=192.168.20.20 is already connected
traddr=192.168.20.20 is already connected
traddr=192.168.20.20 is already connected
traddr=192.168.20.20 is already connected
traddr=192.168.20.21 is already connected
traddr=192.168.20.21 is already connected
traddr=192.168.20.21 is already connected
traddr=192.168.20.21 is already connected
traddr=192.168.20.21 is already connected
traddr=192.168.20.21 is already connected
traddr=192.168.20.21 is already connected
traddr=192.168.20.21 is already connected
```

3. Verify that the dhchap secrets have been enabled for the respective controllers for each subsystem.
   a. Verify the host dhchap keys:

   ```
   cat /sys/class/nvme-subsystem/nvme-subsys0/nvme0/dhchap_secret
   ```

   The following example shows a dhchap key:

   ```
   DHHC-1:03:7zf8I9gaRcDWH3tCH5vLGaoyjzPIvwNWusBfKdpJa+hia1
   aKDKJQ2o53pX3wYM9xdv5DtKNNhJInZ7X8wU2RQpQIngc=:
   ```

   b. Verify the controller dhchap keys:

   ```
   cat /sys/class/nvme-subsystem/nvme-
   subsys0/nvme0/dhchap_ctrl_secret
   ```

   You should see an output similar to the following example:

```
DHHC-1:03:fMCrJharXUOqRoIsOEaG6m2PH1yYvu5+z3jT
mzEKUbcWu26I33b93bil2WR09XDho/ld3L45J+0FeCsStBEAfhYgkQU=:
```

Step 9: Review the known issues

There are no known issues.

Rocky Linux 9

Configure Rocky Linux 9.6 for NVMe-oF with ONTAP storage

NetApp SAN host configurations support the NVMe over Fabrics (NVMe-oF) protocol with Asymmetric Namespace Access (ANA). In NVMe-oF environments, ANA is equivalent to asymmetric logical unit access (ALUA) multipathing in iSCSI and FCP environments. ANA is implemented using the in-kernel NVMe multipath feature.

About this task

You can use the following support and features with the NVMe-oF host configuration for Rocky Linux 9.6. You should also review the known limitations before starting the configuration process.

- Support available:
  - Support for NVMe over TCP (NVMe/TCP) in addition to NVMe over Fibre Channel (NVMe/FC). The NetApp plug-in in the native `nvme-cli` package displays ONTAP details for both NVMe/FC and NVMe/TCP namespaces.
  - Running both NVMe and SCSI traffic on the same host. For example, you can configure dm-multipath for SCSI mpath devices on SCSI LUNs and use NVMe multipath to configure NVMe-oF namespace devices on the host.
  - Rocky Linux 9.6 enables in-kernel NVMe multipath for NVMe namespaces by default, removing the need for explicit settings.
  - Beginning with ONTAP 9.12.1, support for secure in-band authentication is introduced for NVMe/TCP. You can use secure in-band authentication for NVMe/TCP with Rocky Linux 9.6.

  For additional details on supported configurations, see the Interoperability Matrix Tool.
- Features available:
  - There are no new features in this release.
- Known limitations:
  - Avoid issuing the `nvme disconnect-all` command on systems booting from SAN over NVMe-TCP or NVMe-FC namespaces because it disconnects both root and data filesystems and might lead to system instability.

Step 1: Optionally, enable SAN booting

You can configure your host to use SAN booting to simplify deployment and improve scalability.

Before you begin

Use the Interoperability Matrix Tool to verify that your Linux OS, host bus adapter (HBA), HBA firmware, HBA boot BIOS, and ONTAP version support SAN booting.

Steps

1. Create a SAN boot namespace and map it to the host.
2. Enable SAN booting in the server BIOS for the ports to which the SAN boot namespace is mapped.

   For information on how to enable the HBA BIOS, see your vendor-specific documentation.
3. Verify that the configuration was successful by rebooting the host and verifying that the OS is up and running.

Step 2: Validate software versions

Use the following procedure to validate the minimum supported Rocky Linux 9.6 software versions.

Steps

1. Install Rocky Linux 9.6 on the server. After the installation is complete, verify that you are running the specified Rocky Linux 9.6 kernel:

   ```
   uname -r
   ```

   The following example shows a Rocky Linux kernel version:

   ```
   5.14.0-570.12.1.el9_6.x86_64
   ```
2. Install the `nvme-cli` package:

   ```
   rpm -qa|grep nvme-cli
   ```

   The following example shows an nvme-cli package version:

   ```
   nvme-cli-2.11-5.el9.x86_64
   ```
3. Install the `libnvme` package:

   ```
   rpm -qa|grep libnvme
   ```

   The following example shows an `libnvme` package version:

   ```
   libnvme-1.11.1-1.el9.x86_64
   ```
4. On the Rocky Linux host, check the hostnqn string at `/etc/nvme/hostnqn`:

```
cat /etc/nvme/hostnqn
```

The following example shows an `hostnqn` version:

```
nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
```

5. Verify that the `hostnqn` string matches the `hostnqn` string for the corresponding subsystem on the ONTAP array:

```
::> vserver nvme subsystem host show -vserver vs_coexistence_LPE36002
```

Show example

```
Vserver Subsystem Priority  Host NQN
------- --------- --------
------------------------------------------------
vs_coexistence_LPE36002
        nvme
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
        nvme_1
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
        nvme_2
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
        nvme_3
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
4 entries were displayed.
```

If the `hostnqn` strings do not match, use the `vserver modify` command to update the `hostnqn` string on your corresponding ONTAP array subsystem to match the `hostnqn` string from `/etc/nvme/hostnqn` on the host.

Step 3: Configure NVMe/FC

You can configure NVMe/FC with Broadcom/Emulex FC or Marvell/Qlogic FC adapters. For NVMe/FC configured with a Broadcom adapter, you can enable I/O requests of size 1MB.

Broadcom/Emulex

Configure NVMe/FC for a Broadcom/Emulex adapter.

Steps

1. Verify that you are using the supported adapter model:

   a. Display the model names:

   ```
   cat /sys/class/scsi_host/host*/modelname
   ```

   You should see the following output:

   ```
   LPe36002-M64
   LPe36002-M64
   ```

   b. Display the model descriptions:

   ```
   cat /sys/class/scsi_host/host*/modeldesc
   ```

   You should see an output similar to the following example:

   ```
   Emulex LightPulse LPe36002-M64 2-Port 64Gb Fibre Channel Adapter
   Emulex LightPulse LPe36002-M64 2-Port 64Gb Fibre Channel Adapter
   ```

2. Verify that you are using the recommended Broadcom `lpfc` firmware and inbox driver:

   a. Display the firmware version:

   ```
   cat /sys/class/scsi_host/host*/fwrev
   ```

   The following example shows firmware versions:

   ```
   14.0.539.16, sli-4:6:d
   14.0.539.16, sli-4:6:d
   ```

   b. Display the inbox driver version:

   ```
   cat /sys/module/lpfc/version
   ```

   The following example shows a driver version:

```
0:14.4.0.6
```

For the current list of supported adapter driver and firmware versions, see the Interoperability Matrix Tool.

3. Verify that the expected output of `lpfc_enable_fc4_type` is set to `3`:

```
cat /sys/module/lpfc/parameters/lpfc_enable_fc4_type
```

4. Verify that you can view your initiator ports:

```
cat /sys/class/fc_host/host*/port_name
```

The following example shows port identities:

```
0x2100f4c7aa0cd7c2
0x2100f4c7aa0cd7c3
```

5. Verify that your initiator ports are online:

```
cat /sys/class/fc_host/host*/port_state
```

You should see the following output:

```
Online
Online
```

6. Verify that the NVMe/FC initiator ports are enabled and that the target ports are visible:

```
cat /sys/class/scsi_host/host*/nvme_info
```

Show example

```
NVME Initiator Enabled
XRI Dist lpfc0 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc0 WWPN x100000109b954518 WWNN x200000109b954518
DID x000000 ONLINE

NVME Statistics
LS: Xmt 0000000000 Cmpl 0000000000 Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 0000000000000000 Issue 0000000000000000 OutIO
0000000000000000
          abort 00000000 noxri 00000000 nondlp 00000000 qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 00000000 Err 00000000

NVME Initiator Enabled
XRI Dist lpfc1 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc1 WWPN x100000109b954519 WWNN x200000109b954519
DID x020500 ONLINE

NVME Statistics
LS: Xmt 0000000000 Cmpl 0000000000 Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 0000000000000000 Issue 0000000000000000 OutIO
0000000000000000
         abort 00000000 noxri 00000000 nondlp 00000000 qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 00000000 Err 00000000

NVME Initiator Enabled
XRI Dist lpfc2 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc2 WWPN x100000109bf044b1 WWNN x200000109bf044b1
DID x022a00 ONLINE
NVME RPORT       WWPN x200bd039eaa7dfc8 WWNN x2008d039eaa7dfc8
DID x021319 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2155d039eaa7dfc8 WWNN x2154d039eaa7dfc8
DID x02130f TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2001d039eaa7dfc8 WWNN x2000d039eaa7dfc8
DID x021310 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x200dd039eaa7dfc8 WWNN x2008d039eaa7dfc8
DID x020b15 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2156d039eaa7dfc8 WWNN x2154d039eaa7dfc8
DID x020b0d TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2003d039eaa7dfc8 WWNN x2000d039eaa7dfc8
DID x020b10 TARGET DISCSRVC ONLINE
```

```
NVME Statistics
LS: Xmt 0000003049 Cmpl 0000003049 Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 0000000018f9450b Issue 0000000018f5de57 OutIO
fffffffffffc994c
          abort 000036d3 noxri 00000313 nondlp 00000c8d qdepth
00000000 wqerr 00000064 err 00000000
FCP CMPL: xb 000036d1 Err 000fef0f

NVME Initiator Enabled
XRI Dist lpfc3 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc3 WWPN x100000109bf044b2 WWNN x200000109bf044b2
DID x021b00 ONLINE
NVME RPORT       WWPN x2062d039eaa7dfc8 WWNN x2008d039eaa7dfc8
DID x022915 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2157d039eaa7dfc8 WWNN x2154d039eaa7dfc8
DID x02290f TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2002d039eaa7dfc8 WWNN x2000d039eaa7dfc8
DID x022910 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2065d039eaa7dfc8 WWNN x2008d039eaa7dfc8
DID x020119 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2158d039eaa7dfc8 WWNN x2154d039eaa7dfc8
DID x02010d TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2004d039eaa7dfc8 WWNN x2000d039eaa7dfc8
DID x020110 TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 0000002f2c Cmpl 0000002f2c Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000001aaf3eb5 Issue 000000001aab4373 OutIO
fffffffffffc04be
          abort 000035cc noxri 0000038c nondlp 000009e3 qdepth
00000000 wqerr 00000082 err 00000000
FCP CMPL: xb 000035cc Err 000fcfc0
```

Marvell/QLogic

Configure NVMe/FC for a Marvell/QLogic adapter.

Steps

1. Verify that you are running the supported adapter driver and firmware versions:

```
cat /sys/class/fc_host/host*/symbolic_name
```

The follow example shows driver and firmware versions:

```
QLE2872 FW:v9.15.00 DVR:v10.02.09.300-k
QLE2872 FW:v9.15.00 DVR:v10.02.09.300-k
```

2. Verify that `ql2xnvmeenable` is set. This enables the Marvell adapter to function as an NVMe/FC initiator:

```
cat /sys/module/qla2xxx/parameters/ql2xnvmeenable
```

The expected ouptut is 1.

Step 4: Optionally, enable 1MB I/O

ONTAP reports an MDTS (Max Data Transfer Size) of 8 in the Identify Controller data. This means the maximum I/O request size can be up to 1MB. To issue I/O requests of size 1MB for a Broadcom NVMe/FC host, you should increase the `lpfc` value of the `lpfc_sg_seg_cnt` parameter to 256 from the default value of 64.

> ⓘ These steps don't apply to Qlogic NVMe/FC hosts.

Steps

1. Set the `lpfc_sg_seg_cnt` parameter to 256:

```
cat /etc/modprobe.d/lpfc.conf
```

You should see an output similar to the following example:

```
options lpfc lpfc_sg_seg_cnt=256
```

2. Run the `dracut -f` command, and reboot the host.
3. Verify that the value for `lpfc_sg_seg_cnt` is 256:

```
cat /sys/module/lpfc/parameters/lpfc_sg_seg_cnt
```

Step 5: Verify NVMe boot services

With Rocky Linux 9.6, the `nvmefc-boot-connections.service` and `nvmf-autoconnect.service` boot services included in the NVMe/FC `nvme-cli` package are automatically enabled when the system boots.

After booting completes, verify that the `nvmefc-boot-connections.service` and `nvmf-autoconnect.service` boot services are enabled.

Steps

1. Verify that `nvmf-autoconnect.service` is enabled:

```
systemctl status nvmf-autoconnect.service
```

Show example output

```
nvmf-autoconnect.service - Connect NVMe-oF subsystems automatically
during boot
     Loaded: loaded (/usr/lib/systemd/system/nvmf-
autoconnect.service; enabled; preset: disabled)
     Active: inactive (dead)

Jun 10 04:06:26 SR630-13-201.lab.eng.btc.netapp.in systemd[1]:
Starting Connect NVMe-oF subsystems automatically during boot...
Jun 10 04:06:26 SR630-13-201.lab.eng.btc.netapp.in systemd[1]: nvmf-
autoconnect.service: Deactivated successfully.
Jun 10 04:06:26 SR630-13-201.lab.eng.btc.netapp.in systemd[1]:
Finished Connect NVMe-oF subsystems automatically during boot.
```

2. Verify that `nvmefc-boot-connections.service` is enabled:

```
systemctl status nvmefc-boot-connections.service
```

Show example output

```
nvmefc-boot-connections.service - Auto-connect to subsystems on FC-
NVME devices found during boot
     Loaded: loaded (/usr/lib/systemd/system/nvmefc-boot-
connections.service; enabled; preset: enabled)
     Active: inactive (dead) since Tue 2025-06-10 01:08:36 EDT; 2h
59min ago
   Main PID: 7090 (code=exited, status=0/SUCCESS)
        CPU: 30ms

Jun 10 01:08:36 localhost systemd[1]: Starting Auto-connect to
subsystems on FC-NVME devices found during boot...
Jun 10 01:08:36 localhost systemd[1]: nvmefc-boot-
connections.service: Deactivated successfully.
Jun 10 01:08:36 localhost systemd[1]: Finished Auto-connect to
subsystems on FC-NVME devices found during boot.
```

Step 6: Configure NVMe/TCP

The NVMe/TCP protocol doesn’t support the `auto-connect` operation. Instead, you can discover the NVMe/TCP subsystems and namespaces by performing the NVMe/TCP `connect` or `connect-all` operations manually.

Steps

1. Verify that the initiator port can fetch the discovery log page data across the supported NVMe/TCP LIFs:

```
nvme discover -t tcp -w host-traddr -a traddr
```

Show example

```
nvme discover -t tcp -w 192.168.1.31 -a 192.168.1.24

Discovery Log Number of Records 20, Generation counter 25
=====Discovery Log Entry 0======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  4
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr:  192.168.2.25
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 1======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  2
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr:  192.168.1.25
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 2======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  5
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr:  192.168.2.24
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 3======
trtype:  tcp
```

```
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  1
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr:  192.168.1.24
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 4======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.
Unidirectional_DHCP_NONE_1_3
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 5======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.
Unidirectional_DHCP_NONE_1_4
traddr:  192.168.1.25
eflags:  none
sectype: none
=====Discovery Log Entry 6======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  5
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.
```

```
Unidirectional_DHCP_NONE_1_5
traddr:  192.168.2.24
eflags:  none
sectype: none
=====Discovery Log Entry 7======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.
Unidirectional_DHCP_2_2
traddr:  192.168.1.24
eflags:  none
sectype: none
=====Discovery Log Entry 8======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.
Unidirectional_DHCP_2_3
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 9======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.
Unidirectional_DHCP_2_5
traddr:  192.168.1.25
eflags:  none
sectype: none
=====Discovery Log Entry 10======
trtype:  tcp
adrfam:  ipv4
```

```
subtype: nvme subsystem
treq:    not specified
portid:  5
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.
Bidirectional_DHCP_2_2
traddr:  192.168.2.24
eflags:  none
sectype: none
=====Discovery Log Entry 11======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.
Bidirectional_DHCP_2_3
traddr:  192.168.1.24
eflags:  none
sectype: none
=====Discovery Log Entry 12======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.
Bidirectional_DHCP_2_3
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 13======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.
Bidirectional_DHCP_NONE_2_4
```

```
traddr:  192.168.1.25
eflags:  none
sectype: none
=====Discovery Log Entry 14======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  5
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.
Bidirectional_DHCP_NONE_2_5
traddr:  192.168.2.24
eflags:  none
sectype: none
=====Discovery Log Entry 15======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.
Bidirectional_DHCP_NONE_2_6
traddr:  192.168.1.24
eflags:  none
sectype: none
=====Discovery Log Entry 16======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.
Bidirectional_DHCP_NONE_2_7
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 17======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
```

```
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.
Bidirectional_DHCP_NONE_2_8
traddr:  192.168.1.25
eflags:  none
sectype: none
=====Discovery Log Entry 18======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  5
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_2
traddr:  192.168.2.24
eflags:  none
sectype: none
=====Discovery Log Entry 19======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.
Bidirectional_DHCP_NONE_2_9
traddr:  192.168.1.24
eflags:  none
sectype: none
```

2. Verify that the other NVMe/TCP initiator-target LIF combinations can successfully fetch discovery log page data:

```
nvme discover -t tcp -w host-traddr -a traddr
```

Show example

```
nvme discover -t tcp -w 192.168.1.31 -a 192.168.1.24
nvme discover -t tcp -w 192.168.2.31 -a 192.168.2.24
nvme discover -t tcp -w 192.168.1.31 -a 192.168.1.25
nvme discover -t tcp -w 192.168.2.31 -a 192.168.2.25
```

3. Run the `nvme connect-all` command across all the supported NVMe/TCP initiator-target LIFs across the nodes:

```
nvme connect-all -t tcp -w host-traddr -a traddr
```

Show example

```
nvme    connect-all -t  tcp -w  192.168.1.31    -a  192.168.1.24
nvme    connect-all -t  tcp -w  192.168.2.31    -a  192.168.2.24
nvme    connect-all -t  tcp -w  192.168.1.31    -a  192.168.1.25
nvme    connect-all -t  tcp -w  192.168.2.31    -a  192.168.2.25
```

Beginning with Rocky Linux 9.4, the setting for the NVMe/TCP `ctrl_loss_tmo timeout` is automatically set to "off". As a result:

- There are no limits on the number of retries (indefinite retry).
- You don't need to manually configure a specific `ctrl_loss_tmo timeout` duration when using the `nvme connect` or `nvme connect-all` commands (option -l).
- The NVMe/TCP controllers don't experience timeouts in the event of a path failure and remain connected indefinitely.

Step 7: Validate NVMe-oF

Verify that the in-kernel NVMe multipath status, ANA status, and ONTAP namespaces are correct for the NVMe-oF configuration.

Steps

1. Verify that the in-kernel NVMe multipath is enabled:

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

You should see the following output:

```
Y
```

2. Verify that the appropriate NVMe-oF settings (such as, model set to NetApp ONTAP Controller and load balancing iopolicy set to round-robin) for the respective ONTAP namespaces correctly reflect on the host:

   a. Display the subsystems:

   ```
   cat /sys/class/nvme-subsystem/nvme-subsys*/model
   ```

   You should see the following output:

   ```
   NetApp ONTAP Controller
   NetApp ONTAP Controller
   ```

   b. Display the policy:

   ```
   cat /sys/class/nvme-subsystem/nvme-subsys*/iopolicy
   ```

   You should see the following output:

   ```
   round-robin
   round-robin
   ```

3. Verify that the namespaces are created and correctly discovered on the host:

   ```
   nvme list
   ```

   Show example

   ```
   Node         SN                   Model
   ---------------------------------------------------------
   /dev/nvme4n1 81Ix2BVuekWcAAAAAAB   NetApp ONTAP Controller


   Namespace Usage    Format             FW             Rev
   -----------------------------------------------------------
   1                 21.47 GB / 21.47 GB   4 KiB + 0 B   FFFFFFFF
   ```

4. Verify that the controller state of each path is live and has the correct ANA status:

NVMe/FC

```
nvme list-subsys /dev/nvme4n5
```

Show example

```
nvme-subsys4 - NQN=nqn.1992-
08.com.netapp:sn.3a5d31f5502c11ef9f50d039eab6cb6d:subsystem.nvme
_1
               hostnqn=nqn.2014-08.org.nvmexpress:uuid:e6dade64-
216d-
11ec-b7bb-7ed30a5482c3
iopolicy=round-robin\
+- nvme1 fc traddr=nn-0x2082d039eaa7dfc8:pn-
0x2088d039eaa7dfc8,host_traddr=nn-0x20000024ff752e6d:pn-
0x21000024ff752e6d live optimized
+- nvme12 fc traddr=nn-0x2082d039eaa7dfc8:pn-
0x208ad039eaa7dfc8,host_traddr=nn-0x20000024ff752e6d:pn-
0x21000024ff752e6d live non-optimized
+- nvme10 fc traddr=nn-0x2082d039eaa7dfc8:pn-
0x2087d039eaa7dfc8,host_traddr=nn-0x20000024ff752e6c:pn-
0x21000024ff752e6c live non-optimized
+- nvme3 fc traddr=nn-0x2082d039eaa7dfc8:pn-
0x2083d039eaa7dfc8,host_traddr=nn-0x20000024ff752e6c:pn-
0x21000024ff752e6c live optimized
```

NVMe/TCP

```
nvme list-subsys /dev/nvme1n1
```

Show example

```
nvme-subsys5 - NQN=nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme
_tcp_3
hostnqn=nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0035-5910-804b-
b5c04f444d33
iopolicy=round-robin
\
+- nvme13 tcp
traddr=192.168.2.25,trsvcid=4420,host_traddr=192.168.2.31,
src_addr=192.168.2.31 live optimized
+- nvme14 tcp
traddr=192.168.2.24,trsvcid=4420,host_traddr=192.168.2.31,
src_addr=192.168.2.31 live non-optimized
+- nvme5 tcp
traddr=192.168.1.25,trsvcid=4420,host_traddr=192.168.1.31,
src_addr=192.168.1.31 live optimized
+- nvme6 tcp
traddr=192.168.1.24,trsvcid=4420,host_traddr=192.168.1.31,
src_addr=192.168.1.31 live non-optimized
```

5. Verify that the NetApp plug-in displays the correct values for each ONTAP namespace device:

Column

```
nvme netapp ontapdevices -o column
```

Show example

```
Device        Vserver   Namespace Path
----------------------- ------------------------------
/dev/nvme1n1     linux_tcnvme_iscsi
/vol/tcpnvme_1_0_0/tcpnvme_ns

NSID       UUID                                   Size
------------------------------------------------------------
1    5f7f630d-8ea5-407f-a490-484b95b15dd6   21.47GB
```

JSON

```
nvme netapp ontapdevices -o json
```

Show example

```
{
  "ONTAPdevices":[
    {
      "Device":"/dev/nvme1n1",
      "Vserver":"linux_tcnvme_iscsi",
      "Namespace_Path":"/vol/tcpnvme_1_0_0/tcpnvme_ns",
      "NSID":1,
      "UUID":"5f7f630d-8ea5-407f-a490-484b95b15dd6",
      "Size":"21.47GB",
      "LBA_Data_Size":4096,
      "Namespace_Size":5242880
    },
]
}
```

Step 8: Set up secure in-band authentication

Beginning with ONTAP 9.12.1, secure in-band authentication is supported over NVMe/TCP between a Rocky Linux 9.6 host and an ONTAP controller.

Each host or controller must be associated with a `DH-HMAC-CHAP` key to set up secure authentication. A `DH-HMAC-CHAP` key is a combination of the NQN of the NVMe host or controller and an authentication secret configured by the administrator. To authenticate its peer, an NVMe host or controller must recognize the key associated with the peer.

Set up secure in-band authentication using the CLI or a config JSON file. If you need to specify different dhchap keys for different subsystems, you must use a config JSON file.

CLI

Set up secure in-band authentication using the CLI.

Steps

1. Obtain the host NQN:

```
cat /etc/nvme/hostnqn
```

2. Generate the dhchap key for the Rocky Linux 9.6 host.

   The following output describes the `gen-dhchap-key` command paramters:

```
nvme gen-dhchap-key -s optional_secret -l key_length {32|48|64} -m
HMAC_function {0|1|2|3} -n host_nqn
•   -s secret key in hexadecimal characters to be used to initialize
the host key
•   -l length of the resulting key in bytes
•   -m HMAC function to use for key transformation
0 = none, 1- SHA-256, 2 = SHA-384, 3=SHA-512
•   -n host NQN to use for key transformation
```

   In the following example, a random dhchap key with HMAC set to 3 (SHA-512) is generated.

```
nvme gen-dhchap-key -m 3 -n nqn.2014-
08.org.nvmexpress:uuid:e6dade64-216d-11ec-b7bb-7ed30a5482c3
DHHC-
1:03:wSpuuKbBHTzC0W9JZxMBsYd9JFV8Si9aDh22k2BR/4m852vH7KGlrJeMpzhmyjD
WOo0PJJM6yZsTeEpGkDHMHQ255+g=:
```

3. On the ONTAP controller, add the host and specify both dhchap keys:

```
vserver nvme subsystem host add -vserver <svm_name> -subsystem
<subsystem> -host-nqn <host_nqn> -dhchap-host-secret
<authentication_host_secret> -dhchap-controller-secret
<authentication_controller_secret> -dhchap-hash-function {sha-
256|sha-512} -dhchap-group {none|2048-bit|3072-bit|4096-bit|6144-
bit|8192-bit}
```

4. A host supports two types of authentication methods, unidirectional and bidirectional. On the host, connect to the ONTAP controller and specify dhchap keys based on the chosen authentication method:

```
nvme connect -t tcp -w <host-traddr> -a <tr-addr> -n <host_nqn> -S
<authentication_host_secret> -C <authentication_controller_secret>
```

5. Validate the `nvme connect authentication` command by verifying the host and controller dhchap keys:

   a. Verify the host dhchap keys:

   ```
   cat /sys/class/nvme-subsystem/<nvme-subsysX>/nvme*/dhchap_secret
   ```

   Show example output for a unidirectional configuration

   ```
   cat /sys/class/nvme-subsystem/nvme-subsys1/nvme*/dhchap_secret
   DHHC-1:01:CNxTYq73T9vJk0JpOfDBZrhDCqpWBN4XVZI5WxwPgDUieHAi:
   DHHC-1:01:CNxTYq73T9vJk0JpOfDBZrhDCqpWBN4XVZI5WxwPgDUieHAi:
   DHHC-1:01:CNxTYq73T9vJk0JpOfDBZrhDCqpWBN4XVZI5WxwPgDUieHAi:
   DHHC-1:01:CNxTYq73T9vJk0JpOfDBZrhDCqpWBN4XVZI5WxwPgDUieHAi:
   ```

   b. Verify the controller dhchap keys:

   ```
   cat /sys/class/nvme-subsystem/<nvme-
   subsysX>/nvme*/dhchap_ctrl_secret
   ```

   Show example output for a bidirectional configuration

   ```
   cat /sys/class/nvme-subsystem/nvme-
   subsys6/nvme*/dhchap_ctrl_secret
   DHHC-
   1:03:wSpuuKbBHTzC0W9JZxMBsYd9JFV8Si9aDh22k2BR/4m852vH7KGlrJeMp
   zhmyjDWOo0PJJM6yZsTeEpGkDHMHQ255+g=:
   DHHC-
   1:03:wSpuuKbBHTzC0W9JZxMBsYd9JFV8Si9aDh22k2BR/4m852vH7KGlrJeMp
   zhmyjDWOo0PJJM6yZsTeEpGkDHMHQ255+g=:
   DHHC-
   1:03:wSpuuKbBHTzC0W9JZxMBsYd9JFV8Si9aDh22k2BR/4m852vH7KGlrJeMp
   zhmyjDWOo0PJJM6yZsTeEpGkDHMHQ255+g=:
   DHHC-
   1:03:wSpuuKbBHTzC0W9JZxMBsYd9JFV8Si9aDh22k2BR/4m852vH7KGlrJeMp
   zhmyjDWOo0PJJM6yZsTeEpGkDHMHQ255+g=:
   ```

JSON file

When multiple NVMe subsystems are available on the ONTAP controller configuration, you can use the `/etc/nvme/config.json` file with the `nvme connect-all` command.

Use the `-o` option to generate the JSON file. See the NVMe connect-all manual pages for more syntax options.

Steps

1. Configure the JSON file.

   > In the following example, `dhchap_key` corresponds to `dhchap_secret` and `dhchap_ctrl_key` corresponds to `dhchap_ctrl_secret`.

Show example

```
cat /etc/nvme/config.json
[
{
  "hostnqn":"nqn.2014-08.org.nvmexpress:uuid:9796c1ec-0d34-11eb-
  b6b2-3a68dd3bab57",
  "hostid":"b033cd4fd6db4724adb48655bfb55448",
  "dhchap_key":" DHHC-
1:01:CNxTYq73T9vJk0JpOfDBZrhDCqpWBN4XVZI5WxwPgDUieHAi:"
},
{
  "hostnqn":"nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0035-5910-
  804b-b5c04f444d33",
  "subsystems":[
        {
          "nqn":"nqn.1992-

08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.bidi
          r_DHCP",
          "ports":[
              {
                  "transport":"tcp",
                    "traddr":" 192.168.1.24 ",
                  "host_traddr":" 192.168.1.31 ",
                  "trsvcid":"4420",
                  "dhchap_ctrl_key":"DHHC-
                  1:03:
wSpuuKbBHTzC0W9JZxMBsYd9JFV8Si9aDh22k2BR/4m852vH7KGlrJeMpzhmyjDW
Oo0PJJM6yZsTeEpGkDHMHQ255+g=:"
              },
              {
                  "transport":"tcp",
                  "traddr":" 192.168.1.25 ",
                  "host_traddr":" 192.168.1.31",
                  "trsvcid":"4420",
                  "dhchap_ctrl_key":"DHHC-
                  1:03:
wSpuuKbBHTzC0W9JZxMBsYd9JFV8Si9aDh22k2BR/4m852vH7KGlrJeMpzhmyjDW
Oo0PJJM6yZsTeEpGkDHMHQ255+g=:"
              },
              {
                  "transport":"tcp",
                 "traddr":" 192.168.2.24 ",
                  "host_traddr":" 192.168.2.31",
                  "trsvcid":"4420",
```

```
                  "dhchap_ctrl_key":"DHHC-
                  1:03:
wSpuuKbBHTzC0W9JZxMBsYd9JFV8Si9aDh22k2BR/4m852vH7KGlrJeMpzhmyjDW
Oo0PJJM6yZsTeEpGkDHMHQ255+g=:"
              },
              {
                  "transport":"tcp",
                  "traddr":" 192.168.2.25 ",
                    "host_traddr":" 192.168.2.31",
                  "trsvcid":"4420",
                  "dhchap_ctrl_key":"DHHC-
                  1:03:
wSpuuKbBHTzC0W9JZxMBsYd9JFV8Si9aDh22k2BR/4m852vH7KGlrJeMpzhmyjDW
Oo0PJJM6yZsTeEpGkDHMHQ255+g=:"
              }
          ]
      }
  ]
}
]
```

2. Connect to the ONTAP controller using the config JSON file:

```
nvme connect-all -J /etc/nvme/config.json
```

Show example

```
already connected to hostnqn=nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0035-5910-804b-
b5c04f444d33,nqn=nqn.1992-
08.com.netapp:sn.8dde3be2cc7c11efb777d039eab6cb6d:subsystem.
bidi
r_DHCP,transport=tcp,traddr=192.168.1.25,trsvcid=4420
already connected to hostnqn=nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0035-5910-804b-
b5c04f444d33,nqn=nqn.1992-
08.com.netapp:sn.8dde3be2cc7c11efb777d039eab6cb6d:subsystem.
bidi
r_DHCP,transport=tcp,traddr=192.168.2.25,trsvcid=4420
already connected to hostnqn=nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0035-5910-804b-
b5c04f444d33,nqn=nqn.1992-
08.com.netapp:sn.8dde3be2cc7c11efb777d039eab6cb6d:subsystem.
bidi
r_DHCP,transport=tcp,traddr=192.168.1.24,trsvcid=4420
already connected to hostnqn=nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0035-5910-804b-
b5c04f444d33,nqn=nqn.1992-
08.com.netapp:sn.8dde3be2cc7c11efb777d039eab6cb6d:subsystem.
bidi
r_DHCP,transport=tcp,traddr=192.168.2.24,trsvcid=4420
```

3. Verify that the dhchap secrets have been enabled for the respective controllers for each subsystem:
   a. Verify the host dhchap keys:

      ```
      cat /sys/class/nvme-subsystem/nvme-subsys0/nvme0/dhchap_secret
      ```

      The following example shows a dhchap key:

      ```
      DHHC-1:01:CNxTYq73T9vJk0JpOfDBZrhDCqpWBN4XVZI5WxwPgDUieHAi:
      ```

   b. Verify the controller dhchap keys:

      ```
      cat /sys/class/nvme-subsystem/nvme-
      subsys0/nvme0/dhchap_ctrl_secret
      ```

      You should see an output similar to the following example:

```
DHHC-
1:03:wSpuuKbBHTzC0W9JZxMBsYd9JFV8Si9aDh22k2BR/4m852vH7KGlrJeMpzhm
yjDWOo0PJJM6yZsTeEpGkDHMHQ255+g=:
```

Step 9: Review the known issues

There are no known issues.

Configure Rocky Linux 9.5 with NVMe-oF for ONTAP storage

Rocky Linux 9.5 hosts support the NVMe/FC and NVMe/TCP protocols with Asymmetric Namespace Access (ANA). ANA is equivalent to asymmetric logical unit access (ALUA) multipathing in iSCSI and FCP environments and is implemented using the in-kernel NVMe multipath feature.

For additional details on supported configurations, see the Interoperability Matrix Tool.

About this task

You can use the following support and features with the NVMe-oF host configuration for Rocky Linux 9.5. You should also review the known limitations before starting the configuration process.

- Support available:
  - Support for NVMe over TCP (NVMe/TCP) in addition to NVMe over Fibre Channel (NVMe/FC). The NetApp plug-in in the native `nvme-cli` package displays ONTAP details for both NVMe/FC and NVMe/TCP namespaces.
  - Running NVMe and SCSI traffic on the same host. For example, you can configure dm-multipath for SCSI mpath devices on SCSI LUNs and use NVMe multipath to configure NVMe-oF namespace devices on the host.
- Features available:
  - Beginning with ONTAP 9.12.1, you can use secure in-band authentication for NVMe-oF with Rocky Linux 9.5.
  - Rocky Linux 9.5 enables in-kernel NVMe multipath for NVMe namespaces by default, removing the need for explicit settings.
- Known limitations:
  - There are no known limitations.

Step 1: Optionally, enable SAN booting

You can configure your host to use SAN booting to simplify deployment and improve scalability.

Before you begin

Use the Interoperability Matrix Tool to verify that your Linux OS, host bus adapter (HBA), HBA firmware, HBA boot BIOS, and ONTAP version support SAN booting.

Steps

1. Create a SAN boot namespace and map it to the host.

2. Enable SAN booting in the server BIOS for the ports to which the SAN boot namespace is mapped.

   For information on how to enable the HBA BIOS, see your vendor-specific documentation.

3. Verify that the configuration was successful by rebooting the host and verifying that the OS is up and running.

Step 2: Validate software versions

Use the following procedure to validate the minimum supported Rocky Linux 9.5 software versions.

Steps

1. Install Rocky Linux 9.5 on the server. After the installation is complete, verify that you are running the specified Rocky Linux 9.5 kernel:

   ```
   uname -r
   ```

   The following example shows a Rocky Linux kernel version:

   ```
   5.14.0-570.12.1.el9_6.x86_64
   ```

2. Install the `nvme-cli` package:

   ```
   rpm -qa|grep nvme-cli
   ```

   The following example shows an nvme-cli package version:

   ```
   nvme-cli-2.11-5.el9.x86_64
   ```

3. Install the `libnvme` package:

   ```
   rpm -qa|grep libnvme
   ```

   The following example shows an `libnvme` package version:

   ```
   libnvme-1.11.1-1.el9.x86_64
   ```

4. On the Rocky Linux host, check the hostnqn string at `/etc/nvme/hostnqn`:

   ```
   cat /etc/nvme/hostnqn
   ```

The following example shows an `hostnqn` version:

```
nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
```

5. Verify that the `hostnqn` string matches the `hostnqn` string for the corresponding subsystem on the ONTAP array:

```
::> vserver nvme subsystem host show -vserver vs_coexistence_LPE36002
```

Show example

```
Vserver Subsystem Priority  Host NQN
------- --------- --------
------------------------------------------------
vs_coexistence_LPE36002
        nvme
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
        nvme_1
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
        nvme_2
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
        nvme_3
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
4 entries were displayed.
```

If the `hostnqn` strings do not match, use the `vserver modify` command to update the `hostnqn` string on your corresponding ONTAP array subsystem to match the `hostnqn` string from `/etc/nvme/hostnqn` on the host.

Step 3: Configure NVMe/FC

You can configure NVMe/FC with Broadcom/Emulex FC or Marvell/Qlogic FC adapters. You also need to manually discover the NVMe/TCP subsystems and namespaces.

Broadcom/Emulex

Configure NVMe/FC for a Broadcom/Emulex adapter.

Steps

1. Verify that you are using the supported adapter model:

   a. Display the model names:

   ```
   cat /sys/class/scsi_host/host*/modelname
   ```

   You should see the following output:

   ```
   LPe36002-M64
   LPe36002-M64
   ```

   b. Display the model descriptions:

   ```
   cat /sys/class/scsi_host/host*/modeldesc
   ```

   You should see an output similar to the following example:

   ```
   Emulex LightPulse LPe36002-M64 2-Port 64Gb Fibre Channel Adapter
   Emulex LightPulse LPe36002-M64 2-Port 64Gb Fibre Channel Adapter
   ```

2. Verify that you are using the recommended Broadcom `lpfc` firmware and inbox driver:

   a. Display the firmware version:

   ```
   cat /sys/class/scsi_host/host*/fwrev
   ```

   The following example shows firmware versions:

   ```
   14.4.317.10, sli-4:6:d
   14.4.317.10, sli-4:6:d
   ```

   b. Display the inbox driver version:

   ```
   cat /sys/module/lpfc/version`
   ```

   The following example shows a driver version:

```
0:14.4.0.2
```

For the current list of supported adapter driver and firmware versions, see the Interoperability Matrix Tool.

3. Verify that the expected output of `lpfc_enable_fc4_type` is set to `3`:

```
cat /sys/module/lpfc/parameters/lpfc_enable_fc4_type
```

4. Verify that you can view your initiator ports:

```
cat /sys/class/fc_host/host*/port_name
```

The following example shows port identities:

```
0x100000109bf044b1
0x100000109bf044b2
```

5. Verify that your initiator ports are online:

```
cat /sys/class/fc_host/host*/port_state
```

You should see the following output:

```
Online
Online
```

6. Verify that the NVMe/FC initiator ports are enabled and that the target ports are visible:

```
cat /sys/class/scsi_host/host*/nvme_info
```

Show example

```
NVME Initiator Enabled
XRI Dist lpfc2 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc2 WWPN x100000109bf044b1 WWNN x200000109bf044b1
DID x022a00 ONLINE
NVME RPORT       WWPN x202fd039eaa7dfc8 WWNN x202cd039eaa7dfc8
DID x021310 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x202dd039eaa7dfc8 WWNN x202cd039eaa7dfc8
DID x020b10 TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 0000000810 Cmpl 0000000810 Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000007b098f07 Issue 000000007aee27c4 OutIO
fffffffffe498bd
        abort 000013b4 noxri 00000000 nondlp 00000058 qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 000013b4 Err 00021443

NVME Initiator Enabled
XRI Dist lpfc3 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc3 WWPN x100000109bf044b2 WWNN x200000109bf044b2
DID x021b00 ONLINE
NVME RPORT       WWPN x2033d039eaa7dfc8 WWNN x202cd039eaa7dfc8
DID x020110 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2032d039eaa7dfc8 WWNN x202cd039eaa7dfc8
DID x022910 TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 0000000840 Cmpl 0000000840 Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000007afd4434 Issue 000000007ae31b83 OutIO
fffffffffe5d74f
        abort 000014a5 noxri 00000000 nondlp 0000006a qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 000014a5 Err 0002149a
```

Marvell/QLogic

Configure NVMe/FC for a Marvell/QLogic adapter.

> ⓘ The native inbox qla2xxx driver included in the Rocky Linux kernel has the latest fixes. These fixes are essential for ONTAP support.

Steps

1. Verify that you are running the supported adapter driver and firmware versions:

```
cat /sys/class/fc_host/host*/symbolic_name
```

The follow example shows driver and firmware versions:

```
QLE2742 FW:v9.14.00 DVR:v10.02.09.200-k
QLE2742 FW:v9.14.00 DVR:v10.02.09.200-k
```

2. Verify that `ql2xnvmeenable` is set. This enables the Marvell adapter to function as an NVMe/FC initiator:

```
cat /sys/module/qla2xxx/parameters/ql2xnvmeenable
```

The expected output is 1.

Step 4: Optionally, enable 1MB I/O

You can enable I/O requests of size 1MB for NVMe/FC configured with a Broadcom adapter. ONTAP reports a Max Data Transfer Size (MDTS) of 8 in the Identify Controller data. This means the maximum I/O request size can be up to 1MB. To issue I/O requests of size 1MB, you need to increase the lpfc value of the `lpfc_sg_seg_cnt` parameter to 256 from the default value of 64.

ⓘ These steps don't apply to Qlogic NVMe/FC hosts.

Steps

1. Set the `lpfc_sg_seg_cnt` parameter to 256:

```
cat /etc/modprobe.d/lpfc.conf
```

```
options lpfc lpfc_sg_seg_cnt=256
```

2. Run the `dracut -f` command, and reboot the host.
3. Verify that the value for `lpfc_sg_seg_cnt` is 256:

```
cat /sys/module/lpfc/parameters/lpfc_sg_seg_cnt
```

Step 5: Configure NVMe/TCP

The NVMe/TCP protocol doesn’t support the auto-connect operation. Instead, you can discover the NVMe/TCP subsystems and namespaces by performing the NVMe/TCP connect or connect-all operations manually.

Steps

1. Verify that the initiator port can fetch the discovery log page data across the supported NVMe/TCP LIFs:

   ```
   nvme discover -t tcp -w host-traddr -a traddr
   ```

Show example

```
nvme discover -t tcp -w 192.168.1.31 -a 192.168.1.24
Discovery Log Number of Records 20, Generation counter 25
=====Discovery Log Entry 0======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  4
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr:  192.168.2.25
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 1======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  2
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr:  192.168.1.25
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 2======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  5
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr:  192.168.2.24
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 3======
trtype:  tcp
adrfam:  ipv4
```

```
subtype: current discovery subsystem
treq:    not specified
portid:  1
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr:  192.168.1.24
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 4======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_1
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 5======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_1
traddr:  192.168.1.25
eflags:  none
sectype: none
=====Discovery Log Entry 6======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  5
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_1
```

```
traddr:  192.168.2.24
eflags:  none
sectype: none
=====Discovery Log Entry 7======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_1
traddr:  192.168.1.24
eflags:  none
sectype: none
=====Discovery Log Entry 8======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_4
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 9======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_4
traddr:  192.168.1.25
eflags:  none
sectype: none
=====Discovery Log Entry 10======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
```

```
treq:    not specified
portid:  5
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_4
traddr:  192.168.2.24
eflags:  none
sectype: none
=====Discovery Log Entry 11======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_4
traddr:  192.168.1.24
eflags:  none
sectype: none
=====Discovery Log Entry 12======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_3
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 13======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_3
traddr:  192.168.1.25
```

```
eflags:  none
sectype: none
=====Discovery Log Entry 14======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  5
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_3
traddr:  192.168.2.24
eflags:  none
sectype: none
=====Discovery Log Entry 15======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_3
traddr:  192.168.1.24
eflags:  none
sectype: none
=====Discovery Log Entry 16======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_2
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 17======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
```

```
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_2
traddr:  192.168.1.25
eflags:  none
sectype: none
=====Discovery Log Entry 18======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  5
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_2
traddr:  192.168.2.24
eflags:  none
sectype: none
=====Discovery Log Entry 19======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_2
traddr:  192.168.1.24
eflags:  none
sectype: none
```

2. Verify that the other NVMe/TCP initiator-target LIF combinations are able to successfully fetch discovery log page data:

```
nvme discover -t tcp -w host-traddr -a traddr
```

Show example

```
nvme discover -t tcp -w 192.168.1.31 -a 192.168.1.24
nvme discover -t tcp -w 192.168.2.31 -a 192.168.2.24
nvme discover -t tcp -w 192.168.1.31 -a 192.168.1.25
nvme discover -t tcp -w 192.168.2.31 -a 192.168.2.25
```

3. Run the `nvme connect-all` command across all the supported NVMe/TCP initiator-target LIFs across the nodes:

```
nvme connect-all -t tcp -w host-traddr -a traddr
```

Show example

```
nvme    connect-all -t  tcp -w  192.168.1.31    -a  192.168.1.24
nvme    connect-all -t  tcp -w  192.168.2.31    -a  192.168.2.24
nvme    connect-all -t  tcp -w  192.168.1.31    -a  192.168.1.25
nvme    connect-all -t  tcp -w  192.168.2.31    -a  192.168.2.25
```

Beginning with Rocky Linux 9.4, the setting for the NVMe/TCP `ctrl_loss_tmo timeout` is automatically set to "off". As a result:

- There are no limits on the number of retries (indefinite retry).
- You don’t need to manually configure a specific `ctrl_loss_tmo timeout` duration when using the `nvme connect` or `nvme connect-all` commands (option -l).
- The NVMe/TCP controllers don’t experience timeouts in the event of a path failure and remain connected indefinitely.

Step 6: Validate NVMe-oF

Verify that the in-kernel NVMe multipath status, ANA status, and ONTAP namespaces are correct for the NVMe-oF configuration.

Steps

1. Verify that the in-kernel NVMe multipath is enabled:

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

You should see the following output:

```
Y
```

2. Verify that the appropriate NVMe-oF settings (such as, model set to NetApp ONTAP Controller and load balancing iopolicy set to round-robin) for the respective ONTAP namespaces correctly reflect on the host:

   a. Display the subsystems:

   ```
   cat /sys/class/nvme-subsystem/nvme-subsys*/model
   ```

   You should see the following output:

   ```
   NetApp ONTAP Controller
   NetApp ONTAP Controller
   ```

   b. Display the policy:

   ```
   cat /sys/class/nvme-subsystem/nvme-subsys*/iopolicy
   ```

   You should see the following output:

   ```
   round-robin
   round-robin
   ```

3. Verify that the namespaces are created and correctly discovered on the host:

   ```
   nvme list
   ```

   Show example

   ```
   Node         SN                   Model
   ---------------------------------------------------------
   /dev/nvme4n1 81Ix2BVuekWcAAAAAAAB   NetApp ONTAP Controller


   Namespace Usage    Format             FW             Rev
   -----------------------------------------------------------
   1                21.47 GB / 21.47 GB   4 KiB + 0 B   FFFFFFFF
   ```

4. Verify that the controller state of each path is live and has the correct ANA status:

NVMe/FC

```
nvme list-subsys /dev/nvme4n5
```

Show example

```
nvme-subsys4 - NQN=nqn.1992-
08.com.netapp:sn.3a5d31f5502c11ef9f50d039eab6cb6d:subsystem.nvme
_1
               hostnqn=nqn.2014-08.org.nvmexpress:uuid:e6dade64-
216d-
11ec-b7bb-7ed30a5482c3
iopolicy=round-robin\
+- nvme1 fc traddr=nn-0x2082d039eaa7dfc8:pn-
0x2088d039eaa7dfc8,host_traddr=nn-0x20000024ff752e6d:pn-
0x21000024ff752e6d live optimized
+- nvme12 fc traddr=nn-0x2082d039eaa7dfc8:pn-
0x208ad039eaa7dfc8,host_traddr=nn-0x20000024ff752e6d:pn-
0x21000024ff752e6d live non-optimized
+- nvme10 fc traddr=nn-0x2082d039eaa7dfc8:pn-
0x2087d039eaa7dfc8,host_traddr=nn-0x20000024ff752e6c:pn-
0x21000024ff752e6c live non-optimized
+- nvme3 fc traddr=nn-0x2082d039eaa7dfc8:pn-
0x2083d039eaa7dfc8,host_traddr=nn-0x20000024ff752e6c:pn-
0x21000024ff752e6c live optimized
```

NVMe/TCP

```
nvme list-subsys /dev/nvme1n1
```

Show example

```
nvme-subsys5 - NQN=nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme
_tcp_3
hostnqn=nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0035-5910-804b-
b5c04f444d33
iopolicy=round-robin
\
+- nvme13 tcp
traddr=192.168.2.25,trsvcid=4420,host_traddr=192.168.2.31,
src_addr=192.168.2.31 live optimized
+- nvme14 tcp
traddr=192.168.2.24,trsvcid=4420,host_traddr=192.168.2.31,
src_addr=192.168.2.31 live non-optimized
+- nvme5 tcp
traddr=192.168.1.25,trsvcid=4420,host_traddr=192.168.1.31,
src_addr=192.168.1.31 live optimized
+- nvme6 tcp
traddr=192.168.1.24,trsvcid=4420,host_traddr=192.168.1.31,
src_addr=192.168.1.31 live non-optimized
```

5. Verify that the NetApp plug-in displays the correct values for each ONTAP namespace device:

Column

```
nvme netapp ontapdevices -o column
```

Show example

```
Device        Vserver   Namespace Path
----------------------- ------------------------------
/dev/nvme1n1     linux_tcnvme_iscsi
/vol/tcpnvme_1_0_0/tcpnvme_ns

NSID       UUID                                   Size
------------------------------------------------------------
1    5f7f630d-8ea5-407f-a490-484b95b15dd6   21.47GB
```

JSON

```
nvme netapp ontapdevices -o json
```

Show example

```
{
  "ONTAPdevices":[
    {
      "Device":"/dev/nvme1n1",
      "Vserver":"linux_tcnvme_iscsi",
      "Namespace_Path":"/vol/tcpnvme_1_0_0/tcpnvme_ns",
      "NSID":1,
      "UUID":"5f7f630d-8ea5-407f-a490-484b95b15dd6",
      "Size":"21.47GB",
      "LBA_Data_Size":4096,
      "Namespace_Size":5242880
    },
]
}
```

Step 7: Set up secure in-band authentication

Beginning with ONTAP 9.12.1, secure in-band authentication is supported over NVMe/TCP and NVMe/FC between the host and the ONTAP controller.

To set up secure authentication, each host or controller must be associated with a `DH-HMAC-CHAP` key, which is a combination of the NQN of the NVMe host or controller and an authentication secret configured by the administrator. To authenticate its peer, an NVMe host or controller must recognize the key associated with the peer.

You can set up secure in-band authentication using the CLI or a config JSON file. If you need to specify different dhchap keys for different subsystems, you must use a config JSON file.

CLI

Set up secure in-band authentication using the CLI.

Steps

1. Obtain the host NQN:

```
cat /etc/nvme/hostnqn
```

2. Generate the dhchap key for the host.

The following output describes the `gen-dhchap-key` command paramters:

```
nvme gen-dhchap-key -s optional_secret -l key_length {32|48|64} -m
HMAC_function {0|1|2|3} -n host_nqn
•   -s secret key in hexadecimal characters to be used to initialize
the host key
•   -l length of the resulting key in bytes
•   -m HMAC function to use for key transformation
0 = none, 1- SHA-256, 2 = SHA-384, 3=SHA-512
•   -n host NQN to use for key transformation
```

In the following example, a random dhchap key with HMAC set to 3 (SHA-512) is generated.

```
nvme gen-dhchap-key -m 3 -n nqn.2014-
08.org.nvmexpress:uuid:e6dade64-216d-11ec-b7bb-7ed30a5482c3
DHHC-
1:03:1CFivw9ccz58gAcOUJrM7Vs98hd2ZHSr+iw+Amg6xZPl5D2Yk+HDTZiUAg1iGgx
TYqnxukqvYedA55Bw3wtz6sJNpR4=:
```

3. On the ONTAP controller, add the host and specify both dhchap keys:

```
vserver nvme subsystem host add -vserver <svm_name> -subsystem
<subsystem> -host-nqn <host_nqn> -dhchap-host-secret
<authentication_host_secret> -dhchap-controller-secret
<authentication_controller_secret> -dhchap-hash-function {sha-
256|sha-512} -dhchap-group {none|2048-bit|3072-bit|4096-bit|6144-
bit|8192-bit}
```

4. A host supports two types of authentication methods, unidirectional and bidirectional. On the host, connect to the ONTAP controller and specify dhchap keys based on the chosen authentication method:

```
nvme connect -t tcp -w <host-traddr> -a <tr-addr> -n <host_nqn> -S
<authentication_host_secret> -C <authentication_controller_secret>
```

5. Validate the `nvme connect authentication` command by verifying the host and controller dhchap keys:

   a. Verify the host dhchap keys:

   ```
   cat /sys/class/nvme-subsystem/<nvme-subsysX>/nvme*/dhchap_secret
   ```

   Show example output for a unidirectional configuration

   ```
   cat /sys/class/nvme-subsystem/nvme-subsys1/nvme*/dhchap_secret
   DHHC-1:01:iM63E6cX7G5SOKKOju8gmzM53qywsy+C/YwtzxhIt9ZRz+ky:
   DHHC-1:01:iM63E6cX7G5SOKKOju8gmzM53qywsy+C/YwtzxhIt9ZRz+ky:
   DHHC-1:01:iM63E6cX7G5SOKKOju8gmzM53qywsy+C/YwtzxhIt9ZRz+ky:
   DHHC-1:01:iM63E6cX7G5SOKKOju8gmzM53qywsy+C/YwtzxhIt9ZRz+ky:
   ```

   b. Verify the controller dhchap keys:

   ```
   cat /sys/class/nvme-subsystem/<nvme-
   subsysX>/nvme*/dhchap_ctrl_secret
   ```

   Show example output for a bidirectional configuration

   ```
   cat /sys/class/nvme-subsystem/nvme-
   subsys6/nvme*/dhchap_ctrl_secret
   DHHC-
   1:03:1CFivw9ccz58gAcOUJrM7Vs98hd2ZHSr+iw+Amg6xZPl5D2Yk+HDTZiUA
   g1iGgxTYqnxukqvYedA55Bw3wtz6sJNpR4=:
   DHHC-
   1:03:1CFivw9ccz58gAcOUJrM7Vs98hd2ZHSr+iw+Amg6xZPl5D2Yk+HDTZiUA
   g1iGgxTYqnxukqvYedA55Bw3wtz6sJNpR4=:
   DHHC-
   1:03:1CFivw9ccz58gAcOUJrM7Vs98hd2ZHSr+iw+Amg6xZPl5D2Yk+HDTZiUA
   g1iGgxTYqnxukqvYedA55Bw3wtz6sJNpR4=:
   DHHC-
   1:03:1CFivw9ccz58gAcOUJrM7Vs98hd2ZHSr+iw+Amg6xZPl5D2Yk+HDTZiUA
   g1iGgxTYqnxukqvYedA55Bw3wtz6sJNpR4=:
   ```

JSON file

When multiple NVMe subsystems are available on the ONTAP controller configuration, you can use the `/etc/nvme/config.json` file with the `nvme connect-all` command.

Use the `-o` option to generate the JSON file. See the NVMe connect-all manual pages for more syntax options.

Steps

1. Configure the JSON file:

   > In the following example, `dhchap_key` corresponds to `dhchap_secret` and `dhchap_ctrl_key` corresponds to `dhchap_ctrl_secret`.

Show example

```
cat /etc/nvme/config.json
[
{
  "hostnqn":"nqn.2014-08.org.nvmexpress:uuid:9796c1ec-0d34-11eb-
b6b2-3a68dd3bab57",
  "hostid":"b033cd4fd6db4724adb48655bfb55448",
  "dhchap_key":"DHHC-
1:01:zGlgmRyWbplWfUCPMuaP3mAypX0+GHuSczx5vX4Yod9lMPim:"
},
{
  "hostnqn":"nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0035-5910-
804b-b5c04f444d33",
  "subsystems":[
       {
          "nqn":"nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.bidi
r_DHCP",
          "ports":[
              {
                  "transport":"tcp",
                   "traddr":" 192.168.1.24 ",
                  "host_traddr":" 192.168.1.31 ",
                  "trsvcid":"4420",
                  "dhchap_ctrl_key":"DHHC-
1:03:L52ymUoR32zYvnqZFe5OHhMg4gxD79jIyxSShHansXpVN+WiXE222aVc651
JxGZlQCI863iVOz5dNWvgb+14F4B4bTQ=:"
              },
              {
                  "transport":"tcp",
                  "traddr":" 192.168.1.24 ",
                  "host_traddr":" 192.168.1.31",
                  "trsvcid":"4420",
                  "dhchap_ctrl_key":"DHHC-
1:03:L52ymUoR32zYvnqZFe5OHhMg4gxD79jIyxSShHansXpVN+WiXE222aVc651
JxGZlQCI863iVOz5dNWvgb+14F4B4bTQ=:"
              },
              {
                  "transport":"tcp",
                 "traddr":" 192.168.1.24 ",
                  "host_traddr":" 192.168.1.31",
                  "trsvcid":"4420",
                  "dhchap_ctrl_key":"DHHC-
1:03:L52ymUoR32zYvnqZFe5OHhMg4gxD79jIyxSShHansXpVN+WiXE222aVc651
JxGZlQCI863iVOz5dNWvgb+14F4B4bTQ=:"
```

```
                },
                {
                    "transport":"tcp",
                    "traddr":" 192.168.1.24 ",
                     "host_traddr":" 192.168.1.31",
                    "trsvcid":"4420",
                    "dhchap_ctrl_key":"DHHC-
1:03:L52ymUoR32zYvnqZFe5OHhMg4gxD79jIyxSShHansXpVN+WiXE222aVc651
JxGZlQCI863iVOz5dNWvgb+14F4B4bTQ=:"
                }
            ]
        }
    ]
}
]
```

2. Connect to the ONTAP controller using the config JSON file:

```
nvme connect-all -J /etc/nvme/config.json
```

Show example

```
traddr=192.168.1.24 is already connected
traddr=192.168.1.24 is already connected
traddr=192.168.1.24 is already connected
traddr=192.168.1.24 is already connected
traddr=192.168.1.24 is already connected
traddr=192.168.1.24 is already connected
traddr=192.168.1.25 is already connected
traddr=192.168.1.25 is already connected
traddr=192.168.1.25 is already connected
traddr=192.168.1.25 is already connected
traddr=192.168.1.25 is already connected
traddr=192.168.1.25 is already connected
```

3. Verify that the dhchap secrets have been enabled for the respective controllers for each subsystem:
   a. Verify the host dhchap keys:

```
cat /sys/class/nvme-subsystem/nvme-subsys0/nvme0/dhchap_secret
```

```
DHHC-1:01:zGlgmRyWbplWfUCPMuaP3mAypX0+GHuSczx5vX4Yod9lMPim:
```

b. Verify the controller dhchap keys:

```
cat /sys/class/nvme-subsystem/nvme-
subsys0/nvme0/dhchap_ctrl_secret
```

```
DHHC-
1:03:L52ymUoR32zYvnqZFe5OHhMg4gxD79jIyxSShHansXpVN+WiXE222aVc651J
xGZlQCI863iVOz5dNWvgb+14F4B4bTQ=:
```

Step 8: Review the known issues

There are no known issues.

Configure Rocky Linux 9.4 with NVMe-oF for ONTAP storage

Rocky Linux 9.4 hosts support the NVMe/FC and NVMe/TCP protocols with Asymmetric Namespace Access (ANA). ANA is equivalent to asymmetric logical unit access (ALUA) multipathing in iSCSI and FCP environments and is implemented using the in-kernel NVMe multipath feature.

For additional details on supported configurations, see the Interoperability Matrix Tool.

About this task

You can use the following support and features with the NVMe-oF host configuration for Rocky Linux 9.4. You should also review the known limitations before starting the configuration process.

- Support available:
  - Support for NVMe over TCP (NVMe/TCP) in addition to NVMe over Fibre Channel (NVMe/FC). The NetApp plug-in in the native `nvme-cli` package displays ONTAP details for both NVMe/FC and NVMe/TCP namespaces.
  - Running NVMe and SCSI traffic on the same host. For example, you can configure dm-multipath for SCSI mpath devices on SCSI LUNs and use NVMe multipath to configure NVMe-oF namespace devices on the host.
- Features available:
  - Rocky Linux 9.4 enables in-kernel NVMe multipath for NVMe namespaces by default, removing the need for explicit settings.
- Known limitations:
  - There are no known limitations.

Step 1: Optionally, enable SAN booting

You can configure your host to use SAN booting to simplify deployment and improve scalability.

Before you begin

Use the Interoperability Matrix Tool to verify that your Linux OS, host bus adapter (HBA), HBA firmware, HBA boot BIOS, and ONTAP version support SAN booting.

Steps

1. Create a SAN boot namespace and map it to the host.
2. Enable SAN booting in the server BIOS for the ports to which the SAN boot namespace is mapped.

   For information on how to enable the HBA BIOS, see your vendor-specific documentation.

3. Verify that the configuration was successful by rebooting the host and verifying that the OS is up and running.

Step 2: Validate software versions

Use the following procedure to validate the minimum supported Rocky Linux 9.4 software versions.

Steps

1. Install Rocky Linux 9.4 on the server. After the installation is complete, verify that you are running the specified Rocky Linux 9.4 kernel:

   ```
   uname -r
   ```

   The following example shows a Rocky Linux kernel version:

   ```
   5.14.0-570.12.1.el9_6.x86_64
   ```

2. Install the `nvme-cli` package:

   ```
   rpm -qa|grep nvme-cli
   ```

   The following example shows an nvme-cli package version:

   ```
   nvme-cli-2.11-5.el9.x86_64
   ```

3. Install the `libnvme` package:

   ```
   rpm -qa|grep libnvme
   ```

   The following example shows an `libnvme` package version:

```
libnvme-1.11.1-1.el9.x86_64
```

4. On the Rocky Linux host, check the hostnqn string at `/etc/nvme/hostnqn`:

```
cat /etc/nvme/hostnqn
```

The following example shows an `hostnqn` version:

```
nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
```

5. Verify that the `hostnqn` string matches the `hostnqn` string for the corresponding subsystem on the ONTAP array:

```
::> vserver nvme subsystem host show -vserver vs_coexistence_LPE36002
```

Show example

```
Vserver Subsystem Priority  Host NQN
------- --------- --------
------------------------------------------------
vs_coexistence_LPE36002
        nvme
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
        nvme_1
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
        nvme_2
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
        nvme_3
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
4 entries were displayed.
```

If the `hostnqn` strings do not match, use the `vserver modify` command to update the `hostnqn` string on your corresponding ONTAP array subsystem to match the `hostnqn` string from `/etc/nvme/hostnqn` on the host.

Step 3: Configure NVMe/FC

You can configure NVMe/FC with Broadcom/Emulex FC or Marvell/Qlogic FC adapters. You also need to manually discover the NVMe/TCP subsystems and namespaces.

Broadcom/Emulex

Configure NVMe/FC for a Broadcom/Emulex adapter.

Steps

1. Verify that you are using the supported adapter model:
   a. Display the model names:

      ```
      cat /sys/class/scsi_host/host*/modelname
      ```

      You should see the following output:

      ```
      LPe36002-M64
      LPe36002-M64
      ```

   b. Display the model descriptions:

      ```
      cat /sys/class/scsi_host/host*/modeldesc
      ```

      You should see an output similar to the following example:

      ```
      Emulex LightPulse LPe36002-M64 2-Port 64Gb Fibre Channel Adapter
      Emulex LightPulse LPe36002-M64 2-Port 64Gb Fibre Channel Adapter
      ```

2. Verify that you are using the recommended Broadcom `lpfc` firmware and inbox driver:
   a. Display the firmware version:

      ```
      cat /sys/class/scsi_host/host*/fwrev
      ```

      The following example shows firmware versions:

      ```
      14.4.317.10, sli-4:6:d
      14.4.317.10, sli-4:6:d
      ```

   b. Display the inbox driver version:

      ```
      cat /sys/module/lpfc/version`
      ```

      The following example shows a driver version:

```
0:14.4.0.2
```

For the current list of supported adapter driver and firmware versions, see the Interoperability Matrix Tool.

3. Verify that the expected output of `lpfc_enable_fc4_type` is set to `3`:

```
cat /sys/module/lpfc/parameters/lpfc_enable_fc4_type
```

4. Verify that you can view your initiator ports:

```
cat /sys/class/fc_host/host*/port_name
```

The following example shows port identities:

```
0x100000109bf044b1
0x100000109bf044b2
```

5. Verify that your initiator ports are online:

```
cat /sys/class/fc_host/host*/port_state
```

You should see the following output:

```
Online
Online
```

6. Verify that the NVMe/FC initiator ports are enabled and that the target ports are visible:

```
cat /sys/class/scsi_host/host*/nvme_info
```

Show example

```
NVME Initiator Enabled
XRI Dist lpfc2 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc2 WWPN x100000109bf044b1 WWNN x200000109bf044b1
DID x022a00 ONLINE
NVME RPORT       WWPN x202fd039eaa7dfc8 WWNN x202cd039eaa7dfc8
DID x021310 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x202dd039eaa7dfc8 WWNN x202cd039eaa7dfc8
DID x020b10 TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 0000000810 Cmpl 0000000810 Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000007b098f07 Issue 000000007aee27c4 OutIO
fffffffffe498bd
        abort 000013b4 noxri 00000000 nondlp 00000058 qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 000013b4 Err 00021443

NVME Initiator Enabled
XRI Dist lpfc3 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc3 WWPN x100000109bf044b2 WWNN x200000109bf044b2
DID x021b00 ONLINE
NVME RPORT       WWPN x2033d039eaa7dfc8 WWNN x202cd039eaa7dfc8
DID x020110 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2032d039eaa7dfc8 WWNN x202cd039eaa7dfc8
DID x022910 TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 0000000840 Cmpl 0000000840 Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000007afd4434 Issue 000000007ae31b83 OutIO
fffffffffe5d74f
        abort 000014a5 noxri 00000000 nondlp 0000006a qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 000014a5 Err 0002149a
```

Marvell/QLogic

Configure NVMe/FC for a Marvell/QLogic adapter.

> The native inbox qla2xxx driver included in the Rocky Linux kernel has the latest fixes. These fixes are essential for ONTAP support.

Steps

1. Verify that you are running the supported adapter driver and firmware versions:

```
cat /sys/class/fc_host/host*/symbolic_name
```

The follow example shows driver and firmware versions:

```
QLE2742 FW:v9.14.00 DVR:v10.02.09.200-k
QLE2742 FW:v9.14.00 DVR:v10.02.09.200-k
```

2. Verify that `ql2xnvmeenable` is set. This enables the Marvell adapter to function as an NVMe/FC initiator:

```
cat /sys/module/qla2xxx/parameters/ql2xnvmeenable
```

The expected output is 1.

Step 4: Optionally, enable 1MB I/O

You can enable I/O requests of size 1MB for NVMe/FC configured with a Broadcom adapter. ONTAP reports a Max Data Transfer Size (MDTS) of 8 in the Identify Controller data. This means the maximum I/O request size can be up to 1MB. To issue I/O requests of size 1MB, you need to increase the lpfc value of the `lpfc_sg_seg_cnt` parameter to 256 from the default value of 64.

> These steps don't apply to Qlogic NVMe/FC hosts.

Steps

1. Set the `lpfc_sg_seg_cnt` parameter to 256:

```
cat /etc/modprobe.d/lpfc.conf
```

```
options lpfc lpfc_sg_seg_cnt=256
```

2. Run the `dracut -f` command, and reboot the host.
3. Verify that the value for `lpfc_sg_seg_cnt` is 256:

```
cat /sys/module/lpfc/parameters/lpfc_sg_seg_cnt
```

Step 5: Configure NVMe/TCP

The NVMe/TCP protocol doesn't support the auto-connect operation. Instead, you can discover the NVMe/TCP subsystems and namespaces by performing the NVMe/TCP connect or connect-all operations manually.

Steps

1. Verify that the initiator port can fetch the discovery log page data across the supported NVMe/TCP LIFs:

   ```
   nvme discover -t tcp -w host-traddr -a traddr
   ```

Show example

```
nvme discover -t tcp -w 192.168.1.31 -a 192.168.1.24
Discovery Log Number of Records 20, Generation counter 25
=====Discovery Log Entry 0======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  4
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr:  192.168.2.25
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 1======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  2
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr:  192.168.1.25
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 2======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  5
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr:  192.168.2.24
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 3======
trtype:  tcp
adrfam:  ipv4
```

```
subtype: current discovery subsystem
treq:    not specified
portid:  1
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr:  192.168.1.24
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 4======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_1
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 5======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_1
traddr:  192.168.1.25
eflags:  none
sectype: none
=====Discovery Log Entry 6======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  5
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_1
```

```
traddr:  192.168.2.24
eflags:  none
sectype: none
=====Discovery Log Entry 7======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_1
traddr:  192.168.1.24
eflags:  none
sectype: none
=====Discovery Log Entry 8======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_4
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 9======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_4
traddr:  192.168.1.25
eflags:  none
sectype: none
=====Discovery Log Entry 10======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
```

```
treq:    not specified
portid:  5
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_4
traddr:  192.168.2.24
eflags:  none
sectype: none
=====Discovery Log Entry 11======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_4
traddr:  192.168.1.24
eflags:  none
sectype: none
=====Discovery Log Entry 12======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_3
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 13======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_3
traddr:  192.168.1.25
```

```
eflags:  none
sectype: none
=====Discovery Log Entry 14======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  5
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_3
traddr:  192.168.2.24
eflags:  none
sectype: none
=====Discovery Log Entry 15======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_3
traddr:  192.168.1.24
eflags:  none
sectype: none
=====Discovery Log Entry 16======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_2
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 17======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
```

```
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_2
traddr:  192.168.1.25
eflags:  none
sectype: none
=====Discovery Log Entry 18======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  5
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_2
traddr:  192.168.2.24
eflags:  none
sectype: none
=====Discovery Log Entry 19======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_2
traddr:  192.168.1.24
eflags:  none
sectype: none
```

2. Verify that the other NVMe/TCP initiator-target LIF combinations are able to successfully fetch discovery log page data:

```
nvme discover -t tcp -w host-traddr -a traddr
```

Show example

```
nvme discover -t tcp -w 192.168.1.31 -a 192.168.1.24
nvme discover -t tcp -w 192.168.2.31 -a 192.168.2.24
nvme discover -t tcp -w 192.168.1.31 -a 192.168.1.25
nvme discover -t tcp -w 192.168.2.31 -a 192.168.2.25
```

3. Run the `nvme connect-all` command across all the supported NVMe/TCP initiator-target LIFs across the nodes:

```
nvme connect-all -t tcp -w host-traddr -a traddr
```

Show example

```
nvme    connect-all -t  tcp -w  192.168.1.31    -a  192.168.1.24
nvme    connect-all -t  tcp -w  192.168.2.31    -a  192.168.2.24
nvme    connect-all -t  tcp -w  192.168.1.31    -a  192.168.1.25
nvme    connect-all -t  tcp -w  192.168.2.31    -a  192.168.2.25
```

> Beginning with Rocky Linux 9.4, the setting for the NVMe/TCP `ctrl_loss_tmo timeout` is automatically set to "off". As a result:
>
> - There are no limits on the number of retries (indefinite retry).
> - You don’t need to manually configure a specific `ctrl_loss_tmo timeout` duration when using the `nvme connect` or `nvme connect-all` commands (option -l).
> - The NVMe/TCP controllers don’t experience timeouts in the event of a path failure and remain connected indefinitely.

Step 6: Validate NVMe-oF

Verify that the in-kernel NVMe multipath status, ANA status, and ONTAP namespaces are correct for the NVMe-oF configuration.

Steps

1. Verify that the in-kernel NVMe multipath is enabled:

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

You should see the following output:

```
Y
```

2. Verify that the appropriate NVMe-oF settings (such as, model set to NetApp ONTAP Controller and load balancing iopolicy set to round-robin) for the respective ONTAP namespaces correctly reflect on the host:

   a. Display the subsystems:

   ```
   cat /sys/class/nvme-subsystem/nvme-subsys*/model
   ```

   You should see the following output:

   ```
   NetApp ONTAP Controller
   NetApp ONTAP Controller
   ```

   b. Display the policy:

   ```
   cat /sys/class/nvme-subsystem/nvme-subsys*/iopolicy
   ```

   You should see the following output:

   ```
   round-robin
   round-robin
   ```

3. Verify that the namespaces are created and correctly discovered on the host:

   ```
   nvme list
   ```

   Show example

   ```
   Node         SN                   Model
   ---------------------------------------------------------
   /dev/nvme4n1 81Ix2BVuekWcAAAAAAAB   NetApp ONTAP Controller


   Namespace Usage    Format             FW             Rev
   -----------------------------------------------------------
   1                 21.47 GB / 21.47 GB   4 KiB + 0 B   FFFFFFFF
   ```

4. Verify that the controller state of each path is live and has the correct ANA status:

NVMe/FC

```
nvme list-subsys /dev/nvme4n5
```

Show example

```
nvme-subsys4 - NQN=nqn.1992-
08.com.netapp:sn.3a5d31f5502c11ef9f50d039eab6cb6d:subsystem.nvme
_1
               hostnqn=nqn.2014-08.org.nvmexpress:uuid:e6dade64-
216d-
11ec-b7bb-7ed30a5482c3
iopolicy=round-robin\
+- nvme1 fc traddr=nn-0x2082d039eaa7dfc8:pn-
0x2088d039eaa7dfc8,host_traddr=nn-0x20000024ff752e6d:pn-
0x21000024ff752e6d live optimized
+- nvme12 fc traddr=nn-0x2082d039eaa7dfc8:pn-
0x208ad039eaa7dfc8,host_traddr=nn-0x20000024ff752e6d:pn-
0x21000024ff752e6d live non-optimized
+- nvme10 fc traddr=nn-0x2082d039eaa7dfc8:pn-
0x2087d039eaa7dfc8,host_traddr=nn-0x20000024ff752e6c:pn-
0x21000024ff752e6c live non-optimized
+- nvme3 fc traddr=nn-0x2082d039eaa7dfc8:pn-
0x2083d039eaa7dfc8,host_traddr=nn-0x20000024ff752e6c:pn-
0x21000024ff752e6c live optimized
```

NVMe/TCP

```
nvme list-subsys /dev/nvme1n1
```

Show example

```
nvme-subsys5 - NQN=nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme
_tcp_3
hostnqn=nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0035-5910-804b-
b5c04f444d33
iopolicy=round-robin
\
+- nvme13 tcp
traddr=192.168.2.25,trsvcid=4420,host_traddr=192.168.2.31,
src_addr=192.168.2.31 live optimized
+- nvme14 tcp
traddr=192.168.2.24,trsvcid=4420,host_traddr=192.168.2.31,
src_addr=192.168.2.31 live non-optimized
+- nvme5 tcp
traddr=192.168.1.25,trsvcid=4420,host_traddr=192.168.1.31,
src_addr=192.168.1.31 live optimized
+- nvme6 tcp
traddr=192.168.1.24,trsvcid=4420,host_traddr=192.168.1.31,
src_addr=192.168.1.31 live non-optimized
```

5. Verify that the NetApp plug-in displays the correct values for each ONTAP namespace device:

Column

```
nvme netapp ontapdevices -o column
```

Show example

```
Device        Vserver   Namespace Path
----------------------- ------------------------------
/dev/nvme1n1     linux_tcnvme_iscsi
/vol/tcpnvme_1_0_0/tcpnvme_ns

NSID       UUID                                   Size
------------------------------------------------------------
1    5f7f630d-8ea5-407f-a490-484b95b15dd6   21.47GB
```

JSON

```
nvme netapp ontapdevices -o json
```

Show example


```
{
  "ONTAPdevices":[
    {
      "Device":"/dev/nvme1n1",
      "Vserver":"linux_tcnvme_iscsi",
      "Namespace_Path":"/vol/tcpnvme_1_0_0/tcpnvme_ns",
      "NSID":1,
      "UUID":"5f7f630d-8ea5-407f-a490-484b95b15dd6",
      "Size":"21.47GB",
      "LBA_Data_Size":4096,
      "Namespace_Size":5242880
    },
]
}
```


Step 7: Review the known issues

There are no known issues.

Configure Rocky Linux 9.3 with NVMe-oF for ONTAP storage

Rocky Linux 9.3 hosts support the NVMe/FC and NVMe/TCP protocols with Asymmetric Namespace Access (ANA). ANA is equivalent to asymmetric logical unit access (ALUA) multipathing in iSCSI and FCP environments and is implemented using the in-kernel NVMe multipath feature.

For additional details on supported configurations, see the Interoperability Matrix Tool.

About this task

You can use the following support and features with the NVMe-oF host configuration for Rocky Linux 9.3. You should also review the known limitations before starting the configuration process.

- Support available:
  - Support for NVMe over TCP (NVMe/TCP) in addition to NVMe over Fibre Channel (NVMe/FC). The NetApp plug-in in the native `nvme-cli` package displays ONTAP details for both NVMe/FC and NVMe/TCP namespaces.
  - Running NVMe and SCSI traffic on the same host. For example, you can configure dm-multipath for SCSI mpath devices on SCSI LUNs and use NVMe multipath to configure NVMe-oF namespace devices on the host.
- Features available:
  - Rocky Linux 9.3 enables in-kernel NVMe multipath for NVMe namespaces by default, removing the need for explicit settings.
- Known limitations:
  - SAN booting using the NVMe-oF protocol is currently not supported.

Step 1: Optionally, enable SAN booting

You can configure your host to use SAN booting to simplify deployment and improve scalability.

Before you begin

Use the Interoperability Matrix Tool to verify that your Linux OS, host bus adapter (HBA), HBA firmware, HBA boot BIOS, and ONTAP version support SAN booting.

Steps

1. Create a SAN boot namespace and map it to the host.
2. Enable SAN booting in the server BIOS for the ports to which the SAN boot namespace is mapped.

   For information on how to enable the HBA BIOS, see your vendor-specific documentation.

3. Verify that the configuration was successful by rebooting the host and verifying that the OS is up and running.

Step 2: Validate software versions

Use the following procedure to validate the minimum supported Rocky Linux 9.3 software versions.

Steps

1. Install Rocky Linux 9.3 on the server. After the installation is complete, verify that you are running the specified Rocky Linux 9.3 kernel:

```
uname -r
```

The following example shows a Rocky Linux kernel version:

```
5.14.0-570.12.1.el9_6.x86_64
```

2. Install the `nvme-cli` package:

```
rpm -qa|grep nvme-cli
```

The following example shows an nvme-cli package version:

```
nvme-cli-2.11-5.el9.x86_64
```

3. Install the `libnvme` package:

```
rpm -qa|grep libnvme
```

The following example shows an `libnvme` package version:

```
libnvme-1.11.1-1.el9.x86_64
```

4. On the Rocky Linux host, check the hostnqn string at `/etc/nvme/hostnqn`:

```
cat /etc/nvme/hostnqn
```

The following example shows an `hostnqn` version:

```
nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
```

5. Verify that the `hostnqn` string matches the `hostnqn` string for the corresponding subsystem on the ONTAP array:

```
::> vserver nvme subsystem host show -vserver vs_coexistence_LPE36002
```

Show example

```
Vserver Subsystem Priority  Host NQN
------- --------- --------
------------------------------------------------
vs_coexistence_LPE36002
        nvme
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
        nvme_1
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
        nvme_2
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
        nvme_3
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
4 entries were displayed.
```

If the `hostnqn` strings do not match, use the `vserver modify` command to update the `hostnqn` string on your corresponding ONTAP array subsystem to match the `hostnqn` string from `/etc/nvme/hostnqn` on the host.

Step 3: Configure NVMe/FC

You can configure NVMe/FC with Broadcom/Emulex FC or Marvell/Qlogic FC adapters. You also need to manually discover the NVMe/TCP subsystems and namespaces.

Broadcom/Emulex

Configure NVMe/FC for a Broadcom/Emulex adapter.

Steps

1. Verify that you are using the supported adapter model:
   a. Display the model names:

      ```
      cat /sys/class/scsi_host/host*/modelname
      ```

      You should see the following output:

      ```
      LPe36002-M64
      LPe36002-M64
      ```

   b. Display the model descriptions:

      ```
      cat /sys/class/scsi_host/host*/modeldesc
      ```

      You should see an output similar to the following example:

      ```
      Emulex LightPulse LPe36002-M64 2-Port 64Gb Fibre Channel Adapter
      Emulex LightPulse LPe36002-M64 2-Port 64Gb Fibre Channel Adapter
      ```

2. Verify that you are using the recommended Broadcom `lpfc` firmware and inbox driver:
   a. Display the firmware version:

      ```
      cat /sys/class/scsi_host/host*/fwrev
      ```

      The following example shows firmware versions:

      ```
      14.4.317.10, sli-4:6:d
      14.4.317.10, sli-4:6:d
      ```

   b. Display the inbox driver version:

      ```
      cat /sys/module/lpfc/version`
      ```

      The following example shows a driver version:

```
0:14.4.0.2
```

For the current list of supported adapter driver and firmware versions, see the Interoperability Matrix Tool.

3. Verify that the expected output of `lpfc_enable_fc4_type` is set to `3`:

   ```
   cat /sys/module/lpfc/parameters/lpfc_enable_fc4_type
   ```

4. Verify that you can view your initiator ports:

   ```
   cat /sys/class/fc_host/host*/port_name
   ```

   The following example shows port identities:

   ```
   0x100000109bf044b1
   0x100000109bf044b2
   ```

5. Verify that your initiator ports are online:

   ```
   cat /sys/class/fc_host/host*/port_state
   ```

   You should see the following output:

   ```
   Online
   Online
   ```

6. Verify that the NVMe/FC initiator ports are enabled and that the target ports are visible:

   ```
   cat /sys/class/scsi_host/host*/nvme_info
   ```

Show example

```
NVME Initiator Enabled
XRI Dist lpfc2 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc2 WWPN x100000109bf044b1 WWNN x200000109bf044b1
DID x022a00 ONLINE
NVME RPORT       WWPN x202fd039eaa7dfc8 WWNN x202cd039eaa7dfc8
DID x021310 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x202dd039eaa7dfc8 WWNN x202cd039eaa7dfc8
DID x020b10 TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 0000000810 Cmpl 0000000810 Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000007b098f07 Issue 000000007aee27c4 OutIO
fffffffffe498bd
        abort 000013b4 noxri 00000000 nondlp 00000058 qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 000013b4 Err 00021443

NVME Initiator Enabled
XRI Dist lpfc3 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc3 WWPN x100000109bf044b2 WWNN x200000109bf044b2
DID x021b00 ONLINE
NVME RPORT       WWPN x2033d039eaa7dfc8 WWNN x202cd039eaa7dfc8
DID x020110 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2032d039eaa7dfc8 WWNN x202cd039eaa7dfc8
DID x022910 TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 0000000840 Cmpl 0000000840 Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000007afd4434 Issue 000000007ae31b83 OutIO
fffffffffe5d74f
        abort 000014a5 noxri 00000000 nondlp 0000006a qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 000014a5 Err 0002149a
```

Marvell/QLogic

Configure NVMe/FC for a Marvell/QLogic adapter.

> The native inbox qla2xxx driver included in the Rocky Linux kernel has the latest fixes. These fixes are essential for ONTAP support.

Steps

1. Verify that you are running the supported adapter driver and firmware versions:

   ```
   cat /sys/class/fc_host/host*/symbolic_name
   ```

   The follow example shows driver and firmware versions:

   ```
   QLE2742 FW:v9.14.00 DVR:v10.02.09.200-k
   QLE2742 FW:v9.14.00 DVR:v10.02.09.200-k
   ```

2. Verify that `ql2xnvmeenable` is set. This enables the Marvell adapter to function as an NVMe/FC initiator:

   ```
   cat /sys/module/qla2xxx/parameters/ql2xnvmeenable
   ```

   The expected output is 1.

Step 4: Optionally, enable 1MB I/O

You can enable I/O requests of size 1MB for NVMe/FC configured with a Broadcom adapter. ONTAP reports a Max Data Transfer Size (MDTS) of 8 in the Identify Controller data. This means the maximum I/O request size can be up to 1MB. To issue I/O requests of size 1MB, you need to increase the lpfc value of the `lpfc_sg_seg_cnt` parameter to 256 from the default value of 64.

> These steps don't apply to Qlogic NVMe/FC hosts.

Steps

1. Set the `lpfc_sg_seg_cnt` parameter to 256:

   ```
   cat /etc/modprobe.d/lpfc.conf
   ```

   ```
   options lpfc lpfc_sg_seg_cnt=256
   ```

2. Run the `dracut -f` command, and reboot the host.
3. Verify that the value for `lpfc_sg_seg_cnt` is 256:

   ```
   cat /sys/module/lpfc/parameters/lpfc_sg_seg_cnt
   ```

Step 5: Configure NVMe/TCP

The NVMe/TCP protocol doesn’t support the auto-connect operation. Instead, you can discover the NVMe/TCP subsystems and namespaces by performing the NVMe/TCP connect or connect-all operations manually.

Steps

1. Verify that the initiator port can fetch the discovery log page data across the supported NVMe/TCP LIFs:

   ```
   nvme discover -t tcp -w host-traddr -a traddr
   ```

Show example

```
nvme discover -t tcp -w 192.168.1.31 -a 192.168.1.24
Discovery Log Number of Records 20, Generation counter 25
=====Discovery Log Entry 0======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  4
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr:  192.168.2.25
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 1======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  2
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr:  192.168.1.25
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 2======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  5
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr:  192.168.2.24
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 3======
trtype:  tcp
adrfam:  ipv4
```

```
subtype: current discovery subsystem
treq:    not specified
portid:  1
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr:  192.168.1.24
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 4======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_1
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 5======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_1
traddr:  192.168.1.25
eflags:  none
sectype: none
=====Discovery Log Entry 6======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  5
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_1
```

```
traddr:  192.168.2.24
eflags:  none
sectype: none
=====Discovery Log Entry 7======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_1
traddr:  192.168.1.24
eflags:  none
sectype: none
=====Discovery Log Entry 8======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_4
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 9======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_4
traddr:  192.168.1.25
eflags:  none
sectype: none
=====Discovery Log Entry 10======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
```

```
treq:    not specified
portid:  5
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_4
traddr:  192.168.2.24
eflags:  none
sectype: none
=====Discovery Log Entry 11======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_4
traddr:  192.168.1.24
eflags:  none
sectype: none
=====Discovery Log Entry 12======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_3
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 13======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_3
traddr:  192.168.1.25
```

```
eflags:  none
sectype: none
=====Discovery Log Entry 14======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  5
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_3
traddr:  192.168.2.24
eflags:  none
sectype: none
=====Discovery Log Entry 15======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_3
traddr:  192.168.1.24
eflags:  none
sectype: none
=====Discovery Log Entry 16======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_2
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 17======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
```

```
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_2
traddr:  192.168.1.25
eflags:  none
sectype: none
=====Discovery Log Entry 18======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  5
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_2
traddr:  192.168.2.24
eflags:  none
sectype: none
=====Discovery Log Entry 19======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_2
traddr:  192.168.1.24
eflags:  none
sectype: none
```

2. Verify that the other NVMe/TCP initiator-target LIF combinations are able to successfully fetch discovery log page data:

```
nvme discover -t tcp -w host-traddr -a traddr
```

Show example

```
nvme discover -t tcp -w 192.168.1.31 -a 192.168.1.24
nvme discover -t tcp -w 192.168.2.31 -a 192.168.2.24
nvme discover -t tcp -w 192.168.1.31 -a 192.168.1.25
nvme discover -t tcp -w 192.168.2.31 -a 192.168.2.25
```

3. Run the `nvme connect-all` command across all the supported NVMe/TCP initiator-target LIFs across the nodes:

```
nvme connect-all -t tcp -w host-traddr -a traddr
```

Show example

```
nvme    connect-all -t  tcp -w  192.168.1.31    -a  192.168.1.24
nvme    connect-all -t  tcp -w  192.168.2.31    -a  192.168.2.24
nvme    connect-all -t  tcp -w  192.168.1.31    -a  192.168.1.25
nvme    connect-all -t  tcp -w  192.168.2.31    -a  192.168.2.25
```

Step 6: Validate NVMe-oF

Verify that the in-kernel NVMe multipath status, ANA status, and ONTAP namespaces are correct for the NVMe-oF configuration.

Steps

1. Verify that the in-kernel NVMe multipath is enabled:

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

You should see the following output:

```
Y
```

2. Verify that the appropriate NVMe-oF settings (such as, model set to NetApp ONTAP Controller and load balancing iopolicy set to round-robin) for the respective ONTAP namespaces correctly reflect on the host:
   a. Display the subsystems:

```
cat /sys/class/nvme-subsystem/nvme-subsys*/model
```

You should see the following output:

```
NetApp ONTAP Controller
NetApp ONTAP Controller
```

b. Display the policy:

```
cat /sys/class/nvme-subsystem/nvme-subsys*/iopolicy
```

You should see the following output:

```
round-robin
round-robin
```

3. Verify that the namespaces are created and correctly discovered on the host:

```
nvme list
```

Show example

```
Node         SN                   Model
---------------------------------------------------------
/dev/nvme4n1 81Ix2BVuekWcAAAAAAB   NetApp ONTAP Controller


Namespace Usage    Format             FW             Rev
-----------------------------------------------------------
1                 21.47 GB / 21.47 GB   4 KiB + 0 B   FFFFFFFF
```

4. Verify that the controller state of each path is live and has the correct ANA status:

NVMe/FC

```
nvme list-subsys /dev/nvme4n5
```

Show example

```
nvme-subsys4 - NQN=nqn.1992-
08.com.netapp:sn.3a5d31f5502c11ef9f50d039eab6cb6d:subsystem.nvme
_1
               hostnqn=nqn.2014-08.org.nvmexpress:uuid:e6dade64-
216d-
11ec-b7bb-7ed30a5482c3
iopolicy=round-robin\
+- nvme1 fc traddr=nn-0x2082d039eaa7dfc8:pn-
0x2088d039eaa7dfc8,host_traddr=nn-0x20000024ff752e6d:pn-
0x21000024ff752e6d live optimized
+- nvme12 fc traddr=nn-0x2082d039eaa7dfc8:pn-
0x208ad039eaa7dfc8,host_traddr=nn-0x20000024ff752e6d:pn-
0x21000024ff752e6d live non-optimized
+- nvme10 fc traddr=nn-0x2082d039eaa7dfc8:pn-
0x2087d039eaa7dfc8,host_traddr=nn-0x20000024ff752e6c:pn-
0x21000024ff752e6c live non-optimized
+- nvme3 fc traddr=nn-0x2082d039eaa7dfc8:pn-
0x2083d039eaa7dfc8,host_traddr=nn-0x20000024ff752e6c:pn-
0x21000024ff752e6c live optimized
```

NVMe/TCP

```
nvme list-subsys /dev/nvme1n1
```

Show example

```
nvme-subsys5 - NQN=nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme
_tcp_3
hostnqn=nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0035-5910-804b-
b5c04f444d33
iopolicy=round-robin
\
+- nvme13 tcp
traddr=192.168.2.25,trsvcid=4420,host_traddr=192.168.2.31,
src_addr=192.168.2.31 live optimized
+- nvme14 tcp
traddr=192.168.2.24,trsvcid=4420,host_traddr=192.168.2.31,
src_addr=192.168.2.31 live non-optimized
+- nvme5 tcp
traddr=192.168.1.25,trsvcid=4420,host_traddr=192.168.1.31,
src_addr=192.168.1.31 live optimized
+- nvme6 tcp
traddr=192.168.1.24,trsvcid=4420,host_traddr=192.168.1.31,
src_addr=192.168.1.31 live non-optimized
```

5. Verify that the NetApp plug-in displays the correct values for each ONTAP namespace device:

Column

```
nvme netapp ontapdevices -o column
```

Show example

```
Device        Vserver   Namespace Path
----------------------- ------------------------------
/dev/nvme1n1     linux_tcnvme_iscsi
/vol/tcpnvme_1_0_0/tcpnvme_ns

NSID       UUID                                   Size
------------------------------------------------------------
1    5f7f630d-8ea5-407f-a490-484b95b15dd6   21.47GB
```

JSON

```
nvme netapp ontapdevices -o json
```

Show example


```
{
  "ONTAPdevices":[
    {
      "Device":"/dev/nvme1n1",
      "Vserver":"linux_tcnvme_iscsi",
      "Namespace_Path":"/vol/tcpnvme_1_0_0/tcpnvme_ns",
      "NSID":1,
      "UUID":"5f7f630d-8ea5-407f-a490-484b95b15dd6",
      "Size":"21.47GB",
      "LBA_Data_Size":4096,
      "Namespace_Size":5242880
    },
]
}
```


Step 7: Review the known issues

There are no known issues.

Configure Rocky Linux 9.2 with NVMe-oF for ONTAP storage

Rocky Linux 9.2 hosts support the NVMe/FC and NVMe/TCP protocols with Asymmetric Namespace Access (ANA). ANA is equivalent to asymmetric logical unit access (ALUA) multipathing in iSCSI and FCP environments and is implemented using the in-kernel NVMe multipath feature.

For additional details on supported configurations, see the Interoperability Matrix Tool.

About this task

You can use the following support and features with the NVMe-oF host configuration for Rocky Linux 9.2. You should also review the known limitations before starting the configuration process.

- Support available:
  - Support for NVMe over TCP (NVMe/TCP) in addition to NVMe over Fibre Channel (NVMe/FC). The NetApp plug-in in the native `nvme-cli` package displays ONTAP details for both NVMe/FC and NVMe/TCP namespaces.
  - Running NVMe and SCSI traffic on the same host. For example, you can configure dm-multipath for SCSI mpath devices on SCSI LUNs and use NVMe multipath to configure NVMe-oF namespace devices on the host.
- Features available:
  - Rocky Linux 9.2 enables in-kernel NVMe multipath for NVMe namespaces by default, removing the need for explicit settings.
- Known limitations:
  - SAN booting using the NVMe-oF protocol is currently not supported.

Step 1: Optionally, enable SAN booting

You can configure your host to use SAN booting to simplify deployment and improve scalability.

Before you begin

Use the Interoperability Matrix Tool to verify that your Linux OS, host bus adapter (HBA), HBA firmware, HBA boot BIOS, and ONTAP version support SAN booting.

Steps

1. Create a SAN boot namespace and map it to the host.
2. Enable SAN booting in the server BIOS for the ports to which the SAN boot namespace is mapped.

   For information on how to enable the HBA BIOS, see your vendor-specific documentation.

3. Verify that the configuration was successful by rebooting the host and verifying that the OS is up and running.

Step 2: Validate software versions

Use the following procedure to validate the minimum supported Rocky Linux 9.2 software versions.

Steps

1. Install Rocky Linux 9.2 on the server. After the installation is complete, verify that you are running the specified Rocky Linux 9.2 kernel:

```
uname -r
```

The following example shows a Rocky Linux kernel version:

```
5.14.0-570.12.1.el9_6.x86_64
```

2. Install the `nvme-cli` package:

```
rpm -qa|grep nvme-cli
```

The following example shows an nvme-cli package version:

```
nvme-cli-2.11-5.el9.x86_64
```

3. Install the `libnvme` package:

```
rpm -qa|grep libnvme
```

The following example shows an `libnvme` package version:

```
libnvme-1.11.1-1.el9.x86_64
```

4. On the Rocky Linux host, check the hostnqn string at `/etc/nvme/hostnqn`:

```
cat /etc/nvme/hostnqn
```

The following example shows an `hostnqn` version:

```
nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
```

5. Verify that the `hostnqn` string matches the `hostnqn` string for the corresponding subsystem on the ONTAP array:

```
::> vserver nvme subsystem host show -vserver vs_coexistence_LPE36002
```

Show example

```
Vserver Subsystem Priority  Host NQN
------- --------- --------
------------------------------------------------
vs_coexistence_LPE36002
        nvme
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
        nvme_1
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
        nvme_2
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
        nvme_3
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
4 entries were displayed.
```

If the `hostnqn` strings do not match, use the `vserver modify` command to update the `hostnqn` string on your corresponding ONTAP array subsystem to match the `hostnqn` string from `/etc/nvme/hostnqn` on the host.

Step 3: Configure NVMe/FC

You can configure NVMe/FC with Broadcom/Emulex FC or Marvell/Qlogic FC adapters. You also need to manually discover the NVMe/TCP subsystems and namespaces.

Broadcom/Emulex

Configure NVMe/FC for a Broadcom/Emulex adapter.

Steps

1. Verify that you are using the supported adapter model:
   a. Display the model names:

   ```
   cat /sys/class/scsi_host/host*/modelname
   ```

   You should see the following output:

   ```
   LPe36002-M64
   LPe36002-M64
   ```

   b. Display the model descriptions:

   ```
   cat /sys/class/scsi_host/host*/modeldesc
   ```

   You should see an output similar to the following example:

   ```
   Emulex LightPulse LPe36002-M64 2-Port 64Gb Fibre Channel Adapter
   Emulex LightPulse LPe36002-M64 2-Port 64Gb Fibre Channel Adapter
   ```

2. Verify that you are using the recommended Broadcom `lpfc` firmware and inbox driver:
   a. Display the firmware version:

   ```
   cat /sys/class/scsi_host/host*/fwrev
   ```

   The following example shows firmware versions:

   ```
   14.4.317.10, sli-4:6:d
   14.4.317.10, sli-4:6:d
   ```

   b. Display the inbox driver version:

   ```
   cat /sys/module/lpfc/version`
   ```

   The following example shows a driver version:

```
0:14.4.0.2
```

For the current list of supported adapter driver and firmware versions, see the Interoperability Matrix Tool.

3. Verify that the expected output of `lpfc_enable_fc4_type` is set to `3`:

   ```
   cat /sys/module/lpfc/parameters/lpfc_enable_fc4_type
   ```

4. Verify that you can view your initiator ports:

   ```
   cat /sys/class/fc_host/host*/port_name
   ```

   The following example shows port identities:

   ```
   0x100000109bf044b1
   0x100000109bf044b2
   ```

5. Verify that your initiator ports are online:

   ```
   cat /sys/class/fc_host/host*/port_state
   ```

   You should see the following output:

   ```
   Online
   Online
   ```

6. Verify that the NVMe/FC initiator ports are enabled and that the target ports are visible:

   ```
   cat /sys/class/scsi_host/host*/nvme_info
   ```

Show example

```
NVME Initiator Enabled
XRI Dist lpfc2 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc2 WWPN x100000109bf044b1 WWNN x200000109bf044b1
DID x022a00 ONLINE
NVME RPORT       WWPN x202fd039eaa7dfc8 WWNN x202cd039eaa7dfc8
DID x021310 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x202dd039eaa7dfc8 WWNN x202cd039eaa7dfc8
DID x020b10 TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 0000000810 Cmpl 0000000810 Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000007b098f07 Issue 000000007aee27c4 OutIO
fffffffffe498bd
        abort 000013b4 noxri 00000000 nondlp 00000058 qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 000013b4 Err 00021443

NVME Initiator Enabled
XRI Dist lpfc3 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc3 WWPN x100000109bf044b2 WWNN x200000109bf044b2
DID x021b00 ONLINE
NVME RPORT       WWPN x2033d039eaa7dfc8 WWNN x202cd039eaa7dfc8
DID x020110 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2032d039eaa7dfc8 WWNN x202cd039eaa7dfc8
DID x022910 TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 0000000840 Cmpl 0000000840 Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000007afd4434 Issue 000000007ae31b83 OutIO
fffffffffe5d74f
        abort 000014a5 noxri 00000000 nondlp 0000006a qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 000014a5 Err 0002149a
```

Marvell/QLogic

Configure NVMe/FC for a Marvell/QLogic adapter.

> ⓘ The native inbox qla2xxx driver included in the Rocky Linux kernel has the latest fixes. These fixes are essential for ONTAP support.

Steps

1. Verify that you are running the supported adapter driver and firmware versions:

```
cat /sys/class/fc_host/host*/symbolic_name
```

The follow example shows driver and firmware versions:

```
QLE2742 FW:v9.14.00 DVR:v10.02.09.200-k
QLE2742 FW:v9.14.00 DVR:v10.02.09.200-k
```

2. Verify that `ql2xnvmeenable` is set. This enables the Marvell adapter to function as an NVMe/FC initiator:

```
cat /sys/module/qla2xxx/parameters/ql2xnvmeenable
```

The expected output is 1.

Step 4: Optionally, enable 1MB I/O

You can enable I/O requests of size 1MB for NVMe/FC configured with a Broadcom adapter. ONTAP reports a Max Data Transfer Size (MDTS) of 8 in the Identify Controller data. This means the maximum I/O request size can be up to 1MB. To issue I/O requests of size 1MB, you need to increase the lpfc value of the `lpfc_sg_seg_cnt` parameter to 256 from the default value of 64.

> These steps don't apply to Qlogic NVMe/FC hosts.

Steps

1. Set the `lpfc_sg_seg_cnt` parameter to 256:

```
cat /etc/modprobe.d/lpfc.conf
```

```
options lpfc lpfc_sg_seg_cnt=256
```

2. Run the `dracut -f` command, and reboot the host.
3. Verify that the value for `lpfc_sg_seg_cnt` is 256:

```
cat /sys/module/lpfc/parameters/lpfc_sg_seg_cnt
```

Step 5: Configure NVMe/TCP

The NVMe/TCP protocol doesn’t support the auto-connect operation. Instead, you can discover the NVMe/TCP subsystems and namespaces by performing the NVMe/TCP connect or connect-all operations manually.

Steps

1. Verify that the initiator port can fetch the discovery log page data across the supported NVMe/TCP LIFs:

```
nvme discover -t tcp -w host-traddr -a traddr
```

Show example

```
nvme discover -t tcp -w 192.168.1.31 -a 192.168.1.24
Discovery Log Number of Records 20, Generation counter 25
=====Discovery Log Entry 0======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  4
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr:  192.168.2.25
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 1======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  2
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr:  192.168.1.25
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 2======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  5
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr:  192.168.2.24
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 3======
trtype:  tcp
adrfam:  ipv4
```

```
subtype: current discovery subsystem
treq:    not specified
portid:  1
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr:  192.168.1.24
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 4======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_1
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 5======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_1
traddr:  192.168.1.25
eflags:  none
sectype: none
=====Discovery Log Entry 6======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  5
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_1
```

```
traddr:  192.168.2.24
eflags:  none
sectype: none
=====Discovery Log Entry 7======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_1
traddr:  192.168.1.24
eflags:  none
sectype: none
=====Discovery Log Entry 8======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_4
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 9======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_4
traddr:  192.168.1.25
eflags:  none
sectype: none
=====Discovery Log Entry 10======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
```

```
treq:    not specified
portid:  5
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_4
traddr:  192.168.2.24
eflags:  none
sectype: none
=====Discovery Log Entry 11======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_4
traddr:  192.168.1.24
eflags:  none
sectype: none
=====Discovery Log Entry 12======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_3
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 13======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_3
traddr:  192.168.1.25
```

```
eflags:  none
sectype: none
=====Discovery Log Entry 14======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  5
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_3
traddr:  192.168.2.24
eflags:  none
sectype: none
=====Discovery Log Entry 15======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_3
traddr:  192.168.1.24
eflags:  none
sectype: none
=====Discovery Log Entry 16======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_2
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 17======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
```

```
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_2
traddr:  192.168.1.25
eflags:  none
sectype: none
=====Discovery Log Entry 18======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  5
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_2
traddr:  192.168.2.24
eflags:  none
sectype: none
=====Discovery Log Entry 19======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_2
traddr:  192.168.1.24
eflags:  none
sectype: none
```

2. Verify that the other NVMe/TCP initiator-target LIF combinations are able to successfully fetch discovery log page data:

```
nvme discover -t tcp -w host-traddr -a traddr
```

Show example

```
nvme discover -t tcp -w 192.168.1.31 -a 192.168.1.24
nvme discover -t tcp -w 192.168.2.31 -a 192.168.2.24
nvme discover -t tcp -w 192.168.1.31 -a 192.168.1.25
nvme discover -t tcp -w 192.168.2.31 -a 192.168.2.25
```

3. Run the `nvme connect-all` command across all the supported NVMe/TCP initiator-target LIFs across the nodes:

```
nvme connect-all -t tcp -w host-traddr -a traddr
```

Show example

```
nvme    connect-all -t  tcp -w  192.168.1.31    -a  192.168.1.24
nvme    connect-all -t  tcp -w  192.168.2.31    -a  192.168.2.24
nvme    connect-all -t  tcp -w  192.168.1.31    -a  192.168.1.25
nvme    connect-all -t  tcp -w  192.168.2.31    -a  192.168.2.25
```

Step 6: Validate NVMe-oF

Verify that the in-kernel NVMe multipath status, ANA status, and ONTAP namespaces are correct for the NVMe-oF configuration.

Steps

1. Verify that the in-kernel NVMe multipath is enabled:

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

You should see the following output:

```
Y
```

2. Verify that the appropriate NVMe-oF settings (such as, model set to NetApp ONTAP Controller and load balancing iopolicy set to round-robin) for the respective ONTAP namespaces correctly reflect on the host:
   a. Display the subsystems:

```
cat /sys/class/nvme-subsystem/nvme-subsys*/model
```

You should see the following output:

```
NetApp ONTAP Controller
NetApp ONTAP Controller
```

b. Display the policy:

```
cat /sys/class/nvme-subsystem/nvme-subsys*/iopolicy
```

You should see the following output:

```
round-robin
round-robin
```

3. Verify that the namespaces are created and correctly discovered on the host:

```
nvme list
```

Show example

```
Node         SN                    Model
---------------------------------------------------------
/dev/nvme4n1 81Ix2BVuekWcAAAAAAB   NetApp ONTAP Controller


Namespace Usage    Format             FW             Rev
-----------------------------------------------------------
1                 21.47 GB / 21.47 GB   4 KiB + 0 B   FFFFFFFF
```

4. Verify that the controller state of each path is live and has the correct ANA status:

NVMe/FC

```
nvme list-subsys /dev/nvme4n5
```

Show example

```
nvme-subsys4 - NQN=nqn.1992-
08.com.netapp:sn.3a5d31f5502c11ef9f50d039eab6cb6d:subsystem.nvme
_1
               hostnqn=nqn.2014-08.org.nvmexpress:uuid:e6dade64-
216d-
11ec-b7bb-7ed30a5482c3
iopolicy=round-robin\
+- nvme1 fc traddr=nn-0x2082d039eaa7dfc8:pn-
0x2088d039eaa7dfc8,host_traddr=nn-0x20000024ff752e6d:pn-
0x21000024ff752e6d live optimized
+- nvme12 fc traddr=nn-0x2082d039eaa7dfc8:pn-
0x208ad039eaa7dfc8,host_traddr=nn-0x20000024ff752e6d:pn-
0x21000024ff752e6d live non-optimized
+- nvme10 fc traddr=nn-0x2082d039eaa7dfc8:pn-
0x2087d039eaa7dfc8,host_traddr=nn-0x20000024ff752e6c:pn-
0x21000024ff752e6c live non-optimized
+- nvme3 fc traddr=nn-0x2082d039eaa7dfc8:pn-
0x2083d039eaa7dfc8,host_traddr=nn-0x20000024ff752e6c:pn-
0x21000024ff752e6c live optimized
```

NVMe/TCP

```
nvme list-subsys /dev/nvme1n1
```

Show example

```
nvme-subsys5 - NQN=nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme
_tcp_3
hostnqn=nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0035-5910-804b-
b5c04f444d33
iopolicy=round-robin
\
+- nvme13 tcp
traddr=192.168.2.25,trsvcid=4420,host_traddr=192.168.2.31,
src_addr=192.168.2.31 live optimized
+- nvme14 tcp
traddr=192.168.2.24,trsvcid=4420,host_traddr=192.168.2.31,
src_addr=192.168.2.31 live non-optimized
+- nvme5 tcp
traddr=192.168.1.25,trsvcid=4420,host_traddr=192.168.1.31,
src_addr=192.168.1.31 live optimized
+- nvme6 tcp
traddr=192.168.1.24,trsvcid=4420,host_traddr=192.168.1.31,
src_addr=192.168.1.31 live non-optimized
```

5. Verify that the NetApp plug-in displays the correct values for each ONTAP namespace device:

Column

```
nvme netapp ontapdevices -o column
```

Show example

```
Device        Vserver   Namespace Path
----------------------- ------------------------------
/dev/nvme1n1     linux_tcnvme_iscsi
/vol/tcpnvme_1_0_0/tcpnvme_ns

NSID       UUID                                   Size
------------------------------------------------------------
1    5f7f630d-8ea5-407f-a490-484b95b15dd6   21.47GB
```

JSON

```
nvme netapp ontapdevices -o json
```

Show example


```
{
  "ONTAPdevices":[
    {
      "Device":"/dev/nvme1n1",
      "Vserver":"linux_tcnvme_iscsi",
      "Namespace_Path":"/vol/tcpnvme_1_0_0/tcpnvme_ns",
      "NSID":1,
      "UUID":"5f7f630d-8ea5-407f-a490-484b95b15dd6",
      "Size":"21.47GB",
      "LBA_Data_Size":4096,
      "Namespace_Size":5242880
    },
]
}
```


Step 7: Review the known issues

There are no known issues.

Configure Rocky Linux 9.1 with NVMe-oF for ONTAP storage

Rocky Linux 9.1 hosts support the NVMe/FC and NVMe/TCP protocols with Asymmetric Namespace Access (ANA). ANA is equivalent to asymmetric logical unit access (ALUA) multipathing in iSCSI and FCP environments and is implemented using the in-kernel NVMe multipath feature.

For additional details on supported configurations, see the Interoperability Matrix Tool.

About this task

You can use the following support and features with the NVMe-oF host configuration for Rocky Linux 9.1. You should also review the known limitations before starting the configuration process.

- Support available:
  - Support for NVMe over TCP (NVMe/TCP) in addition to NVMe over Fibre Channel (NVMe/FC). The NetApp plug-in in the native `nvme-cli` package displays ONTAP details for both NVMe/FC and NVMe/TCP namespaces.
  - Running NVMe and SCSI traffic on the same host. For example, you can configure dm-multipath for SCSI mpath devices on SCSI LUNs and use NVMe multipath to configure NVMe-oF namespace devices on the host.
- Features available:
  - Rocky Linux 9.1 enables in-kernel NVMe multipath for NVMe namespaces by default, removing the need for explicit settings.
- Known limitations:
  - SAN booting using the NVMe-oF protocol is currently not supported.

Step 1: Optionally, enable SAN booting

You can configure your host to use SAN booting to simplify deployment and improve scalability.

Before you begin

Use the Interoperability Matrix Tool to verify that your Linux OS, host bus adapter (HBA), HBA firmware, HBA boot BIOS, and ONTAP version support SAN booting.

Steps

1. Create a SAN boot namespace and map it to the host.
2. Enable SAN booting in the server BIOS for the ports to which the SAN boot namespace is mapped.

   For information on how to enable the HBA BIOS, see your vendor-specific documentation.

3. Verify that the configuration was successful by rebooting the host and verifying that the OS is up and running.

Step 2: Validate software versions

Use the following procedure to validate the minimum supported Rocky Linux 9.1 software versions.

Steps

1. Install Rocky Linux 9.1 on the server. After the installation is complete, verify that you are running the specified Rocky Linux 9.1 kernel:

```
uname -r
```

The following example shows a Rocky Linux kernel version:

```
5.14.0-570.12.1.el9_6.x86_64
```

2. Install the `nvme-cli` package:

```
rpm -qa|grep nvme-cli
```

The following example shows an nvme-cli package version:

```
nvme-cli-2.11-5.el9.x86_64
```

3. Install the `libnvme` package:

```
rpm -qa|grep libnvme
```

The following example shows an `libnvme` package version:

```
libnvme-1.11.1-1.el9.x86_64
```

4. On the Rocky Linux host, check the hostnqn string at `/etc/nvme/hostnqn`:

```
cat /etc/nvme/hostnqn
```

The following example shows an `hostnqn` version:

```
nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
```

5. Verify that the `hostnqn` string matches the `hostnqn` string for the corresponding subsystem on the ONTAP array:

```
::> vserver nvme subsystem host show -vserver vs_coexistence_LPE36002
```

Show example

```
Vserver Subsystem Priority  Host NQN
------- --------- --------
------------------------------------------------
vs_coexistence_LPE36002
        nvme
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
        nvme_1
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
        nvme_2
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
        nvme_3
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
4 entries were displayed.
```

ⓘ If the `hostnqn` strings do not match, use the `vserver modify` command to update the `hostnqn` string on your corresponding ONTAP array subsystem to match the `hostnqn` string from `/etc/nvme/hostnqn` on the host.

Step 3: Configure NVMe/FC

You can configure NVMe/FC with Broadcom/Emulex FC or Marvell/Qlogic FC adapters. You also need to manually discover the NVMe/TCP subsystems and namespaces.

Broadcom/Emulex

Configure NVMe/FC for a Broadcom/Emulex adapter.

Steps

1. Verify that you are using the supported adapter model:

   a. Display the model names:

   ```
   cat /sys/class/scsi_host/host*/modelname
   ```

   You should see the following output:

   ```
   LPe36002-M64
   LPe36002-M64
   ```

   b. Display the model descriptions:

   ```
   cat /sys/class/scsi_host/host*/modeldesc
   ```

   You should see an output similar to the following example:

   ```
   Emulex LightPulse LPe36002-M64 2-Port 64Gb Fibre Channel Adapter
   Emulex LightPulse LPe36002-M64 2-Port 64Gb Fibre Channel Adapter
   ```

2. Verify that you are using the recommended Broadcom `lpfc` firmware and inbox driver:

   a. Display the firmware version:

   ```
   cat /sys/class/scsi_host/host*/fwrev
   ```

   The following example shows firmware versions:

   ```
   14.4.317.10, sli-4:6:d
   14.4.317.10, sli-4:6:d
   ```

   b. Display the inbox driver version:

   ```
   cat /sys/module/lpfc/version`
   ```

   The following example shows a driver version:

```
0:14.4.0.2
```

For the current list of supported adapter driver and firmware versions, see the Interoperability Matrix Tool.

3. Verify that the expected output of `lpfc_enable_fc4_type` is set to `3`:

```
cat /sys/module/lpfc/parameters/lpfc_enable_fc4_type
```

4. Verify that you can view your initiator ports:

```
cat /sys/class/fc_host/host*/port_name
```

The following example shows port identities:

```
0x100000109bf044b1
0x100000109bf044b2
```

5. Verify that your initiator ports are online:

```
cat /sys/class/fc_host/host*/port_state
```

You should see the following output:

```
Online
Online
```

6. Verify that the NVMe/FC initiator ports are enabled and that the target ports are visible:

```
cat /sys/class/scsi_host/host*/nvme_info
```

Show example

```
NVME Initiator Enabled
XRI Dist lpfc2 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc2 WWPN x100000109bf044b1 WWNN x200000109bf044b1
DID x022a00 ONLINE
NVME RPORT       WWPN x202fd039eaa7dfc8 WWNN x202cd039eaa7dfc8
DID x021310 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x202dd039eaa7dfc8 WWNN x202cd039eaa7dfc8
DID x020b10 TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 0000000810 Cmpl 0000000810 Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000007b098f07 Issue 000000007aee27c4 OutIO
fffffffffe498bd
        abort 000013b4 noxri 00000000 nondlp 00000058 qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 000013b4 Err 00021443

NVME Initiator Enabled
XRI Dist lpfc3 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc3 WWPN x100000109bf044b2 WWNN x200000109bf044b2
DID x021b00 ONLINE
NVME RPORT       WWPN x2033d039eaa7dfc8 WWNN x202cd039eaa7dfc8
DID x020110 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2032d039eaa7dfc8 WWNN x202cd039eaa7dfc8
DID x022910 TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 0000000840 Cmpl 0000000840 Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000007afd4434 Issue 000000007ae31b83 OutIO
fffffffffe5d74f
        abort 000014a5 noxri 00000000 nondlp 0000006a qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 000014a5 Err 0002149a
```

Marvell/QLogic

Configure NVMe/FC for a Marvell/QLogic adapter.

> ⓘ The native inbox qla2xxx driver included in the Rocky Linux kernel has the latest fixes. These fixes are essential for ONTAP support.

Steps

1. Verify that you are running the supported adapter driver and firmware versions:

```
cat /sys/class/fc_host/host*/symbolic_name
```

The follow example shows driver and firmware versions:

```
QLE2742 FW:v9.14.00 DVR:v10.02.09.200-k
QLE2742 FW:v9.14.00 DVR:v10.02.09.200-k
```

2. Verify that `ql2xnvmeenable` is set. This enables the Marvell adapter to function as an NVMe/FC initiator:

```
cat /sys/module/qla2xxx/parameters/ql2xnvmeenable
```

The expected output is 1.

Step 4: Optionally, enable 1MB I/O

You can enable I/O requests of size 1MB for NVMe/FC configured with a Broadcom adapter. ONTAP reports a Max Data Transfer Size (MDTS) of 8 in the Identify Controller data. This means the maximum I/O request size can be up to 1MB. To issue I/O requests of size 1MB, you need to increase the lpfc value of the `lpfc_sg_seg_cnt` parameter to 256 from the default value of 64.

> These steps don't apply to Qlogic NVMe/FC hosts.

Steps

1. Set the `lpfc_sg_seg_cnt` parameter to 256:

```
cat /etc/modprobe.d/lpfc.conf
```

```
options lpfc lpfc_sg_seg_cnt=256
```

2. Run the `dracut -f` command, and reboot the host.
3. Verify that the value for `lpfc_sg_seg_cnt` is 256:

```
cat /sys/module/lpfc/parameters/lpfc_sg_seg_cnt
```

Step 5: Configure NVMe/TCP

The NVMe/TCP protocol doesn't support the auto-connect operation. Instead, you can discover the NVMe/TCP subsystems and namespaces by performing the NVMe/TCP connect or connect-all operations manually.

Steps

1. Verify that the initiator port can fetch the discovery log page data across the supported NVMe/TCP LIFs:

   ```
   nvme discover -t tcp -w host-traddr -a traddr
   ```

Show example

```
nvme discover -t tcp -w 192.168.1.31 -a 192.168.1.24
Discovery Log Number of Records 20, Generation counter 25
=====Discovery Log Entry 0======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  4
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr:  192.168.2.25
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 1======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  2
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr:  192.168.1.25
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 2======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  5
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr:  192.168.2.24
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 3======
trtype:  tcp
adrfam:  ipv4
```

```
subtype: current discovery subsystem
treq:    not specified
portid:  1
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr:  192.168.1.24
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 4======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_1
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 5======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_1
traddr:  192.168.1.25
eflags:  none
sectype: none
=====Discovery Log Entry 6======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  5
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_1
```

```
traddr:  192.168.2.24
eflags:  none
sectype: none
=====Discovery Log Entry 7======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_1
traddr:  192.168.1.24
eflags:  none
sectype: none
=====Discovery Log Entry 8======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_4
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 9======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_4
traddr:  192.168.1.25
eflags:  none
sectype: none
=====Discovery Log Entry 10======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
```

```
treq:    not specified
portid:  5
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_4
traddr:  192.168.2.24
eflags:  none
sectype: none
=====Discovery Log Entry 11======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_4
traddr:  192.168.1.24
eflags:  none
sectype: none
=====Discovery Log Entry 12======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_3
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 13======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_3
traddr:  192.168.1.25
```

```
eflags:  none
sectype: none
=====Discovery Log Entry 14======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  5
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_3
traddr:  192.168.2.24
eflags:  none
sectype: none
=====Discovery Log Entry 15======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_3
traddr:  192.168.1.24
eflags:  none
sectype: none
=====Discovery Log Entry 16======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_2
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 17======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
```

```
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_2
traddr:  192.168.1.25
eflags:  none
sectype: none
=====Discovery Log Entry 18======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  5
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_2
traddr:  192.168.2.24
eflags:  none
sectype: none
=====Discovery Log Entry 19======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_2
traddr:  192.168.1.24
eflags:  none
sectype: none
```

2. Verify that the other NVMe/TCP initiator-target LIF combinations are able to successfully fetch discovery log page data:

```
nvme discover -t tcp -w host-traddr -a traddr
```

Show example

```
nvme discover -t tcp -w 192.168.1.31 -a 192.168.1.24
nvme discover -t tcp -w 192.168.2.31 -a 192.168.2.24
nvme discover -t tcp -w 192.168.1.31 -a 192.168.1.25
nvme discover -t tcp -w 192.168.2.31 -a 192.168.2.25
```

3. Run the `nvme connect-all` command across all the supported NVMe/TCP initiator-target LIFs across the nodes:

```
nvme connect-all -t tcp -w host-traddr -a traddr
```

Show example

```
nvme    connect-all -t  tcp -w  192.168.1.31    -a  192.168.1.24
nvme    connect-all -t  tcp -w  192.168.2.31    -a  192.168.2.24
nvme    connect-all -t  tcp -w  192.168.1.31    -a  192.168.1.25
nvme    connect-all -t  tcp -w  192.168.2.31    -a  192.168.2.25
```

Step 6: Validate NVMe-oF

Verify that the in-kernel NVMe multipath status, ANA status, and ONTAP namespaces are correct for the NVMe-oF configuration.

Steps

1. Verify that the in-kernel NVMe multipath is enabled:

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

You should see the following output:

```
Y
```

2. Verify that the appropriate NVMe-oF settings (such as, model set to NetApp ONTAP Controller and load balancing iopolicy set to round-robin) for the respective ONTAP namespaces correctly reflect on the host:

   a. Display the subsystems:

```
cat /sys/class/nvme-subsystem/nvme-subsys*/model
```

You should see the following output:

```
NetApp ONTAP Controller
NetApp ONTAP Controller
```

b. Display the policy:

```
cat /sys/class/nvme-subsystem/nvme-subsys*/iopolicy
```

You should see the following output:

```
round-robin
round-robin
```

3. Verify that the namespaces are created and correctly discovered on the host:

```
nvme list
```

Show example

```
Node         SN                   Model
---------------------------------------------------------
/dev/nvme4n1 81Ix2BVuekWcAAAAAAB   NetApp ONTAP Controller


Namespace Usage    Format             FW             Rev
-----------------------------------------------------------
1                 21.47 GB / 21.47 GB   4 KiB + 0 B   FFFFFFFF
```

4. Verify that the controller state of each path is live and has the correct ANA status:

NVMe/FC

```
nvme list-subsys /dev/nvme4n5
```

Show example

```
nvme-subsys4 - NQN=nqn.1992-
08.com.netapp:sn.3a5d31f5502c11ef9f50d039eab6cb6d:subsystem.nvme
_1
               hostnqn=nqn.2014-08.org.nvmexpress:uuid:e6dade64-
216d-
11ec-b7bb-7ed30a5482c3
iopolicy=round-robin\
+- nvme1 fc traddr=nn-0x2082d039eaa7dfc8:pn-
0x2088d039eaa7dfc8,host_traddr=nn-0x20000024ff752e6d:pn-
0x21000024ff752e6d live optimized
+- nvme12 fc traddr=nn-0x2082d039eaa7dfc8:pn-
0x208ad039eaa7dfc8,host_traddr=nn-0x20000024ff752e6d:pn-
0x21000024ff752e6d live non-optimized
+- nvme10 fc traddr=nn-0x2082d039eaa7dfc8:pn-
0x2087d039eaa7dfc8,host_traddr=nn-0x20000024ff752e6c:pn-
0x21000024ff752e6c live non-optimized
+- nvme3 fc traddr=nn-0x2082d039eaa7dfc8:pn-
0x2083d039eaa7dfc8,host_traddr=nn-0x20000024ff752e6c:pn-
0x21000024ff752e6c live optimized
```

NVMe/TCP

```
nvme list-subsys /dev/nvme1n1
```

Show example

```
nvme-subsys5 - NQN=nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme
_tcp_3
hostnqn=nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0035-5910-804b-
b5c04f444d33
iopolicy=round-robin
\
+- nvme13 tcp
traddr=192.168.2.25,trsvcid=4420,host_traddr=192.168.2.31,
src_addr=192.168.2.31 live optimized
+- nvme14 tcp
traddr=192.168.2.24,trsvcid=4420,host_traddr=192.168.2.31,
src_addr=192.168.2.31 live non-optimized
+- nvme5 tcp
traddr=192.168.1.25,trsvcid=4420,host_traddr=192.168.1.31,
src_addr=192.168.1.31 live optimized
+- nvme6 tcp
traddr=192.168.1.24,trsvcid=4420,host_traddr=192.168.1.31,
src_addr=192.168.1.31 live non-optimized
```

5. Verify that the NetApp plug-in displays the correct values for each ONTAP namespace device:

Column

```
nvme netapp ontapdevices -o column
```

Show example

```
Device        Vserver   Namespace Path
----------------------- ------------------------------
/dev/nvme1n1     linux_tcnvme_iscsi
/vol/tcpnvme_1_0_0/tcpnvme_ns

NSID       UUID                                   Size
------------------------------------------------------------
1    5f7f630d-8ea5-407f-a490-484b95b15dd6   21.47GB
```

JSON

```
nvme netapp ontapdevices -o json
```

Show example


```
{
  "ONTAPdevices":[
    {
      "Device":"/dev/nvme1n1",
      "Vserver":"linux_tcnvme_iscsi",
      "Namespace_Path":"/vol/tcpnvme_1_0_0/tcpnvme_ns",
      "NSID":1,
      "UUID":"5f7f630d-8ea5-407f-a490-484b95b15dd6",
      "Size":"21.47GB",
      "LBA_Data_Size":4096,
      "Namespace_Size":5242880
    },
]
}
```


Step 7: Review the known issues

The NVMe-oF host configuration for Rocky Linux 9.1 with ONTAP storage storage has the following known issue:

NetApp Bug ID	Title	Description
1503468	`nvme list-subsys` command returns repeated NVMe controllers listed for a given subsystem	The `nvme list-subsys` command should return a unique list of NVMe controllers associated with a given subsystem. In Rocky Linux 9.1, the `nvme list-subsys` command returns NVMe controllers with their respective ANA state for all namespaces that belong to a given subsystem. However, the ANA state is a per-namespace attribute, therefore, unique NVMe controller entries with the path state should be displayed if you list the subsystem command syntax for a given namespace.

Configure Rocky Linux 9.0 with NVMe-oF for ONTAP storage

Rocky Linux 9.0 hosts support the NVMe/FC and NVMe/TCP protocols with Asymmetric Namespace Access (ANA). ANA is equivalent to asymmetric logical unit access (ALUA) multipathing in iSCSI and FCP environments and is implemented using the in-kernel NVMe multipath feature.

For additional details on supported configurations, see the Interoperability Matrix Tool.

About this task

You can use the following support and features with the NVMe-oF host configuration for Rocky Linux 9.0. You should also review the known limitations before starting the configuration process.

- Support available:
  - Support for NVMe over TCP (NVMe/TCP) in addition to NVMe over Fibre Channel (NVMe/FC). The NetApp plug-in in the native `nvme-cli` package displays ONTAP details for both NVMe/FC and NVMe/TCP namespaces.
  - Running NVMe and SCSI traffic on the same host. For example, you can configure dm-multipath for SCSI mpath devices on SCSI LUNs and use NVMe multipath to configure NVMe-oF namespace devices on the host.
- Features available:
  - Rocky Linux 9.0 enables in-kernel NVMe multipath for NVMe namespaces by default, removing the need for explicit settings.
- Known limitations:
  - SAN booting using the NVMe-oF protocol is currently not supported.

Step 1: Optionally, enable SAN booting

You can configure your host to use SAN booting to simplify deployment and improve scalability.

Before you begin

Use the Interoperability Matrix Tool to verify that your Linux OS, host bus adapter (HBA), HBA firmware, HBA boot BIOS, and ONTAP version support SAN booting.

Steps

1. Create a SAN boot namespace and map it to the host.
2. Enable SAN booting in the server BIOS for the ports to which the SAN boot namespace is mapped.

   For information on how to enable the HBA BIOS, see your vendor-specific documentation.

3. Verify that the configuration was successful by rebooting the host and verifying that the OS is up and running.

Step 2: Validate software versions

Use the following procedure to validate the minimum supported Rocky Linux 9.0 software versions.

Steps

1. Install Rocky Linux 9.0 on the server. After the installation is complete, verify that you are running the specified Rocky Linux 9.0 kernel:

   ```
   uname -r
   ```

   The following example shows a Rocky Linux kernel version:

   ```
   5.14.0-570.12.1.el9_6.x86_64
   ```

2. Install the `nvme-cli` package:

   ```
   rpm -qa|grep nvme-cli
   ```

   The following example shows an nvme-cli package version:

   ```
   nvme-cli-2.11-5.el9.x86_64
   ```

3. Install the `libnvme` package:

   ```
   rpm -qa|grep libnvme
   ```

   The following example shows an `libnvme` package version:

   ```
   libnvme-1.11.1-1.el9.x86_64
   ```

4. On the Rocky Linux host, check the hostnqn string at `/etc/nvme/hostnqn`:

```
cat /etc/nvme/hostnqn
```

The following example shows an `hostnqn` version:

```
nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
```

5. Verify that the `hostnqn` string matches the `hostnqn` string for the corresponding subsystem on the ONTAP array:

```
::> vserver nvme subsystem host show -vserver vs_coexistence_LPE36002
```

Show example

```
Vserver Subsystem Priority  Host NQN
------- --------- --------
------------------------------------------------
vs_coexistence_LPE36002
        nvme
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
        nvme_1
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
        nvme_2
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
        nvme_3
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
4 entries were displayed.
```

If the `hostnqn` strings do not match, use the `vserver modify` command to update the `hostnqn` string on your corresponding ONTAP array subsystem to match the `hostnqn` string from `/etc/nvme/hostnqn` on the host.

Step 3: Configure NVMe/FC

You can configure NVMe/FC with Broadcom/Emulex FC or Marvell/Qlogic FC adapters. You also need to manually discover the NVMe/TCP subsystems and namespaces.

Broadcom/Emulex

Configure NVMe/FC for a Broadcom/Emulex adapter.

Steps

1. Verify that you are using the supported adapter model:

   a. Display the model names:

   ```
   cat /sys/class/scsi_host/host*/modelname
   ```

   You should see the following output:

   ```
   LPe36002-M64
   LPe36002-M64
   ```

   b. Display the model descriptions:

   ```
   cat /sys/class/scsi_host/host*/modeldesc
   ```

   You should see an output similar to the following example:

   ```
   Emulex LightPulse LPe36002-M64 2-Port 64Gb Fibre Channel Adapter
   Emulex LightPulse LPe36002-M64 2-Port 64Gb Fibre Channel Adapter
   ```

2. Verify that you are using the recommended Broadcom `lpfc` firmware and inbox driver:

   a. Display the firmware version:

   ```
   cat /sys/class/scsi_host/host*/fwrev
   ```

   The following example shows firmware versions:

   ```
   14.4.317.10, sli-4:6:d
   14.4.317.10, sli-4:6:d
   ```

   b. Display the inbox driver version:

   ```
   cat /sys/module/lpfc/version`
   ```

   The following example shows a driver version:

```
0:14.4.0.2
```

For the current list of supported adapter driver and firmware versions, see the Interoperability Matrix Tool.

3. Verify that the expected output of `lpfc_enable_fc4_type` is set to `3`:

   ```
   cat /sys/module/lpfc/parameters/lpfc_enable_fc4_type
   ```

4. Verify that you can view your initiator ports:

   ```
   cat /sys/class/fc_host/host*/port_name
   ```

   The following example shows port identities:

   ```
   0x100000109bf044b1
   0x100000109bf044b2
   ```

5. Verify that your initiator ports are online:

   ```
   cat /sys/class/fc_host/host*/port_state
   ```

   You should see the following output:

   ```
   Online
   Online
   ```

6. Verify that the NVMe/FC initiator ports are enabled and that the target ports are visible:

   ```
   cat /sys/class/scsi_host/host*/nvme_info
   ```

Show example

```
NVME Initiator Enabled
XRI Dist lpfc2 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc2 WWPN x100000109bf044b1 WWNN x200000109bf044b1
DID x022a00 ONLINE
NVME RPORT       WWPN x202fd039eaa7dfc8 WWNN x202cd039eaa7dfc8
DID x021310 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x202dd039eaa7dfc8 WWNN x202cd039eaa7dfc8
DID x020b10 TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 0000000810 Cmpl 0000000810 Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000007b098f07 Issue 000000007aee27c4 OutIO
fffffffffe498bd
        abort 000013b4 noxri 00000000 nondlp 00000058 qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 000013b4 Err 00021443

NVME Initiator Enabled
XRI Dist lpfc3 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc3 WWPN x100000109bf044b2 WWNN x200000109bf044b2
DID x021b00 ONLINE
NVME RPORT       WWPN x2033d039eaa7dfc8 WWNN x202cd039eaa7dfc8
DID x020110 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2032d039eaa7dfc8 WWNN x202cd039eaa7dfc8
DID x022910 TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 0000000840 Cmpl 0000000840 Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000007afd4434 Issue 000000007ae31b83 OutIO
fffffffffe5d74f
        abort 000014a5 noxri 00000000 nondlp 0000006a qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 000014a5 Err 0002149a
```

Marvell/QLogic

Configure NVMe/FC for a Marvell/QLogic adapter.

> ⓘ The native inbox qla2xxx driver included in the Rocky Linux kernel has the latest fixes. These fixes are essential for ONTAP support.

Steps

1. Verify that you are running the supported adapter driver and firmware versions:

```
cat /sys/class/fc_host/host*/symbolic_name
```

The follow example shows driver and firmware versions:

```
QLE2742 FW:v9.14.00 DVR:v10.02.09.200-k
QLE2742 FW:v9.14.00 DVR:v10.02.09.200-k
```

2. Verify that `ql2xnvmeenable` is set. This enables the Marvell adapter to function as an NVMe/FC initiator:

```
cat /sys/module/qla2xxx/parameters/ql2xnvmeenable
```

The expected output is 1.

Step 4: Optionally, enable 1MB I/O

You can enable I/O requests of size 1MB for NVMe/FC configured with a Broadcom adapter. ONTAP reports a Max Data Transfer Size (MDTS) of 8 in the Identify Controller data. This means the maximum I/O request size can be up to 1MB. To issue I/O requests of size 1MB, you need to increase the lpfc value of the `lpfc_sg_seg_cnt` parameter to 256 from the default value of 64.

> These steps don't apply to Qlogic NVMe/FC hosts.

Steps

1. Set the `lpfc_sg_seg_cnt` parameter to 256:

```
cat /etc/modprobe.d/lpfc.conf
```

```
options lpfc lpfc_sg_seg_cnt=256
```

2. Run the `dracut -f` command, and reboot the host.
3. Verify that the value for `lpfc_sg_seg_cnt` is 256:

```
cat /sys/module/lpfc/parameters/lpfc_sg_seg_cnt
```

Step 5: Configure NVMe/TCP

The NVMe/TCP protocol doesn’t support the auto-connect operation. Instead, you can discover the NVMe/TCP subsystems and namespaces by performing the NVMe/TCP connect or connect-all operations manually.

Steps

1. Verify that the initiator port can fetch the discovery log page data across the supported NVMe/TCP LIFs:

```
nvme discover -t tcp -w host-traddr -a traddr
```

Show example

```
nvme discover -t tcp -w 192.168.1.31 -a 192.168.1.24
Discovery Log Number of Records 20, Generation counter 25
=====Discovery Log Entry 0======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  4
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr:  192.168.2.25
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 1======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  2
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr:  192.168.1.25
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 2======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  5
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr:  192.168.2.24
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 3======
trtype:  tcp
adrfam:  ipv4
```

```
subtype: current discovery subsystem
treq:    not specified
portid:  1
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr:  192.168.1.24
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 4======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_1
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 5======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_1
traddr:  192.168.1.25
eflags:  none
sectype: none
=====Discovery Log Entry 6======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  5
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_1
```

```
traddr:  192.168.2.24
eflags:  none
sectype: none
=====Discovery Log Entry 7======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_1
traddr:  192.168.1.24
eflags:  none
sectype: none
=====Discovery Log Entry 8======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_4
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 9======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_4
traddr:  192.168.1.25
eflags:  none
sectype: none
=====Discovery Log Entry 10======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
```

```
treq:    not specified
portid:  5
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_4
traddr:  192.168.2.24
eflags:  none
sectype: none
=====Discovery Log Entry 11======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_4
traddr:  192.168.1.24
eflags:  none
sectype: none
=====Discovery Log Entry 12======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_3
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 13======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_3
traddr:  192.168.1.25
```

```
eflags:  none
sectype: none
=====Discovery Log Entry 14======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  5
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_3
traddr:  192.168.2.24
eflags:  none
sectype: none
=====Discovery Log Entry 15======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_3
traddr:  192.168.1.24
eflags:  none
sectype: none
=====Discovery Log Entry 16======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_2
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 17======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
```

```
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_2
traddr:  192.168.1.25
eflags:  none
sectype: none
=====Discovery Log Entry 18======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  5
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_2
traddr:  192.168.2.24
eflags:  none
sectype: none
=====Discovery Log Entry 19======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_2
traddr:  192.168.1.24
eflags:  none
sectype: none
```

2. Verify that the other NVMe/TCP initiator-target LIF combinations are able to successfully fetch discovery log page data:

```
nvme discover -t tcp -w host-traddr -a traddr
```

Show example

```
nvme discover -t tcp -w 192.168.1.31 -a 192.168.1.24
nvme discover -t tcp -w 192.168.2.31 -a 192.168.2.24
nvme discover -t tcp -w 192.168.1.31 -a 192.168.1.25
nvme discover -t tcp -w 192.168.2.31 -a 192.168.2.25
```

3. Run the `nvme connect-all` command across all the supported NVMe/TCP initiator-target LIFs across the nodes:

```
nvme connect-all -t tcp -w host-traddr -a traddr
```

Show example

```
nvme    connect-all -t  tcp -w  192.168.1.31    -a  192.168.1.24
nvme    connect-all -t  tcp -w  192.168.2.31    -a  192.168.2.24
nvme    connect-all -t  tcp -w  192.168.1.31    -a  192.168.1.25
nvme    connect-all -t  tcp -w  192.168.2.31    -a  192.168.2.25
```

Step 6: Validate NVMe-oF

Verify that the in-kernel NVMe multipath status, ANA status, and ONTAP namespaces are correct for the NVMe-oF configuration.

Steps

1. Verify that the in-kernel NVMe multipath is enabled:

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

You should see the following output:

```
Y
```

2. Verify that the appropriate NVMe-oF settings (such as, model set to NetApp ONTAP Controller and load balancing iopolicy set to round-robin) for the respective ONTAP namespaces correctly reflect on the host:
   a. Display the subsystems:

```
cat /sys/class/nvme-subsystem/nvme-subsys*/model
```

You should see the following output:

```
NetApp ONTAP Controller
NetApp ONTAP Controller
```

b. Display the policy:

```
cat /sys/class/nvme-subsystem/nvme-subsys*/iopolicy
```

You should see the following output:

```
round-robin
round-robin
```

3. Verify that the namespaces are created and correctly discovered on the host:

```
nvme list
```

Show example

```
Node         SN                   Model
---------------------------------------------------------
/dev/nvme4n1 81Ix2BVuekWcAAAAAAAB   NetApp ONTAP Controller


Namespace Usage    Format             FW             Rev
-----------------------------------------------------------
1                 21.47 GB / 21.47 GB   4 KiB + 0 B   FFFFFFFF
```

4. Verify that the controller state of each path is live and has the correct ANA status:

NVMe/FC

```
nvme list-subsys /dev/nvme4n5
```

Show example

```
nvme-subsys4 - NQN=nqn.1992-
08.com.netapp:sn.3a5d31f5502c11ef9f50d039eab6cb6d:subsystem.nvme
_1
               hostnqn=nqn.2014-08.org.nvmexpress:uuid:e6dade64-
216d-
11ec-b7bb-7ed30a5482c3
iopolicy=round-robin\
+- nvme1 fc traddr=nn-0x2082d039eaa7dfc8:pn-
0x2088d039eaa7dfc8,host_traddr=nn-0x20000024ff752e6d:pn-
0x21000024ff752e6d live optimized
+- nvme12 fc traddr=nn-0x2082d039eaa7dfc8:pn-
0x208ad039eaa7dfc8,host_traddr=nn-0x20000024ff752e6d:pn-
0x21000024ff752e6d live non-optimized
+- nvme10 fc traddr=nn-0x2082d039eaa7dfc8:pn-
0x2087d039eaa7dfc8,host_traddr=nn-0x20000024ff752e6c:pn-
0x21000024ff752e6c live non-optimized
+- nvme3 fc traddr=nn-0x2082d039eaa7dfc8:pn-
0x2083d039eaa7dfc8,host_traddr=nn-0x20000024ff752e6c:pn-
0x21000024ff752e6c live optimized
```

NVMe/TCP

```
nvme list-subsys /dev/nvme1n1
```

Show example

```
nvme-subsys5 - NQN=nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme
_tcp_3
hostnqn=nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0035-5910-804b-
b5c04f444d33
iopolicy=round-robin
\
+- nvme13 tcp
traddr=192.168.2.25,trsvcid=4420,host_traddr=192.168.2.31,
src_addr=192.168.2.31 live optimized
+- nvme14 tcp
traddr=192.168.2.24,trsvcid=4420,host_traddr=192.168.2.31,
src_addr=192.168.2.31 live non-optimized
+- nvme5 tcp
traddr=192.168.1.25,trsvcid=4420,host_traddr=192.168.1.31,
src_addr=192.168.1.31 live optimized
+- nvme6 tcp
traddr=192.168.1.24,trsvcid=4420,host_traddr=192.168.1.31,
src_addr=192.168.1.31 live non-optimized
```

5. Verify that the NetApp plug-in displays the correct values for each ONTAP namespace device:

Column

```
nvme netapp ontapdevices -o column
```

Show example

```
Device        Vserver   Namespace Path
----------------------- ------------------------------
/dev/nvme1n1     linux_tcnvme_iscsi
/vol/tcpnvme_1_0_0/tcpnvme_ns

NSID       UUID                                   Size
------------------------------------------------------------
1    5f7f630d-8ea5-407f-a490-484b95b15dd6   21.47GB
```

JSON

```
nvme netapp ontapdevices -o json
```

Show example

```
{
  "ONTAPdevices":[
    {
      "Device":"/dev/nvme1n1",
      "Vserver":"linux_tcnvme_iscsi",
      "Namespace_Path":"/vol/tcpnvme_1_0_0/tcpnvme_ns",
      "NSID":1,
      "UUID":"5f7f630d-8ea5-407f-a490-484b95b15dd6",
      "Size":"21.47GB",
      "LBA_Data_Size":4096,
      "Namespace_Size":5242880
    },
]
}
```

Step 7: Review the known issues

The NVMe-oF host configuration for Rocky Linux 9.0 with ONTAP storage has the following known issues:

NetApp Bug ID	Title	Description
1479047	Rocky Linux 9.0 NVMe-oF hosts create duplicate Persistent Discovery Controllers	On NVMe over Fabrics (NVMe-oF) hosts, you can use the "nvme discover -p" command to create Persistent Discovery Controllers (PDCs). When this command is used, only one PDC should be created per initiator-target combination. However, if you are running ONTAP 9.10.1 and Rocky Linux 9.0 with an NVMe-oF host, a duplicate PDC is created each time "nvme discover -p" is executed. This leads to unnecessary usage of resources on both the host and the target.

Rocky Linux 8

Configure Rocky Linux 8.10 with NVMe-oF for ONTAP storage

Rocky Linux 8.10 hosts support the NVMe/FC and NVMe/TCP protocols with Asymmetric Namespace Access (ANA). ANA is equivalent to asymmetric logical unit access (ALUA) multipathing in iSCSI and FCP environments and is implemented using the in-kernel NVMe multipath feature.

For additional details on supported configurations, see the Interoperability Matrix Tool.

About this task

You can use the following support and features with the NVMe-oF host configuration for Rocky Linux 8.10. You should also review the known limitations before starting the configuration process.

- Support available:
  - Support for NVMe over TCP (NVMe/TCP) in addition to NVMe over Fibre Channel (NVMe/FC). The NetApp plug-in in the native `nvme-cli` package displays ONTAP details for both NVMe/FC and NVMe/TCP namespaces.
  - Running NVMe and SCSI traffic on the same host. For example, you can configure dm-multipath for SCSI mpath devices on SCSI LUNs and use NVMe multipath to configure NVMe-oF namespace devices on the host.
- Known limitations:
  - In-kernel NVMe multipath is disabled by default for Rocky Linux 8.10 NVMe-oF hosts. Therefore, you need to enable it manually.
  - On Rocky Linux 8.10 hosts, NVMe/TCP is a technology preview feature due to open issues. Refer to the Rocky Linux 8.10 Release Notes for details.
  - SAN booting using the NVMe-oF protocol is currently not supported.

Step 1: Optionally, enable SAN booting

You can configure your host to use SAN booting to simplify deployment and improve scalability.

Before you begin

Use the Interoperability Matrix Tool to verify that your Linux OS, host bus adapter (HBA), HBA firmware, HBA boot BIOS, and ONTAP version support SAN booting.

Steps

1. Create a SAN boot namespace and map it to the host.
2. Enable SAN booting in the server BIOS for the ports to which the SAN boot namespace is mapped.

   For information on how to enable the HBA BIOS, see your vendor-specific documentation.

3. Verify that the configuration was successful by rebooting the host and verifying that the OS is up and running.

Step 2: Validate software versions

Use the following procedure to validate the minimum supported Rocky Linux 8.10 software versions.

Steps

1. Install Rocky Linux 8.10 on the server. After the installation is complete, verify that you are running the specified Rocky Linux 9.6 kernel:

   ```
   uname -r
   ```

   The following example shows a Rocky Linux kernel version:

   ```
   5.14.0-570.12.1.el9_6.x86_64
   ```

2. Install the `nvme-cli` package:

   ```
   rpm -qa|grep nvme-cli
   ```

   The following example shows an nvme-cli package version:

   ```
   nvme-cli-2.11-5.el9.x86_64
   ```

3. Install the `libnvme` package:

   ```
   rpm -qa|grep libnvme
   ```

   The following example shows an `libnvme` package version:

   ```
   libnvme-1.11.1-1.el9.x86_64
   ```

4. On the Rocky Linux host, check the hostnqn string at `/etc/nvme/hostnqn`:

```
cat /etc/nvme/hostnqn
```

The following example shows an `hostnqn` version:

```
nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
```

5. Verify that the `hostnqn` string matches the `hostnqn` string for the corresponding subsystem on the ONTAP array:

```
::> vserver nvme subsystem host show -vserver vs_coexistence_LPE36002
```

Show example

```
Vserver Subsystem Priority  Host NQN
------- --------- --------
------------------------------------------------
vs_coexistence_LPE36002
        nvme
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
        nvme_1
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
        nvme_2
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
        nvme_3
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
4 entries were displayed.
```

> If the `hostnqn` strings do not match, use the `vserver modify` command to update the `hostnqn` string on your corresponding ONTAP array subsystem to match the `hostnqn` string from `/etc/nvme/hostnqn` on the host.

Step 3: Configure NVMe/FC

You can configure NVMe/FC with Broadcom/Emulex FC or Marvell/Qlogic FC adapters. You also need to manually discover the NVMe/TCP subsystems and namespaces.

Broadcom/Emulex

Configure NVMe/FC for a Broadcom/Emulex adapter.

Steps

1. Verify that you are using the supported adapter model:
   a. Display the model names:

      ```
      cat /sys/class/scsi_host/host*/modelname
      ```

      You should see the following output:

      ```
      LPe36002-M64
      LPe36002-M64
      ```

   b. Display the model descriptions:

      ```
      cat /sys/class/scsi_host/host*/modeldesc
      ```

      You should see an output similar to the following example:

      ```
      Emulex LightPulse LPe36002-M64 2-Port 64Gb Fibre Channel Adapter
      Emulex LightPulse LPe36002-M64 2-Port 64Gb Fibre Channel Adapter
      ```

2. Verify that you are using the recommended Broadcom `lpfc` firmware and inbox driver:
   a. Display the firmware version:

      ```
      cat /sys/class/scsi_host/host*/fwrev
      ```

      The following example shows firmware versions:

      ```
      14.4.317.10, sli-4:6:d
      14.4.317.10, sli-4:6:d
      ```

   b. Display the inbox driver version:

      ```
      cat /sys/module/lpfc/version`
      ```

      The following example shows a driver version:

```
0:14.4.0.2
```

For the current list of supported adapter driver and firmware versions, see the Interoperability Matrix Tool.

3. Verify that the expected output of `lpfc_enable_fc4_type` is set to `3`:

   ```
   cat /sys/module/lpfc/parameters/lpfc_enable_fc4_type
   ```

4. Verify that you can view your initiator ports:

   ```
   cat /sys/class/fc_host/host*/port_name
   ```

   The following example shows port identities:

   ```
   0x100000109bf044b1
   0x100000109bf044b2
   ```

5. Verify that your initiator ports are online:

   ```
   cat /sys/class/fc_host/host*/port_state
   ```

   You should see the following output:

   ```
   Online
   Online
   ```

6. Verify that the NVMe/FC initiator ports are enabled and that the target ports are visible:

   ```
   cat /sys/class/scsi_host/host*/nvme_info
   ```

Show example

```
NVME Initiator Enabled
XRI Dist lpfc2 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc2 WWPN x100000109bf044b1 WWNN x200000109bf044b1
DID x022a00 ONLINE
NVME RPORT       WWPN x202fd039eaa7dfc8 WWNN x202cd039eaa7dfc8
DID x021310 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x202dd039eaa7dfc8 WWNN x202cd039eaa7dfc8
DID x020b10 TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 0000000810 Cmpl 0000000810 Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000007b098f07 Issue 000000007aee27c4 OutIO
fffffffffe498bd
        abort 000013b4 noxri 00000000 nondlp 00000058 qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 000013b4 Err 00021443

NVME Initiator Enabled
XRI Dist lpfc3 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc3 WWPN x100000109bf044b2 WWNN x200000109bf044b2
DID x021b00 ONLINE
NVME RPORT       WWPN x2033d039eaa7dfc8 WWNN x202cd039eaa7dfc8
DID x020110 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2032d039eaa7dfc8 WWNN x202cd039eaa7dfc8
DID x022910 TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 0000000840 Cmpl 0000000840 Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000007afd4434 Issue 000000007ae31b83 OutIO
fffffffffe5d74f
        abort 000014a5 noxri 00000000 nondlp 0000006a qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 000014a5 Err 0002149a
```

Marvell/QLogic

Configure NVMe/FC for a Marvell/QLogic adapter.

> ⓘ The native inbox qla2xxx driver included in the Rocky Linux kernel has the latest fixes. These fixes are essential for ONTAP support.

Steps

1. Verify that you are running the supported adapter driver and firmware versions:

```
cat /sys/class/fc_host/host*/symbolic_name
```

The follow example shows driver and firmware versions:

```
QLE2742 FW:v9.14.00 DVR:v10.02.09.200-k
QLE2742 FW:v9.14.00 DVR:v10.02.09.200-k
```

2. Verify that `ql2xnvmeenable` is set. This enables the Marvell adapter to function as an NVMe/FC initiator:

```
cat /sys/module/qla2xxx/parameters/ql2xnvmeenable
```

The expected output is 1.

Step 4: Optionally, enable 1MB I/O

You can enable I/O requests of size 1MB for NVMe/FC configured with a Broadcom adapter. ONTAP reports a Max Data Transfer Size (MDTS) of 8 in the Identify Controller data. This means the maximum I/O request size can be up to 1MB. To issue I/O requests of size 1MB, you need to increase the lpfc value of the `lpfc_sg_seg_cnt` parameter to 256 from the default value of 64.

> These steps don't apply to Qlogic NVMe/FC hosts.

Steps

1. Set the `lpfc_sg_seg_cnt` parameter to 256:

```
cat /etc/modprobe.d/lpfc.conf
```

```
options lpfc lpfc_sg_seg_cnt=256
```

2. Run the `dracut -f` command, and reboot the host.
3. Verify that the value for `lpfc_sg_seg_cnt` is 256:

```
cat /sys/module/lpfc/parameters/lpfc_sg_seg_cnt
```

Step 5: Configure NVMe/TCP

The NVMe/TCP protocol doesn’t support the auto-connect operation. Instead, you can discover the NVMe/TCP subsystems and namespaces by performing the NVMe/TCP connect or connect-all operations manually.

Steps

1. Verify that the initiator port can fetch the discovery log page data across the supported NVMe/TCP LIFs:

```
nvme discover -t tcp -w host-traddr -a traddr
```

Show example

```
nvme discover -t tcp -w 192.168.1.31 -a 192.168.1.24
Discovery Log Number of Records 20, Generation counter 25
=====Discovery Log Entry 0======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  4
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr:  192.168.2.25
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 1======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  2
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr:  192.168.1.25
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 2======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  5
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr:  192.168.2.24
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 3======
trtype:  tcp
adrfam:  ipv4
```

```
subtype: current discovery subsystem
treq:    not specified
portid:  1
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr:  192.168.1.24
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 4======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_1
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 5======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_1
traddr:  192.168.1.25
eflags:  none
sectype: none
=====Discovery Log Entry 6======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  5
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_1
```

```
traddr:  192.168.2.24
eflags:  none
sectype: none
=====Discovery Log Entry 7======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_1
traddr:  192.168.1.24
eflags:  none
sectype: none
=====Discovery Log Entry 8======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_4
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 9======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_4
traddr:  192.168.1.25
eflags:  none
sectype: none
=====Discovery Log Entry 10======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
```

```
treq:    not specified
portid:  5
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_4
traddr:  192.168.2.24
eflags:  none
sectype: none
=====Discovery Log Entry 11======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_4
traddr:  192.168.1.24
eflags:  none
sectype: none
=====Discovery Log Entry 12======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_3
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 13======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_3
traddr:  192.168.1.25
```

```
eflags:  none
sectype: none
=====Discovery Log Entry 14======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  5
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_3
traddr:  192.168.2.24
eflags:  none
sectype: none
=====Discovery Log Entry 15======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_3
traddr:  192.168.1.24
eflags:  none
sectype: none
=====Discovery Log Entry 16======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_2
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 17======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
```

```
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_2
traddr:  192.168.1.25
eflags:  none
sectype: none
=====Discovery Log Entry 18======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  5
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_2
traddr:  192.168.2.24
eflags:  none
sectype: none
=====Discovery Log Entry 19======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_2
traddr:  192.168.1.24
eflags:  none
sectype: none
```

2. Verify that the other NVMe/TCP initiator-target LIF combinations are able to successfully fetch discovery log page data:

```
nvme discover -t tcp -w host-traddr -a traddr
```

Show example

```
nvme discover -t tcp -w 192.168.1.31 -a 192.168.1.24
nvme discover -t tcp -w 192.168.2.31 -a 192.168.2.24
nvme discover -t tcp -w 192.168.1.31 -a 192.168.1.25
nvme discover -t tcp -w 192.168.2.31 -a 192.168.2.25
```

3. Run the `nvme connect-all` command across all the supported NVMe/TCP initiator-target LIFs across the nodes:

```
nvme connect-all -t tcp -w host-traddr -a traddr
```

Show example

```
nvme    connect-all -t  tcp -w  192.168.1.31    -a  192.168.1.24
nvme    connect-all -t  tcp -w  192.168.2.31    -a  192.168.2.24
nvme    connect-all -t  tcp -w  192.168.1.31    -a  192.168.1.25
nvme    connect-all -t  tcp -w  192.168.2.31    -a  192.168.2.25
```

Step 6: Validate NVMe-oF

Verify that the in-kernel NVMe multipath status, ANA status, and ONTAP namespaces are correct for the NVMe-oF configuration.

Steps

1. Verify that the in-kernel NVMe multipath is enabled:

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

You should see the following output:

```
Y
```

2. Verify that the appropriate NVMe-oF settings (such as, model set to NetApp ONTAP Controller and load balancing iopolicy set to round-robin) for the respective ONTAP namespaces correctly reflect on the host:
   a. Display the subsystems:

```
cat /sys/class/nvme-subsystem/nvme-subsys*/model
```

You should see the following output:

```
NetApp ONTAP Controller
NetApp ONTAP Controller
```

b. Display the policy:

```
cat /sys/class/nvme-subsystem/nvme-subsys*/iopolicy
```

You should see the following output:

```
round-robin
round-robin
```

3. Verify that the namespaces are created and correctly discovered on the host:

```
nvme list
```

Show example

```
Node         SN                   Model
---------------------------------------------------------
/dev/nvme4n1 81Ix2BVuekWcAAAAAAAB   NetApp ONTAP Controller


Namespace Usage    Format             FW            Rev
-----------------------------------------------------------
1                 21.47 GB / 21.47 GB   4 KiB + 0 B   FFFFFFFF
```

4. Verify that the controller state of each path is live and has the correct ANA status:

NVMe/FC

```
nvme list-subsys /dev/nvme4n5
```

Show example

```
nvme-subsys4 - NQN=nqn.1992-
08.com.netapp:sn.3a5d31f5502c11ef9f50d039eab6cb6d:subsystem.nvme
_1
               hostnqn=nqn.2014-08.org.nvmexpress:uuid:e6dade64-
216d-
11ec-b7bb-7ed30a5482c3
iopolicy=round-robin\
+- nvme1 fc traddr=nn-0x2082d039eaa7dfc8:pn-
0x2088d039eaa7dfc8,host_traddr=nn-0x20000024ff752e6d:pn-
0x21000024ff752e6d live optimized
+- nvme12 fc traddr=nn-0x2082d039eaa7dfc8:pn-
0x208ad039eaa7dfc8,host_traddr=nn-0x20000024ff752e6d:pn-
0x21000024ff752e6d live non-optimized
+- nvme10 fc traddr=nn-0x2082d039eaa7dfc8:pn-
0x2087d039eaa7dfc8,host_traddr=nn-0x20000024ff752e6c:pn-
0x21000024ff752e6c live non-optimized
+- nvme3 fc traddr=nn-0x2082d039eaa7dfc8:pn-
0x2083d039eaa7dfc8,host_traddr=nn-0x20000024ff752e6c:pn-
0x21000024ff752e6c live optimized
```

NVMe/TCP

```
nvme list-subsys /dev/nvme1n1
```

Show example

```
nvme-subsys5 - NQN=nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme
_tcp_3
hostnqn=nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0035-5910-804b-
b5c04f444d33
iopolicy=round-robin
\
+- nvme13 tcp
traddr=192.168.2.25,trsvcid=4420,host_traddr=192.168.2.31,
src_addr=192.168.2.31 live optimized
+- nvme14 tcp
traddr=192.168.2.24,trsvcid=4420,host_traddr=192.168.2.31,
src_addr=192.168.2.31 live non-optimized
+- nvme5 tcp
traddr=192.168.1.25,trsvcid=4420,host_traddr=192.168.1.31,
src_addr=192.168.1.31 live optimized
+- nvme6 tcp
traddr=192.168.1.24,trsvcid=4420,host_traddr=192.168.1.31,
src_addr=192.168.1.31 live non-optimized
```

5. Verify that the NetApp plug-in displays the correct values for each ONTAP namespace device:

Column

```
nvme netapp ontapdevices -o column
```

Show example

```
Device        Vserver   Namespace Path
----------------------- ------------------------------
/dev/nvme1n1     linux_tcnvme_iscsi
/vol/tcpnvme_1_0_0/tcpnvme_ns

NSID       UUID                                   Size
------------------------------------------------------------
1    5f7f630d-8ea5-407f-a490-484b95b15dd6   21.47GB
```

JSON

```
nvme netapp ontapdevices -o json
```

Show example

```
{
  "ONTAPdevices":[
    {
      "Device":"/dev/nvme1n1",
      "Vserver":"linux_tcnvme_iscsi",
      "Namespace_Path":"/vol/tcpnvme_1_0_0/tcpnvme_ns",
      "NSID":1,
      "UUID":"5f7f630d-8ea5-407f-a490-484b95b15dd6",
      "Size":"21.47GB",
      "LBA_Data_Size":4096,
      "Namespace_Size":5242880
    },
]
}
```

Step 7: Review the known issues

The NVMe-oF host configuration for Rocky Linux 8.10 with ONTAP has the following known issue:

NetApp Bug ID	Title	Description
1479047	Rocky Linux 8.10 NVMe-oF hosts create duplicate persistent discovery controllers	On NVMe over Fabrics (NVMe-oF) hosts, you can use the "nvme discover -p" command to create Persistent Discovery Controllers (PDCs). When this command is used, only one PDC should be created per initiator-target combination. However, if you are running Rocky Linux 8.10 on an NVMe-oF host, a duplicate PDC is created each time "nvme discover -p" is executed. This leads to unnecessary usage of resources on both the host and the target.

Configure Rocky Linux 8.9 with NVMe-oF for ONTAP storage

Rocky Linux 8.9 hosts support the NVMe/FC and NVMe/TCP protocols with Asymmetric Namespace Access (ANA). ANA is equivalent to asymmetric logical unit access (ALUA) multipathing in iSCSI and FCP environments and is implemented using the in-kernel NVMe multipath feature.

For additional details on supported configurations, see the Interoperability Matrix Tool.

About this task

You can use the following support and features with the NVMe-oF host configuration for Rocky Linux 8.9. You should also review the known limitations before starting the configuration process.

- Support available:
  - Support for NVMe over TCP (NVMe/TCP) in addition to NVMe over Fibre Channel (NVMe/FC). The NetApp plug-in in the native `nvme-cli` package displays ONTAP details for both NVMe/FC and NVMe/TCP namespaces.
  - Running NVMe and SCSI traffic on the same host. For example, you can configure dm-multipath for SCSI mpath devices on SCSI LUNs and use NVMe multipath to configure NVMe-oF namespace devices on the host.
- Known limitations:
  - In-kernel NVMe multipath is disabled by default for Rocky Linux 8.9 NVMe-oF hosts. Therefore, you need to enable it manually.
  - On Rocky Linux 8.9 hosts, NVMe/TCP is a technology preview feature due to open issues. Refer to the Rocky Linux 8.9 Release Notes for details.
  - SAN booting using the NVMe-oF protocol is currently not supported.

Step 1: Optionally, enable SAN booting

You can configure your host to use SAN booting to simplify deployment and improve scalability.

Before you begin

Use the Interoperability Matrix Tool to verify that your Linux OS, host bus adapter (HBA), HBA firmware, HBA boot BIOS, and ONTAP version support SAN booting.

Steps

1. Create a SAN boot namespace and map it to the host.
2. Enable SAN booting in the server BIOS for the ports to which the SAN boot namespace is mapped.

   For information on how to enable the HBA BIOS, see your vendor-specific documentation.
3. Verify that the configuration was successful by rebooting the host and verifying that the OS is up and running.

Step 2: Validate software versions

Use the following procedure to validate the minimum supported Rocky Linux 8.9 software versions.

Steps

1. Install Rocky Linux 8.9 on the server. After the installation is complete, verify that you are running the specified Rocky Linux 8.9 kernel:

   ```
   uname -r
   ```

   The following example shows a Rocky Linux kernel version:

   ```
   5.14.0-570.12.1.el9_6.x86_64
   ```

2. Install the `nvme-cli` package:

   ```
   rpm -qa|grep nvme-cli
   ```

   The following example shows an nvme-cli package version:

   ```
   nvme-cli-2.11-5.el9.x86_64
   ```

3. Install the `libnvme` package:

   ```
   rpm -qa|grep libnvme
   ```

   The following example shows an `libnvme` package version:

   ```
   libnvme-1.11.1-1.el9.x86_64
   ```

4. On the Rocky Linux host, check the hostnqn string at `/etc/nvme/hostnqn`:

   ```
   cat /etc/nvme/hostnqn
   ```

The following example shows an `hostnqn` version:

```
nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
```

5. Verify that the `hostnqn` string matches the `hostnqn` string for the corresponding subsystem on the ONTAP array:

```
::> vserver nvme subsystem host show -vserver vs_coexistence_LPE36002
```

Show example

```
Vserver Subsystem Priority  Host NQN
------- --------- --------
------------------------------------------------
vs_coexistence_LPE36002
        nvme
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
        nvme_1
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
        nvme_2
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
        nvme_3
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
4 entries were displayed.
```

If the `hostnqn` strings do not match, use the `vserver modify` command to update the `hostnqn` string on your corresponding ONTAP array subsystem to match the `hostnqn` string from `/etc/nvme/hostnqn` on the host.

Step 3: Configure NVMe/FC

You can configure NVMe/FC with Broadcom/Emulex FC or Marvell/Qlogic FC adapters. You also need to manually discover the NVMe/TCP subsystems and namespaces.

Broadcom/Emulex

Configure NVMe/FC for a Broadcom/Emulex adapter.

Steps

1. Verify that you are using the supported adapter model:

   a. Display the model names:

   ```
   cat /sys/class/scsi_host/host*/modelname
   ```

   You should see the following output:

   ```
   LPe36002-M64
   LPe36002-M64
   ```

   b. Display the model descriptions:

   ```
   cat /sys/class/scsi_host/host*/modeldesc
   ```

   You should see an output similar to the following example:

   ```
   Emulex LightPulse LPe36002-M64 2-Port 64Gb Fibre Channel Adapter
   Emulex LightPulse LPe36002-M64 2-Port 64Gb Fibre Channel Adapter
   ```

2. Verify that you are using the recommended Broadcom `lpfc` firmware and inbox driver:

   a. Display the firmware version:

   ```
   cat /sys/class/scsi_host/host*/fwrev
   ```

   The following example shows firmware versions:

   ```
   14.4.317.10, sli-4:6:d
   14.4.317.10, sli-4:6:d
   ```

   b. Display the inbox driver version:

   ```
   cat /sys/module/lpfc/version`
   ```

   The following example shows a driver version:

```
0:14.4.0.2
```

For the current list of supported adapter driver and firmware versions, see the Interoperability Matrix Tool.

3. Verify that the expected output of `lpfc_enable_fc4_type` is set to `3`:

   ```
   cat /sys/module/lpfc/parameters/lpfc_enable_fc4_type
   ```

4. Verify that you can view your initiator ports:

   ```
   cat /sys/class/fc_host/host*/port_name
   ```

   The following example shows port identities:

   ```
   0x100000109bf044b1
   0x100000109bf044b2
   ```

5. Verify that your initiator ports are online:

   ```
   cat /sys/class/fc_host/host*/port_state
   ```

   You should see the following output:

   ```
   Online
   Online
   ```

6. Verify that the NVMe/FC initiator ports are enabled and that the target ports are visible:

   ```
   cat /sys/class/scsi_host/host*/nvme_info
   ```

Show example

```
NVME Initiator Enabled
XRI Dist lpfc2 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc2 WWPN x100000109bf044b1 WWNN x200000109bf044b1
DID x022a00 ONLINE
NVME RPORT       WWPN x202fd039eaa7dfc8 WWNN x202cd039eaa7dfc8
DID x021310 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x202dd039eaa7dfc8 WWNN x202cd039eaa7dfc8
DID x020b10 TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 0000000810 Cmpl 0000000810 Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000007b098f07 Issue 000000007aee27c4 OutIO
fffffffffe498bd
        abort 000013b4 noxri 00000000 nondlp 00000058 qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 000013b4 Err 00021443

NVME Initiator Enabled
XRI Dist lpfc3 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc3 WWPN x100000109bf044b2 WWNN x200000109bf044b2
DID x021b00 ONLINE
NVME RPORT       WWPN x2033d039eaa7dfc8 WWNN x202cd039eaa7dfc8
DID x020110 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2032d039eaa7dfc8 WWNN x202cd039eaa7dfc8
DID x022910 TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 0000000840 Cmpl 0000000840 Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000007afd4434 Issue 000000007ae31b83 OutIO
fffffffffe5d74f
        abort 000014a5 noxri 00000000 nondlp 0000006a qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 000014a5 Err 0002149a
```

Marvell/QLogic

Configure NVMe/FC for a Marvell/QLogic adapter.

> The native inbox qla2xxx driver included in the Rocky Linux kernel has the latest fixes. These fixes are essential for ONTAP support.

Steps

1. Verify that you are running the supported adapter driver and firmware versions:

```
cat /sys/class/fc_host/host*/symbolic_name
```

The follow example shows driver and firmware versions:

```
QLE2742 FW:v9.14.00 DVR:v10.02.09.200-k
QLE2742 FW:v9.14.00 DVR:v10.02.09.200-k
```

2. Verify that `ql2xnvmeenable` is set. This enables the Marvell adapter to function as an NVMe/FC initiator:

```
cat /sys/module/qla2xxx/parameters/ql2xnvmeenable
```

The expected output is 1.

Step 4: Optionally, enable 1MB I/O

You can enable I/O requests of size 1MB for NVMe/FC configured with a Broadcom adapter. ONTAP reports a Max Data Transfer Size (MDTS) of 8 in the Identify Controller data. This means the maximum I/O request size can be up to 1MB. To issue I/O requests of size 1MB, you need to increase the lpfc value of the `lpfc_sg_seg_cnt` parameter to 256 from the default value of 64.

These steps don't apply to Qlogic NVMe/FC hosts.

Steps

1. Set the `lpfc_sg_seg_cnt` parameter to 256:

```
cat /etc/modprobe.d/lpfc.conf
```

```
options lpfc lpfc_sg_seg_cnt=256
```

2. Run the `dracut -f` command, and reboot the host.
3. Verify that the value for `lpfc_sg_seg_cnt` is 256:

```
cat /sys/module/lpfc/parameters/lpfc_sg_seg_cnt
```

Step 5: Configure NVMe/TCP

The NVMe/TCP protocol doesn’t support the auto-connect operation. Instead, you can discover the NVMe/TCP subsystems and namespaces by performing the NVMe/TCP connect or connect-all operations manually.

Steps

1. Verify that the initiator port can fetch the discovery log page data across the supported NVMe/TCP LIFs:

   ```
   nvme discover -t tcp -w host-traddr -a traddr
   ```

Show example

```
nvme discover -t tcp -w 192.168.1.31 -a 192.168.1.24
Discovery Log Number of Records 20, Generation counter 25
=====Discovery Log Entry 0======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  4
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr:  192.168.2.25
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 1======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  2
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr:  192.168.1.25
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 2======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  5
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr:  192.168.2.24
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 3======
trtype:  tcp
adrfam:  ipv4
```

```
subtype: current discovery subsystem
treq:    not specified
portid:  1
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr:  192.168.1.24
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 4======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_1
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 5======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_1
traddr:  192.168.1.25
eflags:  none
sectype: none
=====Discovery Log Entry 6======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  5
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_1
```

```
traddr:  192.168.2.24
eflags:  none
sectype: none
=====Discovery Log Entry 7======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_1
traddr:  192.168.1.24
eflags:  none
sectype: none
=====Discovery Log Entry 8======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_4
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 9======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_4
traddr:  192.168.1.25
eflags:  none
sectype: none
=====Discovery Log Entry 10======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
```

```
treq:    not specified
portid:  5
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_4
traddr:  192.168.2.24
eflags:  none
sectype: none
=====Discovery Log Entry 11======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_4
traddr:  192.168.1.24
eflags:  none
sectype: none
=====Discovery Log Entry 12======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_3
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 13======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_3
traddr:  192.168.1.25
```

```
eflags:  none
sectype: none
=====Discovery Log Entry 14======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  5
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_3
traddr:  192.168.2.24
eflags:  none
sectype: none
=====Discovery Log Entry 15======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_3
traddr:  192.168.1.24
eflags:  none
sectype: none
=====Discovery Log Entry 16======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_2
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 17======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
```

```
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_2
traddr:  192.168.1.25
eflags:  none
sectype: none
=====Discovery Log Entry 18======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  5
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_2
traddr:  192.168.2.24
eflags:  none
sectype: none
=====Discovery Log Entry 19======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_2
traddr:  192.168.1.24
eflags:  none
sectype: none
```

2. Verify that the other NVMe/TCP initiator-target LIF combinations are able to successfully fetch discovery log page data:

```
nvme discover -t tcp -w host-traddr -a traddr
```

Show example

```
nvme discover -t tcp -w 192.168.1.31 -a 192.168.1.24
nvme discover -t tcp -w 192.168.2.31 -a 192.168.2.24
nvme discover -t tcp -w 192.168.1.31 -a 192.168.1.25
nvme discover -t tcp -w 192.168.2.31 -a 192.168.2.25
```

3. Run the `nvme connect-all` command across all the supported NVMe/TCP initiator-target LIFs across the nodes:

```
nvme connect-all -t tcp -w host-traddr -a traddr
```

Show example

```
nvme    connect-all -t  tcp -w  192.168.1.31    -a  192.168.1.24
nvme    connect-all -t  tcp -w  192.168.2.31    -a  192.168.2.24
nvme    connect-all -t  tcp -w  192.168.1.31    -a  192.168.1.25
nvme    connect-all -t  tcp -w  192.168.2.31    -a  192.168.2.25
```

Step 6: Validate NVMe-oF

Verify that the in-kernel NVMe multipath status, ANA status, and ONTAP namespaces are correct for the NVMe-oF configuration.

Steps

1. Verify that the in-kernel NVMe multipath is enabled:

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

You should see the following output:

```
Y
```

2. Verify that the appropriate NVMe-oF settings (such as, model set to NetApp ONTAP Controller and load balancing iopolicy set to round-robin) for the respective ONTAP namespaces correctly reflect on the host:
   a. Display the subsystems:

   ```
   cat /sys/class/nvme-subsystem/nvme-subsys*/model
   ```

You should see the following output:

```
NetApp ONTAP Controller
NetApp ONTAP Controller
```

b. Display the policy:

```
cat /sys/class/nvme-subsystem/nvme-subsys*/iopolicy
```

You should see the following output:

```
round-robin
round-robin
```

3. Verify that the namespaces are created and correctly discovered on the host:

```
nvme list
```

Show example

```
Node         SN                   Model
---------------------------------------------------------
/dev/nvme4n1 81Ix2BVuekWcAAAAAAB   NetApp ONTAP Controller


Namespace Usage    Format             FW             Rev
-----------------------------------------------------------
1                 21.47 GB / 21.47 GB   4 KiB + 0 B   FFFFFFFF
```

4. Verify that the controller state of each path is live and has the correct ANA status:

NVMe/FC

```
nvme list-subsys /dev/nvme4n5
```

Show example

```
nvme-subsys4 - NQN=nqn.1992-
08.com.netapp:sn.3a5d31f5502c11ef9f50d039eab6cb6d:subsystem.nvme
_1
               hostnqn=nqn.2014-08.org.nvmexpress:uuid:e6dade64-
216d-
11ec-b7bb-7ed30a5482c3
iopolicy=round-robin\
+- nvme1 fc traddr=nn-0x2082d039eaa7dfc8:pn-
0x2088d039eaa7dfc8,host_traddr=nn-0x20000024ff752e6d:pn-
0x21000024ff752e6d live optimized
+- nvme12 fc traddr=nn-0x2082d039eaa7dfc8:pn-
0x208ad039eaa7dfc8,host_traddr=nn-0x20000024ff752e6d:pn-
0x21000024ff752e6d live non-optimized
+- nvme10 fc traddr=nn-0x2082d039eaa7dfc8:pn-
0x2087d039eaa7dfc8,host_traddr=nn-0x20000024ff752e6c:pn-
0x21000024ff752e6c live non-optimized
+- nvme3 fc traddr=nn-0x2082d039eaa7dfc8:pn-
0x2083d039eaa7dfc8,host_traddr=nn-0x20000024ff752e6c:pn-
0x21000024ff752e6c live optimized
```

NVMe/TCP

```
nvme list-subsys /dev/nvme1n1
```

Show example

```
nvme-subsys5 - NQN=nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme
_tcp_3
hostnqn=nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0035-5910-804b-
b5c04f444d33
iopolicy=round-robin
\
+- nvme13 tcp
traddr=192.168.2.25,trsvcid=4420,host_traddr=192.168.2.31,
src_addr=192.168.2.31 live optimized
+- nvme14 tcp
traddr=192.168.2.24,trsvcid=4420,host_traddr=192.168.2.31,
src_addr=192.168.2.31 live non-optimized
+- nvme5 tcp
traddr=192.168.1.25,trsvcid=4420,host_traddr=192.168.1.31,
src_addr=192.168.1.31 live optimized
+- nvme6 tcp
traddr=192.168.1.24,trsvcid=4420,host_traddr=192.168.1.31,
src_addr=192.168.1.31 live non-optimized
```

5. Verify that the NetApp plug-in displays the correct values for each ONTAP namespace device:

Column

```
nvme netapp ontapdevices -o column
```

Show example

```
Device          Vserver    Namespace Path
----------------------- ------------------------------
/dev/nvme1n1      linux_tcnvme_iscsi
/vol/tcpnvme_1_0_0/tcpnvme_ns

NSID       UUID                                   Size
------------------------------------------------------------
1    5f7f630d-8ea5-407f-a490-484b95b15dd6   21.47GB
```

JSON

```
nvme netapp ontapdevices -o json
```

Show example

```
{
  "ONTAPdevices":[
    {
      "Device":"/dev/nvme1n1",
      "Vserver":"linux_tcnvme_iscsi",
      "Namespace_Path":"/vol/tcpnvme_1_0_0/tcpnvme_ns",
      "NSID":1,
      "UUID":"5f7f630d-8ea5-407f-a490-484b95b15dd6",
      "Size":"21.47GB",
      "LBA_Data_Size":4096,
      "Namespace_Size":5242880
    },
]
}
```

Step 7: Review the known issues

The NVMe-oF host configuration for Rocky Linux 8.9 with ONTAP storage release has the following known issue:

NetApp Bug ID	Title	Description
1479047	Rocky Linux 8.9 NVMe-oF hosts create duplicate persistent discovery controllers	On NVMe over Fabrics (NVMe-oF) hosts, you can use the "nvme discover -p" command to create Persistent Discovery Controllers (PDCs). When this command is used, only one PDC should be created per initiator-target combination. However, if you are running Rocky Linux 8.9 on an NVMe-oF host, a duplicate PDC is created each time "nvme discover -p" is executed. This leads to unnecessary usage of resources on both the host and the target.

Configure Rocky Linux 8.8 with NVMe-oF for ONTAP storage

Rocky Linux 8.8 hosts support the NVMe/FC and NVMe/TCP protocols with Asymmetric Namespace Access (ANA). ANA is equivalent to asymmetric logical unit access (ALUA) multipathing in iSCSI and FCP environments and is implemented using the in-kernel NVMe multipath feature.

For additional details on supported configurations, see the Interoperability Matrix Tool.

About this task

You can use the following support and features with the NVMe-oF host configuration for Rocky Linux 8.8. You should also review the known limitations before starting the configuration process.

- Support available:
  - Support for NVMe over TCP (NVMe/TCP) in addition to NVMe over Fibre Channel (NVMe/FC). The NetApp plug-in in the native `nvme-cli` package displays ONTAP details for both NVMe/FC and NVMe/TCP namespaces.
  - Running NVMe and SCSI traffic on the same host. For example, you can configure dm-multipath for SCSI mpath devices on SCSI LUNs and use NVMe multipath to configure NVMe-oF namespace devices on the host.
- Known limitations:
  - In-kernel NVMe multipath is disabled by default for Rocky Linux 8.8 NVMe-oF hosts. Therefore, you need to enable it manually.
  - On Rocky Linux 8.8 hosts, NVMe/TCP is a technology preview feature due to open issues. Refer to the Rocky Linux 8.8 Release Notes for details.
  - SAN booting using the NVMe-oF protocol is currently not supported.

Step 1: Optionally, enable SAN booting

You can configure your host to use SAN booting to simplify deployment and improve scalability.

Before you begin

Use the Interoperability Matrix Tool to verify that your Linux OS, host bus adapter (HBA), HBA firmware, HBA boot BIOS, and ONTAP version support SAN booting.

Steps

1. Create a SAN boot namespace and map it to the host.
2. Enable SAN booting in the server BIOS for the ports to which the SAN boot namespace is mapped.

   For information on how to enable the HBA BIOS, see your vendor-specific documentation.

3. Verify that the configuration was successful by rebooting the host and verifying that the OS is up and running.

Step 2: Validate software versions

Use the following procedure to validate the minimum supported Rocky Linux 8.8 software versions.

Steps

1. Install Rocky Linux 8.8 on the server. After the installation is complete, verify that you are running the specified Rocky Linux 8.8 kernel:

   ```
   uname -r
   ```

   The following example shows a Rocky Linux kernel version:

   ```
   5.14.0-570.12.1.el9_6.x86_64
   ```

2. Install the `nvme-cli` package:

   ```
   rpm -qa|grep nvme-cli
   ```

   The following example shows an nvme-cli package version:

   ```
   nvme-cli-2.11-5.el9.x86_64
   ```

3. Install the `libnvme` package:

   ```
   rpm -qa|grep libnvme
   ```

   The following example shows an `libnvme` package version:

   ```
   libnvme-1.11.1-1.el9.x86_64
   ```

4. On the Rocky Linux host, check the hostnqn string at `/etc/nvme/hostnqn`:

   ```
   cat /etc/nvme/hostnqn
   ```

The following example shows an `hostnqn` version:

```
nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
```

5. Verify that the `hostnqn` string matches the `hostnqn` string for the corresponding subsystem on the ONTAP array:

```
::> vserver nvme subsystem host show -vserver vs_coexistence_LPE36002
```

Show example

```
Vserver Subsystem Priority  Host NQN
------- --------- --------
------------------------------------------------
vs_coexistence_LPE36002
        nvme
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
        nvme_1
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
        nvme_2
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
        nvme_3
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
4 entries were displayed.
```

> If the `hostnqn` strings do not match, use the `vserver modify` command to update the `hostnqn` string on your corresponding ONTAP array subsystem to match the `hostnqn` string from `/etc/nvme/hostnqn` on the host.

Step 3: Configure NVMe/FC

You can configure NVMe/FC with Broadcom/Emulex FC or Marvell/Qlogic FC adapters. You also need to manually discover the NVMe/TCP subsystems and namespaces.

Broadcom/Emulex

Configure NVMe/FC for a Broadcom/Emulex adapter.

Steps

1. Verify that you are using the supported adapter model:
   a. Display the model names:

   ```
   cat /sys/class/scsi_host/host*/modelname
   ```

   You should see the following output:

   ```
   LPe36002-M64
   LPe36002-M64
   ```

   b. Display the model descriptions:

   ```
   cat /sys/class/scsi_host/host*/modeldesc
   ```

   You should see an output similar to the following example:

   ```
   Emulex LightPulse LPe36002-M64 2-Port 64Gb Fibre Channel Adapter
   Emulex LightPulse LPe36002-M64 2-Port 64Gb Fibre Channel Adapter
   ```

2. Verify that you are using the recommended Broadcom `lpfc` firmware and inbox driver:
   a. Display the firmware version:

   ```
   cat /sys/class/scsi_host/host*/fwrev
   ```

   The following example shows firmware versions:

   ```
   14.4.317.10, sli-4:6:d
   14.4.317.10, sli-4:6:d
   ```

   b. Display the inbox driver version:

   ```
   cat /sys/module/lpfc/version`
   ```

   The following example shows a driver version:

```
0:14.4.0.2
```

For the current list of supported adapter driver and firmware versions, see the Interoperability Matrix Tool.

3. Verify that the expected output of `lpfc_enable_fc4_type` is set to `3`:

   ```
   cat /sys/module/lpfc/parameters/lpfc_enable_fc4_type
   ```

4. Verify that you can view your initiator ports:

   ```
   cat /sys/class/fc_host/host*/port_name
   ```

   The following example shows port identities:

   ```
   0x100000109bf044b1
   0x100000109bf044b2
   ```

5. Verify that your initiator ports are online:

   ```
   cat /sys/class/fc_host/host*/port_state
   ```

   You should see the following output:

   ```
   Online
   Online
   ```

6. Verify that the NVMe/FC initiator ports are enabled and that the target ports are visible:

   ```
   cat /sys/class/scsi_host/host*/nvme_info
   ```

Show example

```
NVME Initiator Enabled
XRI Dist lpfc2 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc2 WWPN x100000109bf044b1 WWNN x200000109bf044b1
DID x022a00 ONLINE
NVME RPORT       WWPN x202fd039eaa7dfc8 WWNN x202cd039eaa7dfc8
DID x021310 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x202dd039eaa7dfc8 WWNN x202cd039eaa7dfc8
DID x020b10 TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 0000000810 Cmpl 0000000810 Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000007b098f07 Issue 000000007aee27c4 OutIO
fffffffffe498bd
        abort 000013b4 noxri 00000000 nondlp 00000058 qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 000013b4 Err 00021443

NVME Initiator Enabled
XRI Dist lpfc3 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc3 WWPN x100000109bf044b2 WWNN x200000109bf044b2
DID x021b00 ONLINE
NVME RPORT       WWPN x2033d039eaa7dfc8 WWNN x202cd039eaa7dfc8
DID x020110 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2032d039eaa7dfc8 WWNN x202cd039eaa7dfc8
DID x022910 TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 0000000840 Cmpl 0000000840 Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000007afd4434 Issue 000000007ae31b83 OutIO
fffffffffe5d74f
        abort 000014a5 noxri 00000000 nondlp 0000006a qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 000014a5 Err 0002149a
```

Marvell/QLogic

Configure NVMe/FC for a Marvell/QLogic adapter.

> ⓘ The native inbox qla2xxx driver included in the Rocky Linux kernel has the latest fixes. These fixes are essential for ONTAP support.

Steps

1. Verify that you are running the supported adapter driver and firmware versions:

```
cat /sys/class/fc_host/host*/symbolic_name
```

The follow example shows driver and firmware versions:

```
QLE2742 FW:v9.14.00 DVR:v10.02.09.200-k
QLE2742 FW:v9.14.00 DVR:v10.02.09.200-k
```

2. Verify that `ql2xnvmeenable` is set. This enables the Marvell adapter to function as an NVMe/FC initiator:

```
cat /sys/module/qla2xxx/parameters/ql2xnvmeenable
```

The expected output is 1.

Step 4: Optionally, enable 1MB I/O

You can enable I/O requests of size 1MB for NVMe/FC configured with a Broadcom adapter. ONTAP reports a Max Data Transfer Size (MDTS) of 8 in the Identify Controller data. This means the maximum I/O request size can be up to 1MB. To issue I/O requests of size 1MB, you need to increase the lpfc value of the `lpfc_sg_seg_cnt` parameter to 256 from the default value of 64.

> These steps don't apply to Qlogic NVMe/FC hosts.

Steps

1. Set the `lpfc_sg_seg_cnt` parameter to 256:

```
cat /etc/modprobe.d/lpfc.conf
```

```
options lpfc lpfc_sg_seg_cnt=256
```

2. Run the `dracut -f` command, and reboot the host.
3. Verify that the value for `lpfc_sg_seg_cnt` is 256:

```
cat /sys/module/lpfc/parameters/lpfc_sg_seg_cnt
```

Step 5: Configure NVMe/TCP

The NVMe/TCP protocol doesn't support the auto-connect operation. Instead, you can discover the NVMe/TCP subsystems and namespaces by performing the NVMe/TCP connect or connect-all operations manually.

Steps

1. Verify that the initiator port can fetch the discovery log page data across the supported NVMe/TCP LIFs:

   ```
   nvme discover -t tcp -w host-traddr -a traddr
   ```

Show example

```
nvme discover -t tcp -w 192.168.1.31 -a 192.168.1.24
Discovery Log Number of Records 20, Generation counter 25
=====Discovery Log Entry 0======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  4
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr:  192.168.2.25
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 1======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  2
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr:  192.168.1.25
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 2======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  5
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr:  192.168.2.24
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 3======
trtype:  tcp
adrfam:  ipv4
```

```
subtype: current discovery subsystem
treq:    not specified
portid:  1
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr:  192.168.1.24
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 4======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_1
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 5======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_1
traddr:  192.168.1.25
eflags:  none
sectype: none
=====Discovery Log Entry 6======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  5
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_1
```

```
traddr:  192.168.2.24
eflags:  none
sectype: none
=====Discovery Log Entry 7======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_1
traddr:  192.168.1.24
eflags:  none
sectype: none
=====Discovery Log Entry 8======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_4
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 9======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_4
traddr:  192.168.1.25
eflags:  none
sectype: none
=====Discovery Log Entry 10======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
```

```
treq:    not specified
portid:  5
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_4
traddr:  192.168.2.24
eflags:  none
sectype: none
=====Discovery Log Entry 11======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_4
traddr:  192.168.1.24
eflags:  none
sectype: none
=====Discovery Log Entry 12======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_3
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 13======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_3
traddr:  192.168.1.25
```

```
eflags:  none
sectype: none
=====Discovery Log Entry 14======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  5
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_3
traddr:  192.168.2.24
eflags:  none
sectype: none
=====Discovery Log Entry 15======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_3
traddr:  192.168.1.24
eflags:  none
sectype: none
=====Discovery Log Entry 16======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_2
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 17======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
```

```
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_2
traddr:  192.168.1.25
eflags:  none
sectype: none
=====Discovery Log Entry 18======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  5
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_2
traddr:  192.168.2.24
eflags:  none
sectype: none
=====Discovery Log Entry 19======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_2
traddr:  192.168.1.24
eflags:  none
sectype: none
```

2. Verify that the other NVMe/TCP initiator-target LIF combinations are able to successfully fetch discovery log page data:

```
nvme discover -t tcp -w host-traddr -a traddr
```

Show example

```
nvme discover -t tcp -w 192.168.1.31 -a 192.168.1.24
nvme discover -t tcp -w 192.168.2.31 -a 192.168.2.24
nvme discover -t tcp -w 192.168.1.31 -a 192.168.1.25
nvme discover -t tcp -w 192.168.2.31 -a 192.168.2.25
```

3. Run the `nvme connect-all` command across all the supported NVMe/TCP initiator-target LIFs across the nodes:

```
nvme connect-all -t tcp -w host-traddr -a traddr
```

Show example

```
nvme    connect-all -t  tcp -w  192.168.1.31    -a  192.168.1.24
nvme    connect-all -t  tcp -w  192.168.2.31    -a  192.168.2.24
nvme    connect-all -t  tcp -w  192.168.1.31    -a  192.168.1.25
nvme    connect-all -t  tcp -w  192.168.2.31    -a  192.168.2.25
```

Step 6: Validate NVMe-oF

Verify that the in-kernel NVMe multipath status, ANA status, and ONTAP namespaces are correct for the NVMe-oF configuration.

Steps

1. Verify that the in-kernel NVMe multipath is enabled:

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

You should see the following output:

```
Y
```

2. Verify that the appropriate NVMe-oF settings (such as, model set to NetApp ONTAP Controller and load balancing iopolicy set to round-robin) for the respective ONTAP namespaces correctly reflect on the host:

   a. Display the subsystems:

   ```
   cat /sys/class/nvme-subsystem/nvme-subsys*/model
   ```

You should see the following output:

```
NetApp ONTAP Controller
NetApp ONTAP Controller
```

b. Display the policy:

```
cat /sys/class/nvme-subsystem/nvme-subsys*/iopolicy
```

You should see the following output:

```
round-robin
round-robin
```

3. Verify that the namespaces are created and correctly discovered on the host:

```
nvme list
```

Show example

```
Node         SN                   Model
---------------------------------------------------------
/dev/nvme4n1 81Ix2BVuekWcAAAAAAB   NetApp ONTAP Controller


Namespace Usage    Format             FW            Rev
-----------------------------------------------------------
1                 21.47 GB / 21.47 GB   4 KiB + 0 B   FFFFFFFF
```

4. Verify that the controller state of each path is live and has the correct ANA status:

NVMe/FC

```
nvme list-subsys /dev/nvme4n5
```

Show example

```
nvme-subsys4 - NQN=nqn.1992-
08.com.netapp:sn.3a5d31f5502c11ef9f50d039eab6cb6d:subsystem.nvme
_1
               hostnqn=nqn.2014-08.org.nvmexpress:uuid:e6dade64-
216d-
11ec-b7bb-7ed30a5482c3
iopolicy=round-robin\
+- nvme1 fc traddr=nn-0x2082d039eaa7dfc8:pn-
0x2088d039eaa7dfc8,host_traddr=nn-0x20000024ff752e6d:pn-
0x21000024ff752e6d live optimized
+- nvme12 fc traddr=nn-0x2082d039eaa7dfc8:pn-
0x208ad039eaa7dfc8,host_traddr=nn-0x20000024ff752e6d:pn-
0x21000024ff752e6d live non-optimized
+- nvme10 fc traddr=nn-0x2082d039eaa7dfc8:pn-
0x2087d039eaa7dfc8,host_traddr=nn-0x20000024ff752e6c:pn-
0x21000024ff752e6c live non-optimized
+- nvme3 fc traddr=nn-0x2082d039eaa7dfc8:pn-
0x2083d039eaa7dfc8,host_traddr=nn-0x20000024ff752e6c:pn-
0x21000024ff752e6c live optimized
```

NVMe/TCP

```
nvme list-subsys /dev/nvme1n1
```

Show example

```
nvme-subsys5 - NQN=nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme
_tcp_3
hostnqn=nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0035-5910-804b-
b5c04f444d33
iopolicy=round-robin
\
+- nvme13 tcp
traddr=192.168.2.25,trsvcid=4420,host_traddr=192.168.2.31,
src_addr=192.168.2.31 live optimized
+- nvme14 tcp
traddr=192.168.2.24,trsvcid=4420,host_traddr=192.168.2.31,
src_addr=192.168.2.31 live non-optimized
+- nvme5 tcp
traddr=192.168.1.25,trsvcid=4420,host_traddr=192.168.1.31,
src_addr=192.168.1.31 live optimized
+- nvme6 tcp
traddr=192.168.1.24,trsvcid=4420,host_traddr=192.168.1.31,
src_addr=192.168.1.31 live non-optimized
```

5. Verify that the NetApp plug-in displays the correct values for each ONTAP namespace device:

Column

```
nvme netapp ontapdevices -o column
```

Show example

```
Device        Vserver   Namespace Path
----------------------- ------------------------------
/dev/nvme1n1     linux_tcnvme_iscsi
/vol/tcpnvme_1_0_0/tcpnvme_ns

NSID       UUID                                   Size
------------------------------------------------------------
1    5f7f630d-8ea5-407f-a490-484b95b15dd6   21.47GB
```

JSON

```
nvme netapp ontapdevices -o json
```

Show example


```
{
  "ONTAPdevices":[
    {
      "Device":"/dev/nvme1n1",
      "Vserver":"linux_tcnvme_iscsi",
      "Namespace_Path":"/vol/tcpnvme_1_0_0/tcpnvme_ns",
      "NSID":1,
      "UUID":"5f7f630d-8ea5-407f-a490-484b95b15dd6",
      "Size":"21.47GB",
      "LBA_Data_Size":4096,
      "Namespace_Size":5242880
    },
]
}
```


Step 7: Review the known issues

The NVMe-oF host configuration for Rocky Linux 8.8 with ONTAP storage release has the following known issues:

NetApp Bug ID	Title	Description
1479047	Rocky Linux 8.8 NVMe-oF hosts create duplicate persistent discovery controllers	On NVMe over Fabrics (NVMe-oF) hosts, you can use the "nvme discover -p" command to create Persistent Discovery Controllers (PDCs). When this command is used, only one PDC should be created per initiator-target combination. However, if you are running Rocky Linux 8.8 on an NVMe-oF host, a duplicate PDC is created each time "nvme discover -p" is executed. This leads to unnecessary usage of resources on both the host and the target.

Configure Rocky Linux 8.7 with NVMe-oF for ONTAP storage

Rocky Linux 8.7 hosts support the NVMe/FC and NVMe/TCP protocols with Asymmetric Namespace Access (ANA). ANA is equivalent to asymmetric logical unit access (ALUA) multipathing in iSCSI and FCP environments and is implemented using the in-kernel NVMe multipath feature.

For additional details on supported configurations, see the Interoperability Matrix Tool.

About this task

You can use the following support and features with the NVMe-oF host configuration for Rocky Linux 8.7. You should also review the known limitations before starting the configuration process.

- Support available:
  - Support for NVMe over TCP (NVMe/TCP) in addition to NVMe over Fibre Channel (NVMe/FC). The NetApp plug-in in the native `nvme-cli` package displays ONTAP details for both NVMe/FC and NVMe/TCP namespaces.
  - Running NVMe and SCSI traffic on the same host. For example, you can configure dm-multipath for SCSI mpath devices on SCSI LUNs and use NVMe multipath to configure NVMe-oF namespace devices on the host.
- Known limitations:
  - In-kernel NVMe multipath is disabled by default for Rocky Linux 8.7 NVMe-oF hosts. Therefore, you need to enable it manually.
  - On Rocky Linux 8.7 hosts, NVMe/TCP is a technology preview feature due to open issues. Refer to the Rocky Linux 8.7 release notes for details.
  - SAN booting using the NVMe-oF protocol is currently not supported.

Step 1: Optionally, enable SAN booting

You can configure your host to use SAN booting to simplify deployment and improve scalability.

Before you begin

Use the Interoperability Matrix Tool to verify that your Linux OS, host bus adapter (HBA), HBA firmware, HBA boot BIOS, and ONTAP version support SAN booting.

Steps

1. Create a SAN boot namespace and map it to the host.
2. Enable SAN booting in the server BIOS for the ports to which the SAN boot namespace is mapped.

   For information on how to enable the HBA BIOS, see your vendor-specific documentation.

3. Verify that the configuration was successful by rebooting the host and verifying that the OS is up and running.

Step 2: Validate software versions

Use the following procedure to validate the minimum supported Rocky Linux 8.7 software versions.

Steps

1. Install Rocky Linux 8.7 on the server. After the installation is complete, verify that you are running the specified Rocky Linux 8.7 kernel:

   ```
   uname -r
   ```

   The following example shows a Rocky Linux kernel version:

   ```
   5.14.0-570.12.1.el9_6.x86_64
   ```

2. Install the `nvme-cli` package:

   ```
   rpm -qa|grep nvme-cli
   ```

   The following example shows an nvme-cli package version:

   ```
   nvme-cli-2.11-5.el9.x86_64
   ```

3. Install the `libnvme` package:

   ```
   rpm -qa|grep libnvme
   ```

   The following example shows an `libnvme` package version:

   ```
   libnvme-1.11.1-1.el9.x86_64
   ```

4. On the Rocky Linux host, check the hostnqn string at `/etc/nvme/hostnqn`:

   ```
   cat /etc/nvme/hostnqn
   ```

The following example shows an `hostnqn` version:

```
nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
```

5. Verify that the `hostnqn` string matches the `hostnqn` string for the corresponding subsystem on the ONTAP array:

```
::> vserver nvme subsystem host show -vserver vs_coexistence_LPE36002
```

Show example

```
Vserver Subsystem Priority  Host NQN
------- --------- --------
------------------------------------------------
vs_coexistence_LPE36002
        nvme
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
        nvme_1
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
        nvme_2
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
        nvme_3
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
4 entries were displayed.
```

> If the `hostnqn` strings do not match, use the `vserver modify` command to update the `hostnqn` string on your corresponding ONTAP array subsystem to match the `hostnqn` string from `/etc/nvme/hostnqn` on the host.

Step 3: Configure NVMe/FC

You can configure NVMe/FC with Broadcom/Emulex FC or Marvell/Qlogic FC adapters. You also need to manually discover the NVMe/TCP subsystems and namespaces.

Broadcom/Emulex

Configure NVMe/FC for a Broadcom/Emulex adapter.

Steps

1. Verify that you are using the supported adapter model:
   a. Display the model names:

   ```
   cat /sys/class/scsi_host/host*/modelname
   ```

   You should see the following output:

   ```
   LPe36002-M64
   LPe36002-M64
   ```

   b. Display the model descriptions:

   ```
   cat /sys/class/scsi_host/host*/modeldesc
   ```

   You should see an output similar to the following example:

   ```
   Emulex LightPulse LPe36002-M64 2-Port 64Gb Fibre Channel Adapter
   Emulex LightPulse LPe36002-M64 2-Port 64Gb Fibre Channel Adapter
   ```

2. Verify that you are using the recommended Broadcom `lpfc` firmware and inbox driver:
   a. Display the firmware version:

   ```
   cat /sys/class/scsi_host/host*/fwrev
   ```

   The following example shows firmware versions:

   ```
   14.4.317.10, sli-4:6:d
   14.4.317.10, sli-4:6:d
   ```

   b. Display the inbox driver version:

   ```
   cat /sys/module/lpfc/version`
   ```

   The following example shows a driver version:

```
0:14.4.0.2
```

For the current list of supported adapter driver and firmware versions, see the Interoperability Matrix Tool.

3. Verify that the expected output of `lpfc_enable_fc4_type` is set to `3`:

   ```
   cat /sys/module/lpfc/parameters/lpfc_enable_fc4_type
   ```

4. Verify that you can view your initiator ports:

   ```
   cat /sys/class/fc_host/host*/port_name
   ```

   The following example shows port identities:

   ```
   0x100000109bf044b1
   0x100000109bf044b2
   ```

5. Verify that your initiator ports are online:

   ```
   cat /sys/class/fc_host/host*/port_state
   ```

   You should see the following output:

   ```
   Online
   Online
   ```

6. Verify that the NVMe/FC initiator ports are enabled and that the target ports are visible:

   ```
   cat /sys/class/scsi_host/host*/nvme_info
   ```

Show example

```
NVME Initiator Enabled
XRI Dist lpfc2 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc2 WWPN x100000109bf044b1 WWNN x200000109bf044b1
DID x022a00 ONLINE
NVME RPORT       WWPN x202fd039eaa7dfc8 WWNN x202cd039eaa7dfc8
DID x021310 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x202dd039eaa7dfc8 WWNN x202cd039eaa7dfc8
DID x020b10 TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 0000000810 Cmpl 0000000810 Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000007b098f07 Issue 000000007aee27c4 OutIO
fffffffffe498bd
        abort 000013b4 noxri 00000000 nondlp 00000058 qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 000013b4 Err 00021443

NVME Initiator Enabled
XRI Dist lpfc3 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc3 WWPN x100000109bf044b2 WWNN x200000109bf044b2
DID x021b00 ONLINE
NVME RPORT       WWPN x2033d039eaa7dfc8 WWNN x202cd039eaa7dfc8
DID x020110 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2032d039eaa7dfc8 WWNN x202cd039eaa7dfc8
DID x022910 TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 0000000840 Cmpl 0000000840 Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000007afd4434 Issue 000000007ae31b83 OutIO
fffffffffe5d74f
        abort 000014a5 noxri 00000000 nondlp 0000006a qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 000014a5 Err 0002149a
```

Marvell/QLogic

Configure NVMe/FC for a Marvell/QLogic adapter.

ⓘ The native inbox qla2xxx driver included in the Rocky Linux kernel has the latest fixes. These fixes are essential for ONTAP support.

Steps

1. Verify that you are running the supported adapter driver and firmware versions:

   ```
   cat /sys/class/fc_host/host*/symbolic_name
   ```

   The follow example shows driver and firmware versions:

   ```
   QLE2742 FW:v9.14.00 DVR:v10.02.09.200-k
   QLE2742 FW:v9.14.00 DVR:v10.02.09.200-k
   ```

2. Verify that `ql2xnvmeenable` is set. This enables the Marvell adapter to function as an NVMe/FC initiator:

   ```
   cat /sys/module/qla2xxx/parameters/ql2xnvmeenable
   ```

   The expected output is 1.

Step 4: Optionally, enable 1MB I/O

You can enable I/O requests of size 1MB for NVMe/FC configured with a Broadcom adapter. ONTAP reports a Max Data Transfer Size (MDTS) of 8 in the Identify Controller data. This means the maximum I/O request size can be up to 1MB. To issue I/O requests of size 1MB, you need to increase the lpfc value of the `lpfc_sg_seg_cnt` parameter to 256 from the default value of 64.

> These steps don't apply to Qlogic NVMe/FC hosts.

Steps

1. Set the `lpfc_sg_seg_cnt` parameter to 256:

   ```
   cat /etc/modprobe.d/lpfc.conf
   ```

   ```
   options lpfc lpfc_sg_seg_cnt=256
   ```

2. Run the `dracut -f` command, and reboot the host.
3. Verify that the value for `lpfc_sg_seg_cnt` is 256:

   ```
   cat /sys/module/lpfc/parameters/lpfc_sg_seg_cnt
   ```

Step 5: Configure NVMe/TCP

The NVMe/TCP protocol doesn’t support the auto-connect operation. Instead, you can discover the NVMe/TCP subsystems and namespaces by performing the NVMe/TCP connect or connect-all operations manually.

Steps

1. Verify that the initiator port can fetch the discovery log page data across the supported NVMe/TCP LIFs:

   ```
   nvme discover -t tcp -w host-traddr -a traddr
   ```

Show example

```
nvme discover -t tcp -w 192.168.1.31 -a 192.168.1.24
Discovery Log Number of Records 20, Generation counter 25
=====Discovery Log Entry 0======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  4
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr:  192.168.2.25
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 1======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  2
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr:  192.168.1.25
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 2======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  5
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr:  192.168.2.24
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 3======
trtype:  tcp
adrfam:  ipv4
```

```
subtype: current discovery subsystem
treq:    not specified
portid:  1
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr:  192.168.1.24
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 4======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_1
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 5======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_1
traddr:  192.168.1.25
eflags:  none
sectype: none
=====Discovery Log Entry 6======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  5
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_1
```

```
traddr:  192.168.2.24
eflags:  none
sectype: none
=====Discovery Log Entry 7======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_1
traddr:  192.168.1.24
eflags:  none
sectype: none
=====Discovery Log Entry 8======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_4
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 9======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_4
traddr:  192.168.1.25
eflags:  none
sectype: none
=====Discovery Log Entry 10======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
```

```
treq:    not specified
portid:  5
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_4
traddr:  192.168.2.24
eflags:  none
sectype: none
=====Discovery Log Entry 11======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_4
traddr:  192.168.1.24
eflags:  none
sectype: none
=====Discovery Log Entry 12======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_3
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 13======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_3
traddr:  192.168.1.25
```

```
eflags:  none
sectype: none
=====Discovery Log Entry 14======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  5
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_3
traddr:  192.168.2.24
eflags:  none
sectype: none
=====Discovery Log Entry 15======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_3
traddr:  192.168.1.24
eflags:  none
sectype: none
=====Discovery Log Entry 16======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_2
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 17======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
```

```
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_2
traddr:  192.168.1.25
eflags:  none
sectype: none
=====Discovery Log Entry 18======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  5
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_2
traddr:  192.168.2.24
eflags:  none
sectype: none
=====Discovery Log Entry 19======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_2
traddr:  192.168.1.24
eflags:  none
sectype: none
```

2. Verify that the other NVMe/TCP initiator-target LIF combinations are able to successfully fetch discovery log page data:

```
nvme discover -t tcp -w host-traddr -a traddr
```

Show example

```
nvme discover -t tcp -w 192.168.1.31 -a 192.168.1.24
nvme discover -t tcp -w 192.168.2.31 -a 192.168.2.24
nvme discover -t tcp -w 192.168.1.31 -a 192.168.1.25
nvme discover -t tcp -w 192.168.2.31 -a 192.168.2.25
```

3. Run the `nvme connect-all` command across all the supported NVMe/TCP initiator-target LIFs across the nodes:

```
nvme connect-all -t tcp -w host-traddr -a traddr
```

Show example

```
nvme    connect-all -t  tcp -w  192.168.1.31    -a  192.168.1.24
nvme    connect-all -t  tcp -w  192.168.2.31    -a  192.168.2.24
nvme    connect-all -t  tcp -w  192.168.1.31    -a  192.168.1.25
nvme    connect-all -t  tcp -w  192.168.2.31    -a  192.168.2.25
```

Step 6: Validate NVMe-oF

Verify that the in-kernel NVMe multipath status, ANA status, and ONTAP namespaces are correct for the NVMe-oF configuration.

Steps

1. Verify that the in-kernel NVMe multipath is enabled:

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

   You should see the following output:

```
Y
```

2. Verify that the appropriate NVMe-oF settings (such as, model set to NetApp ONTAP Controller and load balancing iopolicy set to round-robin) for the respective ONTAP namespaces correctly reflect on the host:

   a. Display the subsystems:

```
cat /sys/class/nvme-subsystem/nvme-subsys*/model
```

You should see the following output:

```
NetApp ONTAP Controller
NetApp ONTAP Controller
```

b. Display the policy:

```
cat /sys/class/nvme-subsystem/nvme-subsys*/iopolicy
```

You should see the following output:

```
round-robin
round-robin
```

3. Verify that the namespaces are created and correctly discovered on the host:

```
nvme list
```

Show example

```
Node         SN                   Model
---------------------------------------------------------
/dev/nvme4n1 81Ix2BVuekWcAAAAAAAB   NetApp ONTAP Controller


Namespace Usage    Format             FW            Rev
-----------------------------------------------------------
1                 21.47 GB / 21.47 GB   4 KiB + 0 B   FFFFFFFF
```

4. Verify that the controller state of each path is live and has the correct ANA status:

NVMe/FC

```
nvme list-subsys /dev/nvme4n5
```

Show example

```
nvme-subsys4 - NQN=nqn.1992-
08.com.netapp:sn.3a5d31f5502c11ef9f50d039eab6cb6d:subsystem.nvme
_1
               hostnqn=nqn.2014-08.org.nvmexpress:uuid:e6dade64-
216d-
11ec-b7bb-7ed30a5482c3
iopolicy=round-robin\
+- nvme1 fc traddr=nn-0x2082d039eaa7dfc8:pn-
0x2088d039eaa7dfc8,host_traddr=nn-0x20000024ff752e6d:pn-
0x21000024ff752e6d live optimized
+- nvme12 fc traddr=nn-0x2082d039eaa7dfc8:pn-
0x208ad039eaa7dfc8,host_traddr=nn-0x20000024ff752e6d:pn-
0x21000024ff752e6d live non-optimized
+- nvme10 fc traddr=nn-0x2082d039eaa7dfc8:pn-
0x2087d039eaa7dfc8,host_traddr=nn-0x20000024ff752e6c:pn-
0x21000024ff752e6c live non-optimized
+- nvme3 fc traddr=nn-0x2082d039eaa7dfc8:pn-
0x2083d039eaa7dfc8,host_traddr=nn-0x20000024ff752e6c:pn-
0x21000024ff752e6c live optimized
```

NVMe/TCP

```
nvme list-subsys /dev/nvme1n1
```

Show example

```
nvme-subsys5 - NQN=nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme
_tcp_3
hostnqn=nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0035-5910-804b-
b5c04f444d33
iopolicy=round-robin
\
+- nvme13 tcp
traddr=192.168.2.25,trsvcid=4420,host_traddr=192.168.2.31,
src_addr=192.168.2.31 live optimized
+- nvme14 tcp
traddr=192.168.2.24,trsvcid=4420,host_traddr=192.168.2.31,
src_addr=192.168.2.31 live non-optimized
+- nvme5 tcp
traddr=192.168.1.25,trsvcid=4420,host_traddr=192.168.1.31,
src_addr=192.168.1.31 live optimized
+- nvme6 tcp
traddr=192.168.1.24,trsvcid=4420,host_traddr=192.168.1.31,
src_addr=192.168.1.31 live non-optimized
```

5. Verify that the NetApp plug-in displays the correct values for each ONTAP namespace device:

Column

```
nvme netapp ontapdevices -o column
```

Show example

```
Device        Vserver   Namespace Path
----------------------- ------------------------------
/dev/nvme1n1     linux_tcnvme_iscsi
/vol/tcpnvme_1_0_0/tcpnvme_ns

NSID       UUID                                   Size
------------------------------------------------------------
1    5f7f630d-8ea5-407f-a490-484b95b15dd6   21.47GB
```

JSON

```
nvme netapp ontapdevices -o json
```

Show example

```
{
  "ONTAPdevices":[
    {
      "Device":"/dev/nvme1n1",
      "Vserver":"linux_tcnvme_iscsi",
      "Namespace_Path":"/vol/tcpnvme_1_0_0/tcpnvme_ns",
      "NSID":1,
      "UUID":"5f7f630d-8ea5-407f-a490-484b95b15dd6",
      "Size":"21.47GB",
      "LBA_Data_Size":4096,
      "Namespace_Size":5242880
    },
]
}
```

Step 7: Review the known issues

The NVMe-oF host configuration for Rocky Linux 8.7 with ONTAP storage has the following known issues:

NetApp Bug ID	Title	Description
1479047	Rocky Linux 8.7 NVMe-oF hosts create duplicate Persistent Discovery Controllers	On NVMe over Fabrics (NVMe-oF) hosts, you can use the "nvme discover -p" command to create Persistent Discovery Controllers (PDCs). When this command is used, only one PDC should be created per initiator-target combination. However, if you are running ONTAP 9.10.1 and Rocky Linux 8.7 with an NVMe-oF host, a duplicate PDC is created each time "nvme discover -p" is executed. This leads to unnecessary usage of resources on both the host and the target.

Configure Rocky Linux 8.6 with NVMe-oF for ONTAP storage

Rocky Linux 8.6 hosts support the NVMe/FC and NVMe/TCP protocols with Asymmetric Namespace Access (ANA). ANA is equivalent to asymmetric logical unit access (ALUA) multipathing in iSCSI and FCP environments and is implemented using the in-kernel NVMe multipath feature.

For additional details on supported configurations, see the Interoperability Matrix Tool.

About this task

You can use the following support and features with the NVMe-oF host configuration for Rocky Linux 8.6. You should also review the known limitations before starting the configuration process.

- Support available:
  - Support for NVMe over TCP (NVMe/TCP) in addition to NVMe over Fibre Channel (NVMe/FC). The NetApp plug-in in the native `nvme-cli` package displays ONTAP details for both NVMe/FC and NVMe/TCP namespaces.
  - Running NVMe and SCSI traffic on the same host. For example, you can configure dm-multipath for SCSI mpath devices on SCSI LUNs and use NVMe multipath to configure NVMe-oF namespace devices on the host.
- Known limitations:
  - In-kernel NVMe multipath is disabled by default for Rocky Linux 8.6 NVMe-oF hosts. Therefore, you need to enable it manually.
  - On Rocky Linux 8.6 hosts, NVMe/TCP is a technology preview feature due to open issues. Refer to the Rocky Linux 8.6 Release Notes for details.
  - SAN booting using the NVMe-oF protocol is currently not supported.

Step 1: Optionally, enable SAN booting

You can configure your host to use SAN booting to simplify deployment and improve scalability.

Before you begin

Use the Interoperability Matrix Tool to verify that your Linux OS, host bus adapter (HBA), HBA firmware, HBA boot BIOS, and ONTAP version support SAN booting.

Steps

1. Create a SAN boot namespace and map it to the host.
2. Enable SAN booting in the server BIOS for the ports to which the SAN boot namespace is mapped.

   For information on how to enable the HBA BIOS, see your vendor-specific documentation.
3. Verify that the configuration was successful by rebooting the host and verifying that the OS is up and running.

Step 2: Validate software versions

Use the following procedure to validate the minimum supported Rocky Linux 8.6 software versions.

Steps

1. Install Rocky Linux 8.6 on the server. After the installation is complete, verify that you are running the specified Rocky Linux 8.6 kernel:

   ```
   uname -r
   ```

   The following example shows a Rocky Linux kernel version:

   ```
   5.14.0-570.12.1.el9_6.x86_64
   ```
2. Install the `nvme-cli` package:

   ```
   rpm -qa|grep nvme-cli
   ```

   The following example shows an nvme-cli package version:

   ```
   nvme-cli-2.11-5.el9.x86_64
   ```
3. Install the `libnvme` package:

   ```
   rpm -qa|grep libnvme
   ```

   The following example shows an `libnvme` package version:

   ```
   libnvme-1.11.1-1.el9.x86_64
   ```
4. On the Rocky Linux host, check the hostnqn string at `/etc/nvme/hostnqn`:

```
cat /etc/nvme/hostnqn
```

The following example shows an `hostnqn` version:

```
nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
```

5. Verify that the `hostnqn` string matches the `hostnqn` string for the corresponding subsystem on the ONTAP array:

```
::> vserver nvme subsystem host show -vserver vs_coexistence_LPE36002
```

Show example

```
Vserver Subsystem Priority  Host NQN
------- --------- --------
------------------------------------------------
vs_coexistence_LPE36002
        nvme
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
        nvme_1
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
        nvme_2
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
        nvme_3
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
4 entries were displayed.
```

If the `hostnqn` strings do not match, use the `vserver modify` command to update the `hostnqn` string on your corresponding ONTAP array subsystem to match the `hostnqn` string from `/etc/nvme/hostnqn` on the host.

Step 3: Configure NVMe/FC

You can configure NVMe/FC with Broadcom/Emulex FC or Marvell/Qlogic FC adapters. You also need to manually discover the NVMe/TCP subsystems and namespaces.

Broadcom/Emulex

Configure NVMe/FC for a Broadcom/Emulex adapter.

Steps

1. Verify that you are using the supported adapter model:

   a. Display the model names:

   ```
   cat /sys/class/scsi_host/host*/modelname
   ```

   You should see the following output:

   ```
   LPe36002-M64
   LPe36002-M64
   ```

   b. Display the model descriptions:

   ```
   cat /sys/class/scsi_host/host*/modeldesc
   ```

   You should see an output similar to the following example:

   ```
   Emulex LightPulse LPe36002-M64 2-Port 64Gb Fibre Channel Adapter
   Emulex LightPulse LPe36002-M64 2-Port 64Gb Fibre Channel Adapter
   ```

2. Verify that you are using the recommended Broadcom `lpfc` firmware and inbox driver:

   a. Display the firmware version:

   ```
   cat /sys/class/scsi_host/host*/fwrev
   ```

   The following example shows firmware versions:

   ```
   14.4.317.10, sli-4:6:d
   14.4.317.10, sli-4:6:d
   ```

   b. Display the inbox driver version:

   ```
   cat /sys/module/lpfc/version`
   ```

   The following example shows a driver version:

```
0:14.4.0.2
```

For the current list of supported adapter driver and firmware versions, see the Interoperability Matrix Tool.

3. Verify that the expected output of `lpfc_enable_fc4_type` is set to `3`:

```
cat /sys/module/lpfc/parameters/lpfc_enable_fc4_type
```

4. Verify that you can view your initiator ports:

```
cat /sys/class/fc_host/host*/port_name
```

The following example shows port identities:

```
0x100000109bf044b1
0x100000109bf044b2
```

5. Verify that your initiator ports are online:

```
cat /sys/class/fc_host/host*/port_state
```

You should see the following output:

```
Online
Online
```

6. Verify that the NVMe/FC initiator ports are enabled and that the target ports are visible:

```
cat /sys/class/scsi_host/host*/nvme_info
```

Show example

```
NVME Initiator Enabled
XRI Dist lpfc2 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc2 WWPN x100000109bf044b1 WWNN x200000109bf044b1
DID x022a00 ONLINE
NVME RPORT       WWPN x202fd039eaa7dfc8 WWNN x202cd039eaa7dfc8
DID x021310 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x202dd039eaa7dfc8 WWNN x202cd039eaa7dfc8
DID x020b10 TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 0000000810 Cmpl 0000000810 Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000007b098f07 Issue 000000007aee27c4 OutIO
ffffffffffe498bd
        abort 000013b4 noxri 00000000 nondlp 00000058 qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 000013b4 Err 00021443

NVME Initiator Enabled
XRI Dist lpfc3 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc3 WWPN x100000109bf044b2 WWNN x200000109bf044b2
DID x021b00 ONLINE
NVME RPORT       WWPN x2033d039eaa7dfc8 WWNN x202cd039eaa7dfc8
DID x020110 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2032d039eaa7dfc8 WWNN x202cd039eaa7dfc8
DID x022910 TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 0000000840 Cmpl 0000000840 Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000007afd4434 Issue 000000007ae31b83 OutIO
ffffffffffe5d74f
        abort 000014a5 noxri 00000000 nondlp 0000006a qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 000014a5 Err 0002149a
```

Marvell/QLogic

Configure NVMe/FC for a Marvell/QLogic adapter.

> ⓘ The native inbox qla2xxx driver included in the Rocky Linux kernel has the latest fixes. These fixes are essential for ONTAP support.

Steps

1. Verify that you are running the supported adapter driver and firmware versions:

```
cat /sys/class/fc_host/host*/symbolic_name
```

   The follow example shows driver and firmware versions:

```
QLE2742 FW:v9.14.00 DVR:v10.02.09.200-k
QLE2742 FW:v9.14.00 DVR:v10.02.09.200-k
```

2. Verify that `ql2xnvmeenable` is set. This enables the Marvell adapter to function as an NVMe/FC initiator:

```
cat /sys/module/qla2xxx/parameters/ql2xnvmeenable
```

   The expected output is 1.

Step 4: Optionally, enable 1MB I/O

You can enable I/O requests of size 1MB for NVMe/FC configured with a Broadcom adapter. ONTAP reports a Max Data Transfer Size (MDTS) of 8 in the Identify Controller data. This means the maximum I/O request size can be up to 1MB. To issue I/O requests of size 1MB, you need to increase the lpfc value of the `lpfc_sg_seg_cnt` parameter to 256 from the default value of 64.

> These steps don't apply to Qlogic NVMe/FC hosts.

Steps

1. Set the `lpfc_sg_seg_cnt` parameter to 256:

```
cat /etc/modprobe.d/lpfc.conf
```

```
options lpfc lpfc_sg_seg_cnt=256
```

2. Run the `dracut -f` command, and reboot the host.
3. Verify that the value for `lpfc_sg_seg_cnt` is 256:

```
cat /sys/module/lpfc/parameters/lpfc_sg_seg_cnt
```

Step 5: Configure NVMe/TCP

The NVMe/TCP protocol doesn’t support the auto-connect operation. Instead, you can discover the NVMe/TCP subsystems and namespaces by performing the NVMe/TCP connect or connect-all operations manually.

Steps

1. Verify that the initiator port can fetch the discovery log page data across the supported NVMe/TCP LIFs:

```
nvme discover -t tcp -w host-traddr -a traddr
```

Show example

```
nvme discover -t tcp -w 192.168.1.31 -a 192.168.1.24
Discovery Log Number of Records 20, Generation counter 25
=====Discovery Log Entry 0======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  4
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr:  192.168.2.25
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 1======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  2
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr:  192.168.1.25
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 2======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  5
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr:  192.168.2.24
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 3======
trtype:  tcp
adrfam:  ipv4
```

```
subtype: current discovery subsystem
treq:    not specified
portid:  1
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr:  192.168.1.24
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 4======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_1
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 5======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_1
traddr:  192.168.1.25
eflags:  none
sectype: none
=====Discovery Log Entry 6======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  5
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_1
```

```
traddr:  192.168.2.24
eflags:  none
sectype: none
=====Discovery Log Entry 7======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_1
traddr:  192.168.1.24
eflags:  none
sectype: none
=====Discovery Log Entry 8======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_4
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 9======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_4
traddr:  192.168.1.25
eflags:  none
sectype: none
=====Discovery Log Entry 10======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
```

```
treq:    not specified
portid:  5
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_4
traddr:  192.168.2.24
eflags:  none
sectype: none
=====Discovery Log Entry 11======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_4
traddr:  192.168.1.24
eflags:  none
sectype: none
=====Discovery Log Entry 12======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_3
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 13======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_3
traddr:  192.168.1.25
```

```
eflags:  none
sectype: none
=====Discovery Log Entry 14======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  5
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_3
traddr:  192.168.2.24
eflags:  none
sectype: none
=====Discovery Log Entry 15======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_3
traddr:  192.168.1.24
eflags:  none
sectype: none
=====Discovery Log Entry 16======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_2
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 17======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
```

```
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_2
traddr:  192.168.1.25
eflags:  none
sectype: none
=====Discovery Log Entry 18======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  5
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_2
traddr:  192.168.2.24
eflags:  none
sectype: none
=====Discovery Log Entry 19======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_2
traddr:  192.168.1.24
eflags:  none
sectype: none
```

2. Verify that the other NVMe/TCP initiator-target LIF combinations are able to successfully fetch discovery log page data:

```
nvme discover -t tcp -w host-traddr -a traddr
```

Show example

```
nvme discover -t tcp -w 192.168.1.31 -a 192.168.1.24
nvme discover -t tcp -w 192.168.2.31 -a 192.168.2.24
nvme discover -t tcp -w 192.168.1.31 -a 192.168.1.25
nvme discover -t tcp -w 192.168.2.31 -a 192.168.2.25
```

3. Run the `nvme connect-all` command across all the supported NVMe/TCP initiator-target LIFs across the nodes:

```
nvme connect-all -t tcp -w host-traddr -a traddr
```

Show example

```
nvme    connect-all -t  tcp -w  192.168.1.31    -a  192.168.1.24
nvme    connect-all -t  tcp -w  192.168.2.31    -a  192.168.2.24
nvme    connect-all -t  tcp -w  192.168.1.31    -a  192.168.1.25
nvme    connect-all -t  tcp -w  192.168.2.31    -a  192.168.2.25
```

Step 6: Validate NVMe-oF

Verify that the in-kernel NVMe multipath status, ANA status, and ONTAP namespaces are correct for the NVMe-oF configuration.

Steps

1. Verify that the in-kernel NVMe multipath is enabled:

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

You should see the following output:

```
Y
```

2. Verify that the appropriate NVMe-oF settings (such as, model set to NetApp ONTAP Controller and load balancing iopolicy set to round-robin) for the respective ONTAP namespaces correctly reflect on the host:
    a. Display the subsystems:

    ```
    cat /sys/class/nvme-subsystem/nvme-subsys*/model
    ```

You should see the following output:

```
NetApp ONTAP Controller
NetApp ONTAP Controller
```

b. Display the policy:

```
cat /sys/class/nvme-subsystem/nvme-subsys*/iopolicy
```

You should see the following output:

```
round-robin
round-robin
```

3. Verify that the namespaces are created and correctly discovered on the host:

```
nvme list
```

Show example

```
Node         SN                   Model
---------------------------------------------------------
/dev/nvme4n1 81Ix2BVuekWcAAAAAAAB   NetApp ONTAP Controller


Namespace Usage    Format             FW             Rev
-----------------------------------------------------------
1                 21.47 GB / 21.47 GB   4 KiB + 0 B   FFFFFFFF
```

4. Verify that the controller state of each path is live and has the correct ANA status:

NVMe/FC

```
nvme list-subsys /dev/nvme4n5
```

Show example

```
nvme-subsys4 - NQN=nqn.1992-
08.com.netapp:sn.3a5d31f5502c11ef9f50d039eab6cb6d:subsystem.nvme
_1
               hostnqn=nqn.2014-08.org.nvmexpress:uuid:e6dade64-
216d-
11ec-b7bb-7ed30a5482c3
iopolicy=round-robin\
+- nvme1 fc traddr=nn-0x2082d039eaa7dfc8:pn-
0x2088d039eaa7dfc8,host_traddr=nn-0x20000024ff752e6d:pn-
0x21000024ff752e6d live optimized
+- nvme12 fc traddr=nn-0x2082d039eaa7dfc8:pn-
0x208ad039eaa7dfc8,host_traddr=nn-0x20000024ff752e6d:pn-
0x21000024ff752e6d live non-optimized
+- nvme10 fc traddr=nn-0x2082d039eaa7dfc8:pn-
0x2087d039eaa7dfc8,host_traddr=nn-0x20000024ff752e6c:pn-
0x21000024ff752e6c live non-optimized
+- nvme3 fc traddr=nn-0x2082d039eaa7dfc8:pn-
0x2083d039eaa7dfc8,host_traddr=nn-0x20000024ff752e6c:pn-
0x21000024ff752e6c live optimized
```

NVMe/TCP

```
nvme list-subsys /dev/nvme1n1
```

Show example

```
nvme-subsys5 - NQN=nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme
_tcp_3
hostnqn=nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0035-5910-804b-
b5c04f444d33
iopolicy=round-robin
\
+- nvme13 tcp
traddr=192.168.2.25,trsvcid=4420,host_traddr=192.168.2.31,
src_addr=192.168.2.31 live optimized
+- nvme14 tcp
traddr=192.168.2.24,trsvcid=4420,host_traddr=192.168.2.31,
src_addr=192.168.2.31 live non-optimized
+- nvme5 tcp
traddr=192.168.1.25,trsvcid=4420,host_traddr=192.168.1.31,
src_addr=192.168.1.31 live optimized
+- nvme6 tcp
traddr=192.168.1.24,trsvcid=4420,host_traddr=192.168.1.31,
src_addr=192.168.1.31 live non-optimized
```

5. Verify that the NetApp plug-in displays the correct values for each ONTAP namespace device:

Column

```
nvme netapp ontapdevices -o column
```

Show example

```
Device        Vserver   Namespace Path
----------------------- ------------------------------
/dev/nvme1n1     linux_tcnvme_iscsi
/vol/tcpnvme_1_0_0/tcpnvme_ns

NSID       UUID                                   Size
------------------------------------------------------------
1    5f7f630d-8ea5-407f-a490-484b95b15dd6   21.47GB
```

JSON

```
nvme netapp ontapdevices -o json
```

Show example

```
{
  "ONTAPdevices":[
    {
      "Device":"/dev/nvme1n1",
      "Vserver":"linux_tcnvme_iscsi",
      "Namespace_Path":"/vol/tcpnvme_1_0_0/tcpnvme_ns",
      "NSID":1,
      "UUID":"5f7f630d-8ea5-407f-a490-484b95b15dd6",
      "Size":"21.47GB",
      "LBA_Data_Size":4096,
      "Namespace_Size":5242880
    },
]
}
```

Step 7: Review the known issues

The NVMe-oF host configuration for Rocky Linux 8.6 with ONTAP has the following known issues:

NetApp Bug ID	Title	Description
1479047	Rocky Linux 8.6 NVMe-oF hosts create duplicate Persistent Discovery Controllers	On NVMe over Fabrics (NVMe-oF) hosts, you can use the "nvme discover -p" command to create Persistent Discovery Controllers (PDCs). When this command is used, only one PDC should be created per initiator-target combination. However, if you are running ONTAP 9.10.1 and Rocky Linux 8.6 with an NVMe-oF host, a duplicate PDC is created each time "nvme discover -p" is executed. This leads to unnecessary usage of resources on both the host and the target.

Configure Rocky Linux 8.5 with NVMe-oF for ONTAP storage

Rocky Linux 8.5 hosts support the NVMe/FC and NVMe/TCP protocols with Asymmetric Namespace Access (ANA). ANA is equivalent to asymmetric logical unit access (ALUA) multipathing in iSCSI and FCP environments and is implemented using the in-kernel NVMe multipath feature.

For additional details on supported configurations, see the Interoperability Matrix Tool.

About this task

You can use the following support and features with the NVMe-oF host configuration for Rocky Linux 8.5. You should also review the known limitations before starting the configuration process.

- Support available:
  - Support for NVMe over TCP (NVMe/TCP) in addition to NVMe over Fibre Channel (NVMe/FC). The NetApp plug-in in the native `nvme-cli` package displays ONTAP details for both NVMe/FC and NVMe/TCP namespaces.
  - Running NVMe and SCSI traffic on the same host. For example, you can configure dm-multipath for SCSI mpath devices on SCSI LUNs and use NVMe multipath to configure NVMe-oF namespace devices on the host.
- Known limitations:
  - In-kernel NVMe multipath is disabled by default for Rocky Linux 8.5 NVMe-oF hosts. Therefore, you need to enable it manually.
  - On Rocky Linux 8.5 hosts, NVMe/TCP is a technology preview feature due to open issues. Refer to the Rocky Linux 8.5 Release Notes for details.
  - SAN booting using the NVMe-oF protocol is currently not supported.

Step 1: Optionally, enable SAN booting

You can configure your host to use SAN booting to simplify deployment and improve scalability.

Before you begin

Use the Interoperability Matrix Tool to verify that your Linux OS, host bus adapter (HBA), HBA firmware, HBA boot BIOS, and ONTAP version support SAN booting.

Steps

1. Create a SAN boot namespace and map it to the host.
2. Enable SAN booting in the server BIOS for the ports to which the SAN boot namespace is mapped.

   For information on how to enable the HBA BIOS, see your vendor-specific documentation.

3. Verify that the configuration was successful by rebooting the host and verifying that the OS is up and running.

Step 2: Validate software versions

Use the following procedure to validate the minimum supported Rocky Linux 8.5 software versions.

Steps

1. Install Rocky Linux 8.5 on the server. After the installation is complete, verify that you are running the specified Rocky Linux 8.5 kernel:

   ```
   uname -r
   ```

   The following example shows a Rocky Linux kernel version:

   ```
   5.14.0-570.12.1.el9_6.x86_64
   ```

2. Install the `nvme-cli` package:

   ```
   rpm -qa|grep nvme-cli
   ```

   The following example shows an nvme-cli package version:

   ```
   nvme-cli-2.11-5.el9.x86_64
   ```

3. Install the `libnvme` package:

   ```
   rpm -qa|grep libnvme
   ```

   The following example shows an `libnvme` package version:

   ```
   libnvme-1.11.1-1.el9.x86_64
   ```

4. On the Rocky Linux host, check the hostnqn string at `/etc/nvme/hostnqn`:

```
cat /etc/nvme/hostnqn
```

The following example shows an `hostnqn` version:

```
nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
```

5. Verify that the `hostnqn` string matches the `hostnqn` string for the corresponding subsystem on the ONTAP array:

```
::> vserver nvme subsystem host show -vserver vs_coexistence_LPE36002
```

Show example

```
Vserver Subsystem Priority  Host NQN
------- --------- --------
------------------------------------------------
vs_coexistence_LPE36002
        nvme
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
        nvme_1
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
        nvme_2
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
        nvme_3
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
4 entries were displayed.
```

If the `hostnqn` strings do not match, use the `vserver modify` command to update the `hostnqn` string on your corresponding ONTAP array subsystem to match the `hostnqn` string from `/etc/nvme/hostnqn` on the host.

Step 3: Configure NVMe/FC

You can configure NVMe/FC with Broadcom/Emulex FC or Marvell/Qlogic FC adapters. You also need to manually discover the NVMe/TCP subsystems and namespaces.

Broadcom/Emulex

Configure NVMe/FC for a Broadcom/Emulex adapter.

Steps

1. Verify that you are using the supported adapter model:

   a. Display the model names:

   ```
   cat /sys/class/scsi_host/host*/modelname
   ```

   You should see the following output:

   ```
   LPe36002-M64
   LPe36002-M64
   ```

   b. Display the model descriptions:

   ```
   cat /sys/class/scsi_host/host*/modeldesc
   ```

   You should see an output similar to the following example:

   ```
   Emulex LightPulse LPe36002-M64 2-Port 64Gb Fibre Channel Adapter
   Emulex LightPulse LPe36002-M64 2-Port 64Gb Fibre Channel Adapter
   ```

2. Verify that you are using the recommended Broadcom `lpfc` firmware and inbox driver:

   a. Display the firmware version:

   ```
   cat /sys/class/scsi_host/host*/fwrev
   ```

   The following example shows firmware versions:

   ```
   14.4.317.10, sli-4:6:d
   14.4.317.10, sli-4:6:d
   ```

   b. Display the inbox driver version:

   ```
   cat /sys/module/lpfc/version`
   ```

   The following example shows a driver version:

```
0:14.4.0.2
```

For the current list of supported adapter driver and firmware versions, see the Interoperability Matrix Tool.

3. Verify that the expected output of `lpfc_enable_fc4_type` is set to `3`:

```
cat /sys/module/lpfc/parameters/lpfc_enable_fc4_type
```

4. Verify that you can view your initiator ports:

```
cat /sys/class/fc_host/host*/port_name
```

The following example shows port identities:

```
0x100000109bf044b1
0x100000109bf044b2
```

5. Verify that your initiator ports are online:

```
cat /sys/class/fc_host/host*/port_state
```

You should see the following output:

```
Online
Online
```

6. Verify that the NVMe/FC initiator ports are enabled and that the target ports are visible:

```
cat /sys/class/scsi_host/host*/nvme_info
```

Show example

```
NVME Initiator Enabled
XRI Dist lpfc2 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc2 WWPN x100000109bf044b1 WWNN x200000109bf044b1
DID x022a00 ONLINE
NVME RPORT       WWPN x202fd039eaa7dfc8 WWNN x202cd039eaa7dfc8
DID x021310 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x202dd039eaa7dfc8 WWNN x202cd039eaa7dfc8
DID x020b10 TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 0000000810 Cmpl 0000000810 Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000007b098f07 Issue 000000007aee27c4 OutIO
fffffffffe498bd
        abort 000013b4 noxri 00000000 nondlp 00000058 qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 000013b4 Err 00021443

NVME Initiator Enabled
XRI Dist lpfc3 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc3 WWPN x100000109bf044b2 WWNN x200000109bf044b2
DID x021b00 ONLINE
NVME RPORT       WWPN x2033d039eaa7dfc8 WWNN x202cd039eaa7dfc8
DID x020110 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2032d039eaa7dfc8 WWNN x202cd039eaa7dfc8
DID x022910 TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 0000000840 Cmpl 0000000840 Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000007afd4434 Issue 000000007ae31b83 OutIO
fffffffffe5d74f
        abort 000014a5 noxri 00000000 nondlp 0000006a qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 000014a5 Err 0002149a
```

Marvell/QLogic

Configure NVMe/FC for a Marvell/QLogic adapter.

The native inbox qla2xxx driver included in the Rocky Linux kernel has the latest fixes. These fixes are essential for ONTAP support.

Steps

1. Verify that you are running the supported adapter driver and firmware versions:

```
cat /sys/class/fc_host/host*/symbolic_name
```

The follow example shows driver and firmware versions:

```
QLE2742 FW:v9.14.00 DVR:v10.02.09.200-k
QLE2742 FW:v9.14.00 DVR:v10.02.09.200-k
```

2. Verify that `ql2xnvmeenable` is set. This enables the Marvell adapter to function as an NVMe/FC initiator:

```
cat /sys/module/qla2xxx/parameters/ql2xnvmeenable
```

The expected output is 1.

Step 4: Optionally, enable 1MB I/O

You can enable I/O requests of size 1MB for NVMe/FC configured with a Broadcom adapter. ONTAP reports a Max Data Transfer Size (MDTS) of 8 in the Identify Controller data. This means the maximum I/O request size can be up to 1MB. To issue I/O requests of size 1MB, you need to increase the lpfc value of the `lpfc_sg_seg_cnt` parameter to 256 from the default value of 64.

> These steps don't apply to Qlogic NVMe/FC hosts.

Steps

1. Set the `lpfc_sg_seg_cnt` parameter to 256:

```
cat /etc/modprobe.d/lpfc.conf
```

```
options lpfc lpfc_sg_seg_cnt=256
```

2. Run the `dracut -f` command, and reboot the host.
3. Verify that the value for `lpfc_sg_seg_cnt` is 256:

```
cat /sys/module/lpfc/parameters/lpfc_sg_seg_cnt
```

Step 5: Configure NVMe/TCP

The NVMe/TCP protocol doesn’t support the auto-connect operation. Instead, you can discover the NVMe/TCP subsystems and namespaces by performing the NVMe/TCP connect or connect-all operations manually.

Steps

1. Verify that the initiator port can fetch the discovery log page data across the supported NVMe/TCP LIFs:

```
nvme discover -t tcp -w host-traddr -a traddr
```

Show example

```
nvme discover -t tcp -w 192.168.1.31 -a 192.168.1.24
Discovery Log Number of Records 20, Generation counter 25
=====Discovery Log Entry 0======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  4
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr:  192.168.2.25
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 1======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  2
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr:  192.168.1.25
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 2======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  5
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr:  192.168.2.24
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 3======
trtype:  tcp
adrfam:  ipv4
```

```
subtype: current discovery subsystem
treq:    not specified
portid:  1
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr:  192.168.1.24
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 4======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_1
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 5======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_1
traddr:  192.168.1.25
eflags:  none
sectype: none
=====Discovery Log Entry 6======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  5
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_1
```

```
traddr:  192.168.2.24
eflags:  none
sectype: none
=====Discovery Log Entry 7======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_1
traddr:  192.168.1.24
eflags:  none
sectype: none
=====Discovery Log Entry 8======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_4
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 9======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_4
traddr:  192.168.1.25
eflags:  none
sectype: none
=====Discovery Log Entry 10======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
```

```
treq:    not specified
portid:  5
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_4
traddr:  192.168.2.24
eflags:  none
sectype: none
=====Discovery Log Entry 11======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_4
traddr:  192.168.1.24
eflags:  none
sectype: none
=====Discovery Log Entry 12======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_3
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 13======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_3
traddr:  192.168.1.25
```

```
eflags:  none
sectype: none
=====Discovery Log Entry 14======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  5
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_3
traddr:  192.168.2.24
eflags:  none
sectype: none
=====Discovery Log Entry 15======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_3
traddr:  192.168.1.24
eflags:  none
sectype: none
=====Discovery Log Entry 16======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_2
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 17======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
```

```
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_2
traddr:  192.168.1.25
eflags:  none
sectype: none
=====Discovery Log Entry 18======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  5
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_2
traddr:  192.168.2.24
eflags:  none
sectype: none
=====Discovery Log Entry 19======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_2
traddr:  192.168.1.24
eflags:  none
sectype: none
```

2. Verify that the other NVMe/TCP initiator-target LIF combinations are able to successfully fetch discovery log page data:

```
nvme discover -t tcp -w host-traddr -a traddr
```

Show example

```
nvme discover -t tcp -w 192.168.1.31 -a 192.168.1.24
nvme discover -t tcp -w 192.168.2.31 -a 192.168.2.24
nvme discover -t tcp -w 192.168.1.31 -a 192.168.1.25
nvme discover -t tcp -w 192.168.2.31 -a 192.168.2.25
```

3. Run the `nvme connect-all` command across all the supported NVMe/TCP initiator-target LIFs across the nodes:

```
nvme connect-all -t tcp -w host-traddr -a traddr
```

Show example

```
nvme    connect-all -t  tcp -w  192.168.1.31    -a  192.168.1.24
nvme    connect-all -t  tcp -w  192.168.2.31    -a  192.168.2.24
nvme    connect-all -t  tcp -w  192.168.1.31    -a  192.168.1.25
nvme    connect-all -t  tcp -w  192.168.2.31    -a  192.168.2.25
```

Step 6: Validate NVMe-oF

Verify that the in-kernel NVMe multipath status, ANA status, and ONTAP namespaces are correct for the NVMe-oF configuration.

Steps

1. Verify that the in-kernel NVMe multipath is enabled:

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

You should see the following output:

```
Y
```

2. Verify that the appropriate NVMe-oF settings (such as, model set to NetApp ONTAP Controller and load balancing iopolicy set to round-robin) for the respective ONTAP namespaces correctly reflect on the host:

   a. Display the subsystems:

```
cat /sys/class/nvme-subsystem/nvme-subsys*/model
```

You should see the following output:

```
NetApp ONTAP Controller
NetApp ONTAP Controller
```

b. Display the policy:

```
cat /sys/class/nvme-subsystem/nvme-subsys*/iopolicy
```

You should see the following output:

```
round-robin
round-robin
```

3. Verify that the namespaces are created and correctly discovered on the host:

```
nvme list
```

Show example

```
Node         SN                   Model
---------------------------------------------------------
/dev/nvme4n1 81Ix2BVuekWcAAAAAAAB   NetApp ONTAP Controller


Namespace Usage    Format             FW             Rev
-----------------------------------------------------------
1                 21.47 GB / 21.47 GB   4 KiB + 0 B   FFFFFFFF
```

4. Verify that the controller state of each path is live and has the correct ANA status:

NVMe/FC

```
nvme list-subsys /dev/nvme4n5
```

Show example

```
nvme-subsys4 - NQN=nqn.1992-
08.com.netapp:sn.3a5d31f5502c11ef9f50d039eab6cb6d:subsystem.nvme
_1
               hostnqn=nqn.2014-08.org.nvmexpress:uuid:e6dade64-
216d-
11ec-b7bb-7ed30a5482c3
iopolicy=round-robin\
+- nvme1 fc traddr=nn-0x2082d039eaa7dfc8:pn-
0x2088d039eaa7dfc8,host_traddr=nn-0x20000024ff752e6d:pn-
0x21000024ff752e6d live optimized
+- nvme12 fc traddr=nn-0x2082d039eaa7dfc8:pn-
0x208ad039eaa7dfc8,host_traddr=nn-0x20000024ff752e6d:pn-
0x21000024ff752e6d live non-optimized
+- nvme10 fc traddr=nn-0x2082d039eaa7dfc8:pn-
0x2087d039eaa7dfc8,host_traddr=nn-0x20000024ff752e6c:pn-
0x21000024ff752e6c live non-optimized
+- nvme3 fc traddr=nn-0x2082d039eaa7dfc8:pn-
0x2083d039eaa7dfc8,host_traddr=nn-0x20000024ff752e6c:pn-
0x21000024ff752e6c live optimized
```

NVMe/TCP

```
nvme list-subsys /dev/nvme1n1
```

Show example

```
nvme-subsys5 - NQN=nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme
_tcp_3
hostnqn=nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0035-5910-804b-
b5c04f444d33
iopolicy=round-robin
\
+- nvme13 tcp
traddr=192.168.2.25,trsvcid=4420,host_traddr=192.168.2.31,
src_addr=192.168.2.31 live optimized
+- nvme14 tcp
traddr=192.168.2.24,trsvcid=4420,host_traddr=192.168.2.31,
src_addr=192.168.2.31 live non-optimized
+- nvme5 tcp
traddr=192.168.1.25,trsvcid=4420,host_traddr=192.168.1.31,
src_addr=192.168.1.31 live optimized
+- nvme6 tcp
traddr=192.168.1.24,trsvcid=4420,host_traddr=192.168.1.31,
src_addr=192.168.1.31 live non-optimized
```

5. Verify that the NetApp plug-in displays the correct values for each ONTAP namespace device:

Column

```
nvme netapp ontapdevices -o column
```

Show example

```
Device         Vserver   Namespace Path
----------------------- ------------------------------
/dev/nvme1n1     linux_tcnvme_iscsi
/vol/tcpnvme_1_0_0/tcpnvme_ns

NSID       UUID                                   Size
------------------------------------------------------------
1    5f7f630d-8ea5-407f-a490-484b95b15dd6   21.47GB
```

JSON

```
nvme netapp ontapdevices -o json
```

Show example

```
{
  "ONTAPdevices":[
    {
      "Device":"/dev/nvme1n1",
      "Vserver":"linux_tcnvme_iscsi",
      "Namespace_Path":"/vol/tcpnvme_1_0_0/tcpnvme_ns",
      "NSID":1,
      "UUID":"5f7f630d-8ea5-407f-a490-484b95b15dd6",
      "Size":"21.47GB",
      "LBA_Data_Size":4096,
      "Namespace_Size":5242880
    },
]
}
```

Step 7: Review the known issues

No known issues exist for the NVMe-oF host configuration on Rocky Linux 8.5 with ONTAP release.

Configure Rocky Linux 8.4 with NVMe-oF for ONTAP storage

Rocky Linux 8.4 hosts support the NVMe/FC and NVMe/TCP protocols with Asymmetric Namespace Access (ANA). ANA is equivalent to asymmetric logical unit access (ALUA) multipathing in iSCSI and FCP environments and is implemented using the in-kernel NVMe multipath feature.

For additional details on supported configurations, see the Interoperability Matrix Tool.

About this task

You can use the following support and features with the NVMe-oF host configuration for Rocky Linux 8.4. You should also review the known limitations before starting the configuration process.

- Support available:
  - Support for NVMe over TCP (NVMe/TCP) in addition to NVMe over Fibre Channel (NVMe/FC). The NetApp plug-in in the native `nvme-cli` package displays ONTAP details for both NVMe/FC and NVMe/TCP namespaces.
  - Running NVMe and SCSI traffic on the same host. For example, you can configure dm-multipath for SCSI mpath devices on SCSI LUNs and use NVMe multipath to configure NVMe-oF namespace devices on the host.
- Known limitations:
  - In-kernel NVMe multipath is disabled by default for Rocky Linux 8.4 NVMe-oF hosts. Therefore, you need to enable it manually.
  - On Rocky Linux 8.4 hosts, NVMe/TCP is a technology preview feature due to open issues. Refer to the Rocky Linux 8.4 Release Notes for details.
  - SAN booting using the NVMe-oF protocol is currently not supported.

Step 1: Optionally, enable SAN booting

You can configure your host to use SAN booting to simplify deployment and improve scalability.

Before you begin

Use the Interoperability Matrix Tool to verify that your Linux OS, host bus adapter (HBA), HBA firmware, HBA boot BIOS, and ONTAP version support SAN booting.

Steps

1. Create a SAN boot namespace and map it to the host.
2. Enable SAN booting in the server BIOS for the ports to which the SAN boot namespace is mapped.

   For information on how to enable the HBA BIOS, see your vendor-specific documentation.

3. Verify that the configuration was successful by rebooting the host and verifying that the OS is up and running.

Step 2: Validate software versions

Use the following procedure to validate the minimum supported Rocky Linux 8.4 software versions.

Steps

1. Install Rocky Linux 8.4 on the server. After the installation is complete, verify that you are running the

specified Rocky Linux 8.4 kernel:

```
uname -r
```

The following example shows a Rocky Linux kernel version:

```
5.14.0-570.12.1.el9_6.x86_64
```

2. Install the `nvme-cli` package:

```
rpm -qa|grep nvme-cli
```

The following example shows an nvme-cli package version:

```
nvme-cli-2.11-5.el9.x86_64
```

3. Install the `libnvme` package:

```
rpm -qa|grep libnvme
```

The following example shows an `libnvme` package version:

```
libnvme-1.11.1-1.el9.x86_64
```

4. On the Rocky Linux host, check the hostnqn string at `/etc/nvme/hostnqn`:

```
cat /etc/nvme/hostnqn
```

The following example shows an `hostnqn` version:

```
nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
```

5. Verify that the `hostnqn` string matches the `hostnqn` string for the corresponding subsystem on the ONTAP array:

```
::> vserver nvme subsystem host show -vserver vs_coexistence_LPE36002
```

Show example

```
Vserver Subsystem Priority  Host NQN
------- --------- --------
------------------------------------------------
vs_coexistence_LPE36002
        nvme
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
        nvme_1
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
        nvme_2
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
        nvme_3
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
4 entries were displayed.
```

If the `hostnqn` strings do not match, use the `vserver modify` command to update the `hostnqn` string on your corresponding ONTAP array subsystem to match the `hostnqn` string from `/etc/nvme/hostnqn` on the host.

Step 3: Configure NVMe/FC

You can configure NVMe/FC with Broadcom/Emulex FC or Marvell/Qlogic FC adapters. You also need to manually discover the NVMe/TCP subsystems and namespaces.

Broadcom/Emulex

Configure NVMe/FC for a Broadcom/Emulex adapter.

Steps

1. Verify that you are using the supported adapter model:

   a. Display the model names:

   ```
   cat /sys/class/scsi_host/host*/modelname
   ```

   You should see the following output:

   ```
   LPe36002-M64
   LPe36002-M64
   ```

   b. Display the model descriptions:

   ```
   cat /sys/class/scsi_host/host*/modeldesc
   ```

   You should see an output similar to the following example:

   ```
   Emulex LightPulse LPe36002-M64 2-Port 64Gb Fibre Channel Adapter
   Emulex LightPulse LPe36002-M64 2-Port 64Gb Fibre Channel Adapter
   ```

2. Verify that you are using the recommended Broadcom `lpfc` firmware and inbox driver:

   a. Display the firmware version:

   ```
   cat /sys/class/scsi_host/host*/fwrev
   ```

   The following example shows firmware versions:

   ```
   14.4.317.10, sli-4:6:d
   14.4.317.10, sli-4:6:d
   ```

   b. Display the inbox driver version:

   ```
   cat /sys/module/lpfc/version`
   ```

   The following example shows a driver version:

```
0:14.4.0.2
```

For the current list of supported adapter driver and firmware versions, see the Interoperability Matrix Tool.

3. Verify that the expected output of `lpfc_enable_fc4_type` is set to `3`:

   ```
   cat /sys/module/lpfc/parameters/lpfc_enable_fc4_type
   ```

4. Verify that you can view your initiator ports:

   ```
   cat /sys/class/fc_host/host*/port_name
   ```

   The following example shows port identities:

   ```
   0x100000109bf044b1
   0x100000109bf044b2
   ```

5. Verify that your initiator ports are online:

   ```
   cat /sys/class/fc_host/host*/port_state
   ```

   You should see the following output:

   ```
   Online
   Online
   ```

6. Verify that the NVMe/FC initiator ports are enabled and that the target ports are visible:

   ```
   cat /sys/class/scsi_host/host*/nvme_info
   ```

Show example

```
NVME Initiator Enabled
XRI Dist lpfc2 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc2 WWPN x100000109bf044b1 WWNN x200000109bf044b1
DID x022a00 ONLINE
NVME RPORT       WWPN x202fd039eaa7dfc8 WWNN x202cd039eaa7dfc8
DID x021310 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x202dd039eaa7dfc8 WWNN x202cd039eaa7dfc8
DID x020b10 TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 0000000810 Cmpl 0000000810 Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000007b098f07 Issue 000000007aee27c4 OutIO
fffffffffe498bd
        abort 000013b4 noxri 00000000 nondlp 00000058 qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 000013b4 Err 00021443

NVME Initiator Enabled
XRI Dist lpfc3 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc3 WWPN x100000109bf044b2 WWNN x200000109bf044b2
DID x021b00 ONLINE
NVME RPORT       WWPN x2033d039eaa7dfc8 WWNN x202cd039eaa7dfc8
DID x020110 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2032d039eaa7dfc8 WWNN x202cd039eaa7dfc8
DID x022910 TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 0000000840 Cmpl 0000000840 Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000007afd4434 Issue 000000007ae31b83 OutIO
fffffffffe5d74f
        abort 000014a5 noxri 00000000 nondlp 0000006a qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 000014a5 Err 0002149a
```

Marvell/QLogic

Configure NVMe/FC for a Marvell/QLogic adapter.

> ⓘ The native inbox qla2xxx driver included in the Rocky Linux kernel has the latest fixes. These fixes are essential for ONTAP support.

Steps

1. Verify that you are running the supported adapter driver and firmware versions:

```
cat /sys/class/fc_host/host*/symbolic_name
```

The follow example shows driver and firmware versions:

```
QLE2742 FW:v9.14.00 DVR:v10.02.09.200-k
QLE2742 FW:v9.14.00 DVR:v10.02.09.200-k
```

2. Verify that `ql2xnvmeenable` is set. This enables the Marvell adapter to function as an NVMe/FC initiator:

```
cat /sys/module/qla2xxx/parameters/ql2xnvmeenable
```

The expected output is 1.

Step 4: Optionally, enable 1MB I/O

You can enable I/O requests of size 1MB for NVMe/FC configured with a Broadcom adapter. ONTAP reports a Max Data Transfer Size (MDTS) of 8 in the Identify Controller data. This means the maximum I/O request size can be up to 1MB. To issue I/O requests of size 1MB, you need to increase the lpfc value of the `lpfc_sg_seg_cnt` parameter to 256 from the default value of 64.

> These steps don't apply to Qlogic NVMe/FC hosts.

Steps

1. Set the `lpfc_sg_seg_cnt` parameter to 256:

```
cat /etc/modprobe.d/lpfc.conf
```

```
options lpfc lpfc_sg_seg_cnt=256
```

2. Run the `dracut -f` command, and reboot the host.
3. Verify that the value for `lpfc_sg_seg_cnt` is 256:

```
cat /sys/module/lpfc/parameters/lpfc_sg_seg_cnt
```

Step 5: Configure NVMe/TCP

The NVMe/TCP protocol doesn’t support the auto-connect operation. Instead, you can discover the NVMe/TCP subsystems and namespaces by performing the NVMe/TCP connect or connect-all operations manually.

Steps

1. Verify that the initiator port can fetch the discovery log page data across the supported NVMe/TCP LIFs:

   ```
   nvme discover -t tcp -w host-traddr -a traddr
   ```

Show example

```
nvme discover -t tcp -w 192.168.1.31 -a 192.168.1.24
Discovery Log Number of Records 20, Generation counter 25
=====Discovery Log Entry 0======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  4
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr:  192.168.2.25
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 1======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  2
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr:  192.168.1.25
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 2======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  5
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr:  192.168.2.24
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 3======
trtype:  tcp
adrfam:  ipv4
```

```
subtype: current discovery subsystem
treq:    not specified
portid:  1
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr:  192.168.1.24
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 4======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_1
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 5======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_1
traddr:  192.168.1.25
eflags:  none
sectype: none
=====Discovery Log Entry 6======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  5
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_1
```

```
traddr:  192.168.2.24
eflags:  none
sectype: none
=====Discovery Log Entry 7======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_1
traddr:  192.168.1.24
eflags:  none
sectype: none
=====Discovery Log Entry 8======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_4
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 9======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_4
traddr:  192.168.1.25
eflags:  none
sectype: none
=====Discovery Log Entry 10======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
```

```
treq:    not specified
portid:  5
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_4
traddr:  192.168.2.24
eflags:  none
sectype: none
=====Discovery Log Entry 11======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_4
traddr:  192.168.1.24
eflags:  none
sectype: none
=====Discovery Log Entry 12======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_3
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 13======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_3
traddr:  192.168.1.25
```

```
eflags:  none
sectype: none
=====Discovery Log Entry 14======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  5
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_3
traddr:  192.168.2.24
eflags:  none
sectype: none
=====Discovery Log Entry 15======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_3
traddr:  192.168.1.24
eflags:  none
sectype: none
=====Discovery Log Entry 16======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_2
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 17======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
```

```
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_2
traddr:  192.168.1.25
eflags:  none
sectype: none
=====Discovery Log Entry 18======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  5
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_2
traddr:  192.168.2.24
eflags:  none
sectype: none
=====Discovery Log Entry 19======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme_tcp
_2
traddr:  192.168.1.24
eflags:  none
sectype: none
```

2. Verify that the other NVMe/TCP initiator-target LIF combinations are able to successfully fetch discovery log page data:

```
nvme discover -t tcp -w host-traddr -a traddr
```

Show example

```
nvme discover -t tcp -w 192.168.1.31 -a 192.168.1.24
nvme discover -t tcp -w 192.168.2.31 -a 192.168.2.24
nvme discover -t tcp -w 192.168.1.31 -a 192.168.1.25
nvme discover -t tcp -w 192.168.2.31 -a 192.168.2.25
```

3. Run the `nvme connect-all` command across all the supported NVMe/TCP initiator-target LIFs across the nodes:

```
nvme connect-all -t tcp -w host-traddr -a traddr
```

Show example

```
nvme    connect-all -t  tcp -w  192.168.1.31    -a  192.168.1.24
nvme    connect-all -t  tcp -w  192.168.2.31    -a  192.168.2.24
nvme    connect-all -t  tcp -w  192.168.1.31    -a  192.168.1.25
nvme    connect-all -t  tcp -w  192.168.2.31    -a  192.168.2.25
```

Step 6: Validate NVMe-oF

Verify that the in-kernel NVMe multipath status, ANA status, and ONTAP namespaces are correct for the NVMe-oF configuration.

Steps

1. Verify that the in-kernel NVMe multipath is enabled:

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

You should see the following output:

```
Y
```

2. Verify that the appropriate NVMe-oF settings (such as, model set to NetApp ONTAP Controller and load balancing iopolicy set to round-robin) for the respective ONTAP namespaces correctly reflect on the host:
   a. Display the subsystems:

```
cat /sys/class/nvme-subsystem/nvme-subsys*/model
```

You should see the following output:

```
NetApp ONTAP Controller
NetApp ONTAP Controller
```

b. Display the policy:

```
cat /sys/class/nvme-subsystem/nvme-subsys*/iopolicy
```

You should see the following output:

```
round-robin
round-robin
```

3. Verify that the namespaces are created and correctly discovered on the host:

```
nvme list
```

Show example

```
Node         SN                   Model
---------------------------------------------------------
/dev/nvme4n1 81Ix2BVuekWcAAAAAAAB   NetApp ONTAP Controller


Namespace Usage    Format             FW             Rev
-----------------------------------------------------------
1                 21.47 GB / 21.47 GB   4 KiB + 0 B   FFFFFFFF
```

4. Verify that the controller state of each path is live and has the correct ANA status:

NVMe/FC

```
nvme list-subsys /dev/nvme4n5
```

Show example

```
nvme-subsys4 - NQN=nqn.1992-
08.com.netapp:sn.3a5d31f5502c11ef9f50d039eab6cb6d:subsystem.nvme
_1
               hostnqn=nqn.2014-08.org.nvmexpress:uuid:e6dade64-
216d-
11ec-b7bb-7ed30a5482c3
iopolicy=round-robin\
+- nvme1 fc traddr=nn-0x2082d039eaa7dfc8:pn-
0x2088d039eaa7dfc8,host_traddr=nn-0x20000024ff752e6d:pn-
0x21000024ff752e6d live optimized
+- nvme12 fc traddr=nn-0x2082d039eaa7dfc8:pn-
0x208ad039eaa7dfc8,host_traddr=nn-0x20000024ff752e6d:pn-
0x21000024ff752e6d live non-optimized
+- nvme10 fc traddr=nn-0x2082d039eaa7dfc8:pn-
0x2087d039eaa7dfc8,host_traddr=nn-0x20000024ff752e6c:pn-
0x21000024ff752e6c live non-optimized
+- nvme3 fc traddr=nn-0x2082d039eaa7dfc8:pn-
0x2083d039eaa7dfc8,host_traddr=nn-0x20000024ff752e6c:pn-
0x21000024ff752e6c live optimized
```

NVMe/TCP

```
nvme list-subsys /dev/nvme1n1
```

Show example

```
nvme-subsys5 - NQN=nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme
_tcp_3
hostnqn=nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0035-5910-804b-
b5c04f444d33
iopolicy=round-robin
\
+- nvme13 tcp
traddr=192.168.2.25,trsvcid=4420,host_traddr=192.168.2.31,
src_addr=192.168.2.31 live optimized
+- nvme14 tcp
traddr=192.168.2.24,trsvcid=4420,host_traddr=192.168.2.31,
src_addr=192.168.2.31 live non-optimized
+- nvme5 tcp
traddr=192.168.1.25,trsvcid=4420,host_traddr=192.168.1.31,
src_addr=192.168.1.31 live optimized
+- nvme6 tcp
traddr=192.168.1.24,trsvcid=4420,host_traddr=192.168.1.31,
src_addr=192.168.1.31 live non-optimized
```

5. Verify that the NetApp plug-in displays the correct values for each ONTAP namespace device:

Column

```
nvme netapp ontapdevices -o column
```

Show example

```
Device         Vserver   Namespace Path
----------------------- ------------------------------
/dev/nvme1n1     linux_tcnvme_iscsi
/vol/tcpnvme_1_0_0/tcpnvme_ns

NSID       UUID                                   Size
------------------------------------------------------------
1    5f7f630d-8ea5-407f-a490-484b95b15dd6   21.47GB
```

JSON

```
nvme netapp ontapdevices -o json
```

Show example

```
{
  "ONTAPdevices":[
    {
      "Device":"/dev/nvme1n1",
      "Vserver":"linux_tcnvme_iscsi",
      "Namespace_Path":"/vol/tcpnvme_1_0_0/tcpnvme_ns",
      "NSID":1,
      "UUID":"5f7f630d-8ea5-407f-a490-484b95b15dd6",
      "Size":"21.47GB",
      "LBA_Data_Size":4096,
      "Namespace_Size":5242880
    },
]
}
```

Step 7: Review the known issues

No known issues exist for the NVMe-oF host configuration on Rocky Linux 8.4 with ONTAP release.

SUSE Linux Enterprise Server

SUSE Linux Enterprise Server 15

NVMe-oF host configuration for SUSE Linux Enterprise Server 15 SP7 with ONTAP

NetApp SAN host configurations support the NVMe over Fabrics (NVMe-oF) protocol with Asymmetric Namespace Access (ANA). In NVMe-oF environments, ANA is equivalent to asymmetric logical unit access (ALUA) multipathing in iSCSI and FCP environments. ANA is implemented using the in-kernel NVMe multipath feature.

About this task

You can use the following support and features with the NVMe-oF host configuration for SUSE Linux Enterprise Server 15 SP7 (SLES15 SP7). You should also review the known limitations before starting the configuration process.

- Support available:
  - Support for NVMe over TCP (NVMe/TCP) in addition to NVMe over Fibre Channel (NVMe/FC). The NetApp plug-in in the native `nvme-cli` package displays ONTAP details for both NVMe/FC and NVMe/TCP namespaces.
  - Running both NVMe and SCSI traffic on the same host. For example, you can configure dm-multipath for SCSI mpath devices for SCSI LUNs and use NVMe multipath to configure NVMe-oF namespace devices on the host.
  - Support for SAN booting using the NVMe/FC protocol.
  - Beginning with ONTAP 9.12.1, support for secure in-band authentication is introduced for NVMe/TCP and NVMe/FC. You can use secure in-band authentication for NVMe/TCP and NVMe/FC with SLES15 SP7.
  - Support for persistent discovery controllers (PDCs) using a unique discovery NQN.
  - TLS 1.3 encryption support for NVMe/TCP.
  - NetApp `sanlun` host utility support isn't available for NVMe-oF on a SLES15 SP7 host. Instead, you can rely on the NetApp plug-in included in the native `nvme-cli` package for all NVMe-oF transports.

    For additional details on supported configurations, see the Interoperability Matrix Tool.
- Features available:
  - There are no new features available.
- Known limitations
  - Avoid issuing the `nvme disconnect-all` command on systems booting from SAN over NVMe-TCP or NVMe-FC namespaces because it disconnects both root and data filesystems and might lead to system instability.

Step 1: Optionally, enable SAN booting

You can configure your host to use SAN booting to simplify deployment and improve scalability.

Before you begin

Use the Interoperability Matrix Tool to verify that your Linux OS, host bus adapter (HBA), HBA firmware, HBA boot BIOS, and ONTAP version support SAN booting.

Steps

1. Create a SAN boot namespace and map it to the host.

   See Provision NVMe storage.

2. Enable SAN booting in the server BIOS for the ports to which the SAN boot namespace is mapped.

   For information on how to enable the HBA BIOS, see your vendor-specific documentation.

3. Verify that the configuration was successful by rebooting the host and verifying that the OS is up and running.

Step 2: Validate software versions

Use the following procedure to validate the minimum supported SLES15 SP7 software versions.

Steps

1. Install SLES15 SP7 on the server. After the installation is complete, verify that you are running the specified SLES15 SP7 kernel:

   ```
   uname -r
   ```

   The following example shows a SLES kernel version:

   ```
   6.4.0-150700.53.3-default
   ```

2. Install the `nvme-cli` package:

   ```
   rpm -qa|grep nvme-cli
   ```

   The following example shows an `nvme-cli` package version:

   ```
   nvme-cli-2.11+22.gd31b1a01-150700.3.3.2.x86_64
   ```

3. Install the `libnvme` package:

   ```
   rpm -qa|grep libnvme
   ```

   The following example shows an `libnvme` package version:

   ```
   libnvme1-1.11+4.ge68a91ae-150700.4.3.2.x86_64
   ```

4. On the host, check the hostnqn string at `/etc/nvme/hostnqn`:

```
cat /etc/nvme/hostnqn
```

The following example shows a `hostnqn` version:

```
nqn.2014-08.org.nvmexpress:uuid:f6517cae-3133-11e8-bbff-7ed30aef123f
```

5. Verify that the `hostnqn` string matches the `hostnqn` string for the corresponding subsystem on the ONTAP array:

```
::> vserver nvme subsystem host show -vserver vs_coexistence_LPE36002
```

Show example

```
Vserver Subsystem Priority  Host NQN
------- --------- --------
------------------------------------------------
vs_coexistence_LPE36002
        nvme
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
        nvme_1
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
        nvme_2
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
        nvme_3
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
4 entries were displayed.
```

> If the `hostnqn` strings do not match, use the `vserver modify` command to update the `hostnqn` string on your corresponding ONTAP array subsystem to match the `hostnqn` string from `/etc/nvme/hostnqn` on the host.

Step 3: Configure NVMe/FC

You can configure NVMe/FC with Broadcom/Emulex FC or Marvell/Qlogic FC adapters. You also need to manually discover the NVMe/TCP subsystems and namespaces.

Broadcom/Emulex

Configure NVMe/FC for a Broadcom/Emulex FC adapter.

Steps

1. Verify that you are using the supported adapter model:

   a. Display the model names:

   ```
   cat /sys/class/scsi_host/host*/modelname
   ```

   You should see the following output:

   ```
   LPe36002-M64
   LPe36002-M64
   ```

   b. Display the model descriptions:

   ```
   cat /sys/class/scsi_host/host*/modeldesc
   ```

   You should see an output similar to the following example:

   ```
   Emulex LightPulse LPe36002-M64 2-Port 64Gb Fibre Channel Adapter
   Emulex LightPulse LPe36002-M64 2-Port 64Gb Fibre Channel Adapter
   ```

2. Verify that you are using the recommended Broadcom `lpfc` firmware and inbox driver:

   a. Display the firmware version:

   ```
   cat /sys/class/scsi_host/host*/fwrev
   ```

   The following example shows firmware versions:

   ```
   14.4.393.25, sli-4:2:c
   14.4.393.25, sli-4:2:c
   ```

   b. Display the inbox driver version:

   ```
   cat /sys/module/lpfc/version
   ```

   The following example shows a driver version:

```
0:14.4.0.8
```

For the current list of supported adapter driver and firmware versions, see the Interoperability Matrix Tool.

3. Verify that the expected output of `lpfc_enable_fc4_type` is set to `3`:

```
cat /sys/module/lpfc/parameters/lpfc_enable_fc4_type
```

4. Verify that you can view your initiator ports:

```
cat /sys/class/fc_host/host*/port_name
```

The following example shows port identities:

```
0x10000090fae0ec88
0x10000090fae0ec89
```

5. Verify that your initiator ports are online:

```
cat /sys/class/fc_host/host*/port_state
```

You should see the following output:

```
Online
Online
```

6. Verify that the NVMe/FC initiator ports are enabled and that the target ports are visible:

```
cat /sys/class/scsi_host/host*/nvme_info
```

Show example output

```
NVME Initiator Enabled
XRI Dist lpfc0 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc0 WWPN x10000090fae0ec88 WWNN x20000090fae0ec88
DID x0a1300 ONLINE
NVME RPORT       WWPN x23b1d039ea359e4a WWNN x23aed039ea359e4a
DID x0a1c01 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x22bbd039ea359e4a WWNN x22b8d039ea359e4a
DID x0a1c0b TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2362d039ea359e4a WWNN x234ed039ea359e4a
DID x0a1c10 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x23afd039ea359e4a WWNN x23aed039ea359e4a
DID x0a1a02 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x22b9d039ea359e4a WWNN x22b8d039ea359e4a
DID x0a1a0b TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2360d039ea359e4a WWNN x234ed039ea359e4a
DID x0a1a11 TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 0000004ea0 Cmpl 0000004ea0 Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 0000000000102c35 Issue 0000000000102c2d OutIO
fffffffffffffff8
        abort 00000175 noxri 00000000 nondlp 0000021d qdepth
00000000 wqerr 00000007 err 00000000
FCP CMPL: xb 00000175 Err 0000058b

NVME Initiator Enabled
XRI Dist lpfc1 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc1 WWPN x10000090fae0ec89 WWNN x20000090fae0ec89
DID x0a1200 ONLINE
NVME RPORT       WWPN x23b2d039ea359e4a WWNN x23aed039ea359e4a
DID x0a1d01 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x22bcd039ea359e4a WWNN x22b8d039ea359e4a
DID x0a1d0b TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2363d039ea359e4a WWNN x234ed039ea359e4a
DID x0a1d10 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x23b0d039ea359e4a WWNN x23aed039ea359e4a
DID x0a1b02 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x22bad039ea359e4a WWNN x22b8d039ea359e4a
DID x0a1b0b TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2361d039ea359e4a WWNN x234ed039ea359e4a
DID x0a1b11 TARGET DISCSRVC ONLINE

NVME Statistics
```

```
LS: Xmt 0000004e31 Cmpl 0000004e31 Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 00000000001017f2 Issue 00000000001017ef OutIO
fffffffffffffffd
        abort 0000018a noxri 00000000 nondlp 0000012e qdepth
00000000 wqerr 00000004 err 00000000
FCP CMPL: xb 0000018a Err 000005ca
```

Marvell/QLogic

Configure NVMe/FC for a Marvell/QLogic adapter.

Steps

1. Verify that you are running the supported adapter driver and firmware versions:

   ```
   cat /sys/class/fc_host/host*/symbolic_name
   ```

   The follow example shows driver and firmware versions:

   ```
   QLE2742 FW:v9.14.00 DVR:v10.02.09.400-k-debug
   QLE2742 FW:v9.14.00 DVR:v10.02.09.400-k-debug
   ```

2. Verify that `ql2xnvmeenable` is set. This enables the Marvell adapter to function as an NVMe/FC initiator:

   ```
   cat /sys/module/qla2xxx/parameters/ql2xnvmeenable
   ```

   The expected output is 1.

Step 4: Optionally, enable 1MB I/O

ONTAP reports an MDTS (Max Data Transfer Size) of 8 in the Identify Controller data. This means the maximum I/O request size can be up to 1MB. To issue I/O requests of size 1MB for a Broadcom NVMe/FC host, you should increase the `lpfc` value of the `lpfc_sg_seg_cnt` parameter to 256 from the default value of 64.

> These steps don't apply to Qlogic NVMe/FC hosts.

Steps

1. Set the `lpfc_sg_seg_cnt` parameter to 256:

```
cat /etc/modprobe.d/lpfc.conf
```

You should see an output similar to the following example:

```
options lpfc lpfc_sg_seg_cnt=256
```

2. Run the `dracut -f` command, and reboot the host.
3. Verify that the value for `lpfc_sg_seg_cnt` is 256:

```
cat /sys/module/lpfc/parameters/lpfc_sg_seg_cnt
```

Step 5: Verify NVMe boot services

With SLES 15 SP7, the `nvmefc-boot-connections.service` and `nvmf-autoconnect.service` boot services included in the NVMe/FC `nvme-cli` package are automatically enabled to start during the system boot. After the system boot completes, verify that the boot services are enabled.

Steps

1. Verify that `nvmf-autoconnect.service` is enabled:

```
systemctl status nvmf-autoconnect.service
```

Show example output

```
nvmf-autoconnect.service - Connect NVMe-oF subsystems automatically
during boot
  Loaded: loaded (/usr/lib/systemd/system/nvmf-autoconnect.service;
enabled; preset: enabled)
  Active: inactive (dead) since Fri 2025-07-04 23:56:38 IST; 4 days
ago
  Main PID: 12208 (code=exited, status=0/SUCCESS)
    CPU: 62ms

Jul 04 23:56:26 localhost systemd[1]: Starting Connect NVMe-oF
subsystems automatically during boot...
Jul 04 23:56:38 localhost systemd[1]: nvmf-autoconnect.service:
Deactivated successfully.
Jul 04 23:56:38 localhost systemd[1]: Finished Connect NVMe-oF
subsystems automatically during boot.
```

2. Verify that `nvmefc-boot-connections.service` is enabled:

```
systemctl status nvmefc-boot-connections.service
```

Show example output

```
nvmefc-boot-connections.service - Auto-connect to subsystems on FC-NVME devices found during boot
    Loaded: loaded (/usr/lib/systemd/system/nvmefc-boot-connections.service; enabled; preset: enabled)
    Active: inactive (dead) since Mon 2025-07-07 19:52:30 IST; 1 day 4h ago
  Main PID: 2945 (code=exited, status=0/SUCCESS)
       CPU: 14ms

Jul 07 19:52:30 HP-DL360-14-168 systemd[1]: Starting Auto-connect to subsystems on FC-NVME devices found during boot...
Jul 07 19:52:30 HP-DL360-14-168 systemd[1]: nvmefc-boot-connections.service: Deactivated successfully.
Jul 07 19:52:30 HP-DL360-14-168 systemd[1]: Finished Auto-connect to subsystems on FC-NVME devices found during boot.
```

Step 6: Configure NVMe/TCP

The NVMe/TCP protocol doesn't support the `auto-connect` operation. Instead, you can discover the NVMe/TCP subsystems and namespaces by performing the NVMe/TCP `connect` or `connect-all` operations manually.

Steps

1. Verify that the initiator port can fetch the discovery log page data across the supported NVMe/TCP LIFs:

```
nvme discover -t tcp -w <host-traddr> -a <traddr>
```

Show example output

```
nvme discover -t tcp -w 192.168.111.80 -a 192.168.111.70
Discovery Log Number of Records 8, Generation counter 42
=====Discovery Log Entry 0======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  4
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.f8e2af201b7211f0ac2bd039eab67a95:discovery
traddr:  192.168.211.71
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 1======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  3
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.f8e2af201b7211f0ac2bd039eab67a95:discovery
traddr:  192.168.111.71
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 2======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  2
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.f8e2af201b7211f0ac2bd039eab67a95:discovery
traddr:  192.168.211.70
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 3======
trtype:  tcp
adrfam:  ipv4
```

```
subtype: current discovery subsystem
treq:    not specified
portid:  1
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.f8e2af201b7211f0ac2bd039eab67a95:discovery
traddr:  192.168.111.70
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 4======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.f8e2af201b7211f0ac2bd039eab67a95:subsystem.sample_t
cp_sub
traddr:  192.168.211.71
eflags:  none
sectype: none
=====Discovery Log Entry 5======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  3
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.f8e2af201b7211f0ac2bd039eab67a95:subsystem.sample_t
cp_sub
traddr:  192.168.111.71
eflags:  none
sectype: none
=====Discovery Log Entry 6======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.f8e2af201b7211f0ac2bd039eab67a95:subsystem.sample_t
cp_sub
```

```
traddr:  192.168.211.70
eflags:  none
sectype: none
=====Discovery Log Entry 7======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.f8e2af201b7211f0ac2bd039eab67a95:subsystem.sample_t
cp_sub
traddr:  192.168.111.70
eflags:  none
sectype: none
localhost:~ #
```

2. Verify that all other NVMe/TCP initiator-target LIF combinations can successfully fetch discovery log page data:

```
nvme discover -t tcp -w <host-traddr> -a <traddr>
```

Show example

```
nvme discover -t tcp -w 192.168.111.80 -a 192.168.111.66
nvme discover -t tcp -w 192.168.111.80 -a 192.168.111.67
nvme discover -t tcp -w 192.168.211.80 -a 192.168.211.66
nvme discover -t tcp -w 192.168.211.80 -a 192.168.211.67
```

3. Run the `nvme connect-all` command across all the supported NVMe/TCP initiator-target LIFs across the nodes:

```
nvme connect-all -t tcp -w <host-traddr> -a <traddr>
```

Show example

```
nvme    connect-all -t  tcp -w  192.168.111.80  -a  192.168.111.66
nvme    connect-all -t  tcp -w  192.168.111.80  -a  192.168.111.67
nvme    connect-all -t  tcp -w  192.168.211.80  -a  192.168.211.66
nvme    connect-all -t  tcp -w  192.168.211.80  -a  192.168.211.67
```

> Beginning with SLES 15 SP6, the default setting for the NVMe/TCP `ctrl-loss-tmo` timeout is turned off. This means that there is no limit on the number of retries (indefinite retry), and you don't need to manually configure a specific `ctrl-loss-tmo` timeout duration when using the `nvme connect` or `nvme connect-all` commands (option `-l`). Additonally, the NVMe/TCP controllers don't experience timeouts in the event of a path failure and remain connected indefinitely.

Step 7: Validate NVMe-oF

Verify that the in-kernel NVMe multipath status, ANA status, and ONTAP namespaces are correct for the NVMe-oF configuration.

Steps

1. Verify that the in-kernel NVMe multipath is enabled:

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

You should see the following output:

```
Y
```

2. Verify that the appropriate NVMe-oF settings (such as, model set to NetApp ONTAP Controller and load balancing iopolicy set to round-robin) for the respective ONTAP namespaces correctly reflect on the host:
   a. Display the subsystems:

```
cat /sys/class/nvme-subsystem/nvme-subsys*/model
```

You should see the following output:

```
NetApp ONTAP Controller
NetApp ONTAP Controller
```

   b. Display the policy:

```
cat /sys/class/nvme-subsystem/nvme-subsys*/iopolicy
```

You should see the following output:

```
round-robin
round-robin
```

3. Verify that the namespaces are created and correctly discovered on the host:

```
nvme list
```

Show example

```
Node          SN                   Model
---------------------------------------------------------
/dev/nvme4n1 81Ix2BVuekWcAAAAAAAB   NetApp ONTAP Controller


Namespace Usage    Format             FW            Rev
-----------------------------------------------------------
1                 21.47 GB / 21.47 GB   4 KiB + 0 B   FFFFFFFF
```

4. Verify that the controller state of each path is live and has the correct ANA status:

NVMe/FC

```
nvme list-subsys /dev/nvme4n5
```

Show example output

```
nvme-subsys114 - NQN=nqn.1992-
08.com.netapp:sn.9e30b9760a4911f08c87d039eab67a95:subsystem.sles
_161_27
                 hostnqn=nqn.2014-
08.org.nvmexpress:uuid:f6517cae-3133-11e8-bbff-7ed30aef123f
iopolicy=round-robin\
+- nvme114 fc traddr=nn-0x234ed039ea359e4a:pn-
0x2360d039ea359e4a,host_traddr=nn-0x20000090fae0ec88:pn-
0x10000090fae0ec88 live optimized
+- nvme115 fc traddr=nn-0x234ed039ea359e4a:pn-
0x2362d039ea359e4a,host_traddr=nn-0x20000090fae0ec88:pn-
0x10000090fae0ec88 live non-optimized
+- nvme116 fc traddr=nn-0x234ed039ea359e4a:pn-
0x2361d039ea359e4a,host_traddr=nn-0x20000090fae0ec89:pn-
0x10000090fae0ec89 live optimized
+- nvme117 fc traddr=nn-0x234ed039ea359e4a:pn-
0x2363d039ea359e4a,host_traddr=nn-0x20000090fae0ec89:pn-
0x10000090fae0ec89 live non-optimized
```

NVMe/TCP

```
nvme list-subsys /dev/nvme9n1
```

Show example output

```
nvme-subsys9 - NQN=nqn.1992-
08.com.netapp:sn.f8e2af201b7211f0ac2bd039eab67a95:subsystem.with
_inband_with_json hostnqn=nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0035-5910-804b-b2c04f444d33
iopolicy=round-robin
\
+- nvme10 tcp
traddr=192.168.111.71,trsvcid=4420,src_addr=192.168.111.80 live
non-optimized
 +- nvme11 tcp
traddr=192.168.211.70,trsvcid=4420,src_addr=192.168.211.80 live
optimized
 +- nvme12 tcp
traddr=192.168.111.70,trsvcid=4420,src_addr=192.168.111.80 live
optimized
 +- nvme9 tcp
traddr=192.168.211.71,trsvcid=4420,src_addr=192.168.211.80 live
non-optimized
```

5. Verify that the NetApp plug-in displays the correct values for each ONTAP namespace device:

Column

```
nvme netapp ontapdevices -o column
```

Show example

```
Device           Vserver                   Namespace Path
NSID UUID                                   Size
---------------- -------------------------
-------------------------------------------------- ----
------------------------------------- ---------
/dev/nvme0n1     vs_161
/vol/fc_nvme_vol1/fc_nvme_ns1                     1    32fd92c7-
0797-428e-a577-fdb3f14d0dc3   5.37GB
```

JSON

```
nvme netapp ontapdevices -o json
```

Show example


```
{
      "Device":"/dev/nvme98n2",
      "Vserver":"vs_161",
      "Namespace_Path":"/vol/fc_nvme_vol71/fc_nvme_ns71",
      "NSID":2,
      "UUID":"39d634c4-a75e-4fbd-ab00-3f9355a26e43",
      "LBA_Size":4096,
      "Namespace_Size":5368709120,
      "UsedBytes":430649344,
    }
  ]
}
```


Step 8: Create a persistent discovery controller

Beginning with ONTAP 9.11.1, you can create a persistent discovery controller (PDC) for a SLES 15 SP7 host. A PDC is required to automatically detect an NVMe subsystem add or remove operation and changes to the discovery log page data.

Steps

1. Verify that the discovery log page data is available and can be retrieved through the initiator port and target LIF combination:

```
nvme discover -t <trtype> -w <host-traddr> -a <traddr>
```

Show example output

```
Discovery Log Number of Records 8, Generation counter 18
=====Discovery Log Entry 0======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  4
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.4f7af2bd221811f0afadd039eab0dadd:discovery
traddr:  192.168.111.66
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 1======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  2
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.4f7af2bd221811f0afadd039eab0dadd:discovery
traddr:  192.168.211.66
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 2======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  3
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.4f7af2bd221811f0afadd039eab0dadd:discovery
traddr:  192.168.111.67
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 3======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
```

```
treq:    not specified
portid:  1
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.4f7af2bd221811f0afadd039eab0dadd:discovery
traddr:  192.168.211.67
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 4======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.4f7af2bd221811f0afadd039eab0dadd:subsystem.pdc
traddr:  192.168.111.66
eflags:  none
sectype: none
=====Discovery Log Entry 5======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.4f7af2bd221811f0afadd039eab0dadd:subsystem.pdc
traddr:  192.168.211.66
eflags:  none
sectype: none
=====Discovery Log Entry 6======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  3
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.4f7af2bd221811f0afadd039eab0dadd:subsystem.pdc
traddr:  192.168.111.67
eflags:  none
sectype: none
=====Discovery Log Entry 7======
```

```
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.4f7af2bd221811f0afadd039eab0dadd:subsystem.pdc
traddr:  192.168.211.67
eflags:  none
sectype: none
```

2. Create a PDC for the discovery subsystem:

```
nvme discover -t <trtype> -w <host-traddr> -a <traddr> -p
```

You should see the following output:

```
nvme discover -t tcp -w 192.168.111.80 -a 192.168.111.66 -p
```

3. From the ONTAP controller, verify that the PDC has been created:

```
vserver nvme show-discovery-controller -instance -vserver <vserver_name>
```

Show example output

```
vserver nvme show-discovery-controller -instance -vserver vs_pdc

           Vserver Name: vs_pdc
               Controller ID: 0101h
     Discovery Subsystem NQN: nqn.1992-
08.com.netapp:sn.4f7af2bd221811f0afadd039eab0dadd:discovery
           Logical Interface: lif2
                        Node: A400-12-181
                    Host NQN: nqn.2014-
08.org.nvmexpress:uuid:9796c1ec-0d34-11eb-b6b2-3a68dd3bab57
          Transport Protocol: nvme-tcp
 Initiator Transport Address: 192.168.111.80
Transport Service Identifier: 8009
             Host Identifier: 9796c1ec0d3411ebb6b23a68dd3bab57
           Admin Queue Depth: 32
       Header Digest Enabled: false
         Data Digest Enabled: false
   Keep-Alive Timeout (msec): 30000
```

Step 9: Set up secure in-band authentication

Beginning with ONTAP 9.12.1, secure in-band authentication is supported over NVMe/TCP and NVMe/FC between the host and the ONTAP controller.

To set up secure authentication, each host or controller must be associated with a `DH-HMAC-CHAP` key, which is a combination of the NQN of the NVMe host or controller and an authentication secret configured by the administrator. To authenticate its peer, an NVMe host or controller must recognize the key associated with the peer.

You can set up secure in-band authentication using the CLI or a config JSON file. If you need to specify different dhchap keys for different subsystems, you must use a config JSON file.

CLI

Set up secure in-band authentication using the CLI.

Steps

1. Obtain the host NQN:

```
cat /etc/nvme/hostnqn
```

2. Generate the dhchap key for the host.

   The following output describes the `gen-dhchap-key` command paramters:

```
nvme gen-dhchap-key -s optional_secret -l key_length {32|48|64} -m
HMAC_function {0|1|2|3} -n host_nqn
•   -s secret key in hexadecimal characters to be used to initialize
the host key
•   -l length of the resulting key in bytes
•   -m HMAC function to use for key transformation
0 = none, 1- SHA-256, 2 = SHA-384, 3=SHA-512
•   -n host NQN to use for key transformation
```

   In the following example, a random dhchap key with HMAC set to 3 (SHA-512) is generated.

```
nvme gen-dhchap-key -m 3 -n nqn.2014-
08.org.nvmexpress:uuid:e6dade64-216d-11ec-b7bb-7ed30a5482c3
DHHC-
1:03:1CFivw9ccz58gAcOUJrM7Vs98hd2ZHSr+iw+Amg6xZPl5D2Yk+HDTZiUAg1iGgx
TYqnxukqvYedA55Bw3wtz6sJNpR4=:
```

3. On the ONTAP controller, add the host and specify both dhchap keys:

```
vserver nvme subsystem host add -vserver <svm_name> -subsystem
<subsystem> -host-nqn <host_nqn> -dhchap-host-secret
<authentication_host_secret> -dhchap-controller-secret
<authentication_controller_secret> -dhchap-hash-function {sha-
256|sha-512} -dhchap-group {none|2048-bit|3072-bit|4096-bit|6144-
bit|8192-bit}
```

4. A host supports two types of authentication methods, unidirectional and bidirectional. On the host, connect to the ONTAP controller and specify dhchap keys based on the chosen authentication method:

```
nvme connect -t tcp -w <host-traddr> -a <tr-addr> -n <host_nqn> -S
<authentication_host_secret> -C <authentication_controller_secret>
```

5. Validate the `nvme connect authentication` command by verifying the host and controller dhchap keys:

   a. Verify the host dhchap keys:

   ```
   cat /sys/class/nvme-subsystem/<nvme-subsysX>/nvme*/dhchap_secret
   ```

   Show example output for a unidirectional configuration

   ```
   # cat /sys/class/nvme-subsystem/nvme-
   subsys1/nvme*/dhchap_secret
   DHHC-1:01:iM63E6cX7G5SOKKOju8gmzM53qywsy+C/YwtzxhIt9ZRz+ky:
   DHHC-1:01:iM63E6cX7G5SOKKOju8gmzM53qywsy+C/YwtzxhIt9ZRz+ky:
   DHHC-1:01:iM63E6cX7G5SOKKOju8gmzM53qywsy+C/YwtzxhIt9ZRz+ky:
   DHHC-1:01:iM63E6cX7G5SOKKOju8gmzM53qywsy+C/YwtzxhIt9ZRz+ky:
   ```

   b. Verify the controller dhchap keys:

   ```
   cat /sys/class/nvme-subsystem/<nvme-
   subsysX>/nvme*/dhchap_ctrl_secret
   ```

   Show example output for a bidirectional configuration

   ```
   # cat /sys/class/nvme-subsystem/nvme-
   subsys6/nvme*/dhchap_ctrl_secret
   DHHC-
   1:03:1CFivw9ccz58gAcOUJrM7Vs98hd2ZHSr+iw+Amg6xZPl5D2Yk+HDTZiUA
   g1iGgxTYqnxukqvYedA55Bw3wtz6sJNpR4=:
   DHHC-
   1:03:1CFivw9ccz58gAcOUJrM7Vs98hd2ZHSr+iw+Amg6xZPl5D2Yk+HDTZiUA
   g1iGgxTYqnxukqvYedA55Bw3wtz6sJNpR4=:
   DHHC-
   1:03:1CFivw9ccz58gAcOUJrM7Vs98hd2ZHSr+iw+Amg6xZPl5D2Yk+HDTZiUA
   g1iGgxTYqnxukqvYedA55Bw3wtz6sJNpR4=:
   DHHC-
   1:03:1CFivw9ccz58gAcOUJrM7Vs98hd2ZHSr+iw+Amg6xZPl5D2Yk+HDTZiUA
   g1iGgxTYqnxukqvYedA55Bw3wtz6sJNpR4=:
   ```

JSON file

When multiple NVMe subsystems are available on the ONTAP controller configuration, you can use the `/etc/nvme/config.json` file with the `nvme connect-all` command.

To generate the JSON file, you can use the `-o` option. See the NVMe connect-all manual pages for more syntax options.

Steps

1. Configure the JSON file:

Show example output

```
# cat /etc/nvme/config.json
[
 {
    "hostnqn":"nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0035-
5910-804b-b2c04f444d33",
    "hostid":"4c4c4544-0035-5910-804b-b2c04f444d33",
    "dhchap_key":"DHHC-
1:01:i4i789R11sMuHLCY27RVI8XloC\/GzjRwyhxip5hmIELsHrBq:",
    "subsystems":[
      {
        "nqn":"nqn.1992-
08.com.netapp:sn.f8e2af201b7211f0ac2bd039eab67a95:subsystem.samp
le_tcp_sub",
        "ports":[
          {
            "transport":"tcp",
            "traddr":"192.168.111.70",
            "host_traddr":"192.168.111.80",
            "trsvcid":"4420"
            "dhchap_ctrl_key":"DHHC-
1:03:jqgYcJSKp73+XqAf2X6twr9ngBpr2n0MGWbmZIZq4PieKZCoilKGef8lAvh
YS0PNK7T+04YD5CRPjh+m3qjJU++yR8s=:"
          },
               {
                    "transport":"tcp",
                    "traddr":"192.168.111.71",
                    "host_traddr":"192.168.111.80",
                    "trsvcid":"4420",
                    "dhchap_ctrl_key":"DHHC-
1:03:jqgYcJSKp73+XqAf2X6twr9ngBpr2n0MGWbmZIZq4PieKZCoilKGef8lAvh
YS0PNK7T+04YD5CRPjh+m3qjJU++yR8s=:"
               },
               {
                    "transport":"tcp",
                    "traddr":"192.168.211.70",
                    "host_traddr":"192.168.211.80",
                    "trsvcid":"4420",
                    "dhchap_ctrl_key":"DHHC-
1:03:jqgYcJSKp73+XqAf2X6twr9ngBpr2n0MGWbmZIZq4PieKZCoilKGef8lAvh
YS0PNK7T+04YD5CRPjh+m3qjJU++yR8s=:"
               },
               {
                    "transport":"tcp",
                    "traddr":"192.168.211.71",
```

```
                    "host_traddr":"192.168.211.80",
                    "trsvcid":"4420",
                    "dhchap_ctrl_key":"DHHC-
1:03:jqgYcJSKp73+XqAf2X6twr9ngBpr2n0MGWbmZIZq4PieKZCoilKGef8lAvh
YS0PNK7T+04YD5CRPjh+m3qjJU++yR8s=:"
               }
           ]
       }
   ]
 }
]
```

In the preceding example, `dhchap_key` corresponds to `dhchap_secret` and `dhchap_ctrl_key` corresponds to `dhchap_ctrl_secret`.

2. Connect to the ONTAP controller using the config JSON file:

```
nvme connect-all -J /etc/nvme/config.json
```

Show example output

```
traddr=192.168.211.70 is already connected
traddr=192.168.111.71 is already connected
traddr=192.168.211.71 is already connected
traddr=192.168.111.70 is already connected
traddr=192.168.211.70 is already connected
traddr=192.168.111.70 is already connected
traddr=192.168.211.71 is already connected
traddr=192.168.111.71 is already connected
traddr=192.168.211.70 is already connected
traddr=192.168.111.71 is already connected
traddr=192.168.211.71 is already connected
traddr=192.168.111.70 is already connected
```

3. Verify that the dhchap secrets have been enabled for the respective controllers for each subsystem:
   a. Verify the host dhchap keys:

```
cat /sys/class/nvme-subsystem/nvme-subsys0/nvme0/dhchap_secret
```

You should see the following output:

```
DHHC-1:01:i4i789R11sMuHLCY27RVI8XloC/GzjRwyhxip5hmIELsHrBq:
```

b. Verify the controller dhchap keys:

```
cat /sys/class/nvme-subsystem/nvme-
subsys0/nvme0/dhchap_ctrl_secret
```

You should see the following output:

```
DHHC-
1:03:jqgYcJSKp73+XqAf2X6twr9ngBpr2n0MGWbmZIZq4PieKZCoilKGef8lAvhY
S0PNK7T+04YD5CRPjh+m3qjJU++yR8s=:
```

Step 10: Configure Transport Layer Security

Transport Layer Security (TLS) provides secure end-to-end encryption for NVMe connections between NVMe-oF hosts and an ONTAP array. Beginning with ONTAP 9.16.1, you can configure TLS 1.3 using the CLI and a configured pre-shared key (PSK).

About this task

You perform the steps in this procedure on the SUSE Linux Enterprise Server host, except where it specifies that you perform a step on the ONTAP controller.

Steps

1. Check that you have the following `ktls-utils`, `openssl`, and `libopenssl` packages installed on the host:

   a. Verify the `ktls-utils`:

   ```
   rpm -qa | grep ktls
   ```

   You should see the following output displayed:

   ```
   ktls-utils-0.10+33.g311d943-150700.1.5.x86_64
   ```

   b. Verify the SSL packages:

   ```
   rpm -qa | grep ssl
   ```

Show example output

```
libopenssl3-3.2.3-150700.3.20.x86_64
openssl-3-3.2.3-150700.3.20.x86_64
libopenssl1_1-1.1.1w-150700.9.37.x86_64
```

2. Verify that you have the correct setup for `/etc/tlshd.conf`:

```
cat /etc/tlshd.conf
```

Show example output

```
[debug]
loglevel=0
tls=0
nl=0
[authenticate]
keyrings=.nvme
[authenticate.client]
#x509.truststore= <pathname>
#x509.certificate= <pathname>
#x509.private_key= <pathname>
[authenticate.server]
#x509.truststore= <pathname>
#x509.certificate= <pathname>
#x509.private_key= <pathname>
```

3. Enable `tlshd` to start at system boot:

```
systemctl enable tlshd
```

4. Verify that the `tlshd` daemon is running:

```
systemctl status tlshd
```

Show example output

```
tlshd.service - Handshake service for kernel TLS consumers
   Loaded: loaded (/usr/lib/systemd/system/tlshd.service; enabled;
preset: disabled)
   Active: active (running) since Wed 2024-08-21 15:46:53 IST; 4h
57min ago
     Docs: man:tlshd(8)
Main PID: 961 (tlshd)
   Tasks: 1
     CPU: 46ms
   CGroup: /system.slice/tlshd.service
        └─961 /usr/sbin/tlshd
Aug 21 15:46:54 RX2530-M4-17-153 tlshd[961]: Built from ktls-utils
0.11-dev on Mar 21 2024 12:00:00
```

5. Generate the TLS PSK by using the `nvme gen-tls-key`:
   a. Verify the host:

      ```
      cat /etc/nvme/hostnqn
      ```

      You should see the following output:

      ```
      nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0035-5910-804b-b2c04f444d33
      ```

   b. Verify the key:

      ```
      nvme gen-tls-key --hmac=1 --identity=1 --subsysnqn= nqn.1992-
      08.com.netapp:sn.a2d41235b78211efb57dd039eab67a95:subsystem.nvme1
      ```

      You should see the following output:

      ```
      NVMeTLSkey-1:01:C50EsaGtuOp8n5fGE9EuWjbBCtshmfoHx4XTqTJUmydf0gIj:
      ```

6. On the ONTAP controller, add the TLS PSK to the ONTAP subsystem:

Show example output

```
nvme subsystem host add -vserver vs_iscsi_tcp -subsystem nvme1 -host
-nqn nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0035-5910-804b-
b2c04f444d33 -tls-configured-psk NVMeTLSkey-
1:01:C50EsaGtuOp8n5fGE9EuWjbBCtshmfoHx4XTqTJUmydf0gIj:
```

7. Insert the TLS PSK into the host kernel keyring:

```
nvme check-tls-key --identity=1 --subsysnqn=nqn.1992
-08.com.netapp:sn.a2d41235b78211efb57dd039eab67a95:subsystem.nvme1
--keydata=NVMeTLSkey
-1:01:C50EsaGtuOp8n5fGE9EuWjbBCtshmfoHx4XTqTJUmydf0gIj: --insert
```

You should see the following TLS key:

```
Inserted TLS key 22152a7e
```

The PSK shows as `NVMe1R01` because it uses `identity v1` from the TLS handshake algorithm. Identity v1 is the only version that ONTAP supports.

8. Verify that the TLS PSK is inserted correctly:

```
cat /proc/keys | grep NVMe
```

Show example output

```
069f56bb I--Q---     5 perm 3b010000     0      0 psk        NVMe1R01
nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0035-5910-804b-b2c04f444d33
nqn.1992-
08.com.netapp:sn.a2d41235b78211efb57dd039eab67a95:subsystem.nvme1
oYVLelmiOwnvDjXKBmrnIgGVpFIBDJtc4hmQXE/36Sw=: 32
```

9. Connect to the ONTAP subsystem using the inserted TLS PSK:

   a. Verify the TLS PSK:

```
nvme connect -t tcp -w 192.168.111.80 -a 192.168.111.66  -n nqn.1992-
08.com.netapp:sn.a2d41235b78211efb57dd039eab67a95:subsystem.nvme1
--tls_key=0x069f56bb –tls
```

You should see the following output:

```
connecting to device: nvme0
```

b. Verify the list-subsys:

```
nvme list-subsys
```

Show example output

```
nvme-subsys0 - NQN=nqn.1992-
08.com.netapp:sn.a2d41235b78211efb57dd039eab67a95:subsystem.nvme1
               hostnqn=nqn.2014-08.org.nvmexpress:uuid:4c4c4544-
0035-5910-804b-b2c04f444d33
\
 +- nvme0 tcp
traddr=192.168.111.66,trsvcid=4420,host_traddr=192.168.111.80,src
_addr=192.168.111.80 live
```

10. Add the target, and verify the TLS connection to the specified ONTAP subsystem:

```
nvme subsystem controller show -vserver sles15_tls -subsystem sles15
-instance
```

Show example output

```
(vserver nvme subsystem controller show)
                          Vserver Name: vs_iscsi_tcp
                          Subsystem: nvme1
                      Controller ID: 0040h
                  Logical Interface: tcpnvme_lif1_1
                               Node: A400-12-181
                           Host NQN: nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0035-5910-804b-b2c04f444d33
                 Transport Protocol: nvme-tcp
        Initiator Transport Address: 192.168.111.80
                    Host Identifier:
4c4c454400355910804bb2c04f444d33
               Number of I/O Queues: 2
                   I/O Queue Depths: 128, 128
                  Admin Queue Depth: 32
              Max I/O Size in Bytes: 1048576
          Keep-Alive Timeout (msec): 5000
                     Subsystem UUID: 8bbfb403-1602-11f0-ac2b-
d039eab67a95
              Header Digest Enabled: false
                Data Digest Enabled: false
       Authentication Hash Function: sha-256
Authentication Diffie-Hellman Group: 3072-bit
                Authentication Mode: unidirectional
       Transport Service Identifier: 4420
                       TLS Key Type: configured
                   TLS PSK Identity: NVMe1R01 nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0035-5910-804b-b2c04f444d33
nqn.1992-
08.com.netapp:sn.a2d41235b78211efb57dd039eab67a95:subsystem.nvme1
oYVLelmiOwnvDjXKBmrnIgGVpFIBDJtc4hmQXE/36Sw=
                         TLS Cipher: TLS-AES-128-GCM-SHA256
```

Step 11: Review the known issues

There are no known issues.

NVMe-oF host configuration for SUSE Linux Enterprise Server 15 SP6 with ONTAP

NVMe over Fabrics (NVMe-oF), including NVMe over Fibre Channel (NVMe/FC) and other transports, is supported for SUSE Linux Enterprise Server 15 SP6 with Asymmetric Namespace Access (ANA). In NVMe-oF environments, ANA is the equivalent of ALUA multipathing in iSCSI and FCP environments and is implemented with in-kernel NVMe multipath.

The following support is available for the NVMe-oF host configuration for SUSE Linux Enterprise Server 15 SP6 with ONTAP:

- Running NVMe and SCSI traffic on the same host. For example, you can configure dm-multipath for SCSI `mpath` devices for SCSI LUNs and use NVMe multipath to configure NVMe-oF namespace devices on the host.
- Support for NVMe over TCP (NVMe/TCP) and NVMe/FC. This gives the NetApp plug-in in the native `nvme-cli` package the capability to display the ONTAP details for both NVMe/FC and NVMe/TCP namespaces.

For additional details on supported configurations, see the Interoperability Matrix Tool.

Features

- Support for NVMe secure, in-band authentication
- Support for persistent discovery controllers (PDCs) using a unique discovery NQN
- TLS 1.3 encryption support for NVMe/TCP

Known limitations

- SAN booting using the NVMe-oF protocol is currently not supported.
- NetApp `sanlun` host utility support isn't available for NVMe-oF on a SUSE Linux Enterprise Server 15 SP6 host. Instead, you can rely on the NetApp plug-in included in the native `nvme-cli` package for all NVMe-oF transports.

Configure NVMe/FC

You can configure NVMe/FC with Broadcom/Emulex FC or Marvell/Qlogic FC adapters for a SUSE Linux Enterprise Server 15 SP6 with ONTAP configuration.

Broadcom/Emulex

Configure NVMe/FC for a Broadcom/Emulex FC adapter.

Steps

1. Verify that you are using the recommended adapter model:

   ```
   cat /sys/class/scsi_host/host*/modelname
   ```

   Example output

   ```
   LPe32002 M2
   LPe32002-M2
   ```

2. Verify the adapter model description:

   ```
   cat /sys/class/scsi_host/host*/modeldesc
   ```

   Example output

   ```
   Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
   Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
   ```

3. Verify that you are using the recommended Emulex host bus adapter (HBA) firmware versions:

   ```
   cat /sys/class/scsi_host/host*/fwrev
   ```

   Example output

   ```
   14.2.673.40, sli-4:2:c
   14.2.673.40, sli-4:2:c
   ```

4. Verify that you are using the recommended LPFC driver version:

   ```
   cat /sys/module/lpfc/version
   ```

   Example output

   ```
   0:14.4.0.1
   ```

5. Verify that you can view your initiator ports:

```
cat /sys/class/fc_host/host*/port_name
```

Example output

```
0x10000090fae0ec88
0x10000090fae0ec89
```

6. Verify that your initiator ports are online:

```
cat /sys/class/fc_host/host*/port_state
```

Example output

```
Online
Online
```

7. Verify that the NVMe/FC initiator ports are enabled and that the target ports are visible:

```
cat /sys/class/scsi_host/host*/nvme_info
```

In the following example, one initiator port is enabled and connected with two target LIFs.

Show example output

```
NVME Initiator Enabled
XRI Dist lpfc0 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc0 WWPN x10000090fae0ec88 WWNN x20000090fae0ec88
DID x0a1300 ONLINE
NVME RPORT WWPN x2070d039ea359e4a WWNN x206bd039ea359e4a DID
x0a0a05 TARGET DISCSRVC
ONLINE
NVME Statistics
LS: Xmt 00000003ba Cmpl 00000003ba Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 0000000014e3dfb8 Issue 0000000014e308db OutIO
fffffffffffff2923
 abort 00000845 noxri 00000000 nondlp 00000063 qdepth 00000000
wqerr 00000003 err 00000000
FCP CMPL: xb 00000847 Err 00027f33
NVME Initiator Enabled
XRI Dist lpfc1 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc1 WWPN x10000090fae0ec89 WWNN x20000090fae0ec89
DID x0a1200 ONLINE
NVME RPORT WWPN x2071d039ea359e4a WWNN x206bd039ea359e4a DID
x0a0305 TARGET DISCSRVC
ONLINE
NVME Statistics
LS: Xmt 00000003ba Cmpl 00000003ba Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 0000000014e39f78 Issue 0000000014e2b832 OutIO
ffffffffffff18ba
 abort 0000082d noxri 00000000 nondlp 00000028 qdepth 00000000
wqerr 00000007 err 00000000
FCP CMPL: xb 0000082d Err 000283bb
```

Marvell/QLogic

The native inbox qla2xxx driver included in the SUSE Linux Enterprise Server 15 SP6 kernel has the latest fixes. These fixes are essential for ONTAP support.

Configure NVMe/FC for a Marvell/QLogic adapter.

Steps

1. Verify that you are running the supported adapter driver and firmware versions:

```
cat /sys/class/fc_host/host*/symbolic_name
```

Example output

```
QLE2742 FW:v9.14.01 DVR: v10.02.09.200-k
QLE2742 FW:v9.14.01 DVR: v10.02.09.200-k
```

2. Verify that the `ql2xnvmeenable` parameter is set to 1:

```
cat /sys/module/qla2xxx/parameters/ql2xnvmeenable
```

The expected value is 1.

Enable 1MB I/O size (Optional)

ONTAP reports an MDTS (Max Data Transfer Size) of 8 in the Identify Controller data. This means the maximum I/O request size can be up to 1MB. To issue I/O requests of size 1MB for a Broadcom NVMe/FC host, you should increase the `lpfc` value of the `lpfc_sg_seg_cnt` parameter to 256 from the default value of 64.

> These steps don't apply to Qlogic NVMe/FC hosts.

Steps

1. Set the `lpfc_sg_seg_cnt` parameter to 256:

```
cat /etc/modprobe.d/lpfc.conf
```

You should see an output similar to the following example:

```
options lpfc lpfc_sg_seg_cnt=256
```

2. Run the `dracut -f` command, and reboot the host.
3. Verify that the value for `lpfc_sg_seg_cnt` is 256:

```
cat /sys/module/lpfc/parameters/lpfc_sg_seg_cnt
```

Verify NVMe services

Beginning with SUSE Linux Enterprise Server 15 SP6, the `nvmefc-boot-connections.service` and `nvmf-autoconnect.service` boot services included in the NVMe/FC `nvme-cli` package are automatically enabled to start during the system boot. After the system boot completes, you should verify that the boot services have been enabled.

Steps

1. Verify that `nvmf-autoconnect.service` is enabled:

```
systemctl status nvmf-autoconnect.service
```

Show example output

```
nvmf-autoconnect.service - Connect NVMe-oF subsystems automatically
during boot
  Loaded: loaded (/usr/lib/systemd/system/nvmf-autoconnect.service;
enabled; vendor preset: disabled)
  Active: inactive (dead) since Thu 2024-05-25 14:55:00 IST; 11min
ago
Process: 2108 ExecStartPre=/sbin/modprobe nvme-fabrics (code=exited,
status=0/SUCCESS)
Process: 2114 ExecStart=/usr/sbin/nvme connect-all (code=exited,
status=0/SUCCESS)
Main PID: 2114 (code=exited, status=0/SUCCESS)

systemd[1]: Starting Connect NVMe-oF subsystems automatically during
boot...
nvme[2114]: traddr=nn-0x201700a098fd4ca6:pn-0x201800a098fd4ca6 is
already connected
systemd[1]: nvmf-autoconnect.service: Deactivated successfully.
systemd[1]: Finished Connect NVMe-oF subsystems automatically during
boot.
```

2. Verify that `nvmefc-boot-connections.service` is enabled:

```
systemctl status nvmefc-boot-connections.service
```

Show example output

```
nvmefc-boot-connections.service - Auto-connect to subsystems on FC-
NVME devices found during boot
   Loaded: loaded (/usr/lib/systemd/system/nvmefc-boot-
connections.service; enabled; vendor preset: enabled)
   Active: inactive (dead) since Thu 2024-05-25 14:55:00 IST; 11min
ago
 Main PID: 1647 (code=exited, status=0/SUCCESS)

systemd[1]: Starting Auto-connect to subsystems on FC-NVME devices
found during boot...
systemd[1]: nvmefc-boot-connections.service: Succeeded.
systemd[1]: Finished Auto-connect to subsystems on FC-NVME devices
found during boot.
```

Configure NVMe/TCP

NVMe/TCP doesn't have an auto-connect functionality. Instead, you can discover the NVMe/TCP subsystems and namespaces by performing the NVMe/TCP `connect` or `connect-all` operations manually.

Steps

1. Verify that the initiator port can fetch the discovery log page data across the supported NVMe/TCP LIFs:

```
nvme discover -t tcp -w <host-traddr> -a <traddr>
```

Show example output

```
Discovery Log Number of Records 8, Generation counter 18
=====Discovery Log Entry 0======
trtype: tcp
adrfam: ipv4
subtype: current discovery subsystem
treq: not specified
portid: 4
trsvcid: 8009
subnqn: nqn.1992-
08.com.netapp:sn.8b5ee9199ff411eea468d039ea36a106:discovery
traddr: 192.168.211.67
eflags: explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 1======
trtype: tcp
adrfam: ipv4
subtype: current discovery subsystem
treq: not specified
portid: 2
trsvcid: 8009
subnqn: nqn.1992-
08.com.netapp:sn.8b5ee9199ff411eea468d039ea36a106:discovery
traddr: 192.168.111.67
eflags: explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 2======
trtype: tcp
adrfam: ipv4
subtype: current discovery subsystem
treq: not specified
portid: 3
trsvcid: 8009
subnqn: nqn.1992-
08.com.netapp:sn.8b5ee9199ff411eea468d039ea36a106:discovery
traddr: 192.168.211.66
eflags: explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 3======
trtype: tcp
adrfam: ipv4
subtype: current discovery subsystem
```

```
treq: not specified
portid: 1
trsvcid: 8009
subnqn: nqn.1992-
08.com.netapp:sn.8b5ee9199ff411eea468d039ea36a106:discovery
traddr: 192.168.111.66
eflags: explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 4======
trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 4
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.8b5ee9199ff411eea468d039ea36a106:subsystem.nvme_tcp
_1
traddr: 192.168.211.67
eflags: none
sectype: none
=====Discovery Log Entry 5======
trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 2
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.8b5ee9199ff411eea468d039ea36a106:subsystem.nvme_tcp
_1
traddr: 192.168.111.67
eflags: none
sectype: none
=====Discovery Log Entry 6======
trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 3
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.8b5ee9199ff411eea468d039ea36a106:subsystem.nvme_tcp
_1
traddr: 192.168.211.66
```

```
eflags: none
sectype: none
=====Discovery Log Entry 7======
trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 1
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.8b5ee9199ff411eea468d039ea36a106:subsystem.nvme_tcp
_1
traddr: 192.168.111.66
eflags: none
sectype: none
```

2. Verify that all other NVMe/TCP initiator-target LIF combinations can successfully fetch discovery log page data:

```
nvme discover -t tcp -w <host-traddr> -a <traddr>
```

Example output

```
nvme discover -t tcp -w 192.168.111.79 -a 192.168.111.66
nvme discover -t tcp -w 192.168.111.79 -a 192.168.111.67
nvme discover -t tcp -w 192.168.211.79 -a 192.168.211.66
nvme discover -t tcp -w 192.168.211.79 -a 192.168.211.67
```

3. Run the `nvme connect-all` command across all the supported NVMe/TCP initiator-target LIFs across the nodes:

```
nvme connect-all -t tcp -w <host-traddr> -a <traddr>
```

Example output

```
nvme connect-all -t tcp -w 192.168.111.79 -a 192.168.111.66
nvme connect-all -t tcp -w 192.168.111.79 -a 192.168.111.67
nvme connect-all -t tcp -w 192.168.211.79 -a 192.168.211.66
nvme connect-all -t tcp -w 192.168.211.79 -a 192.168.211.67
```

> Beginning with SUSE Linux Enterprise Server 15 SP6, the default setting for the NVMe/TCP `ctrl-loss-tmo` timeout is turned off. This means there is no limit on the number of retries (indefinite retry), and you don't need to manually configure a specific `ctrl-loss-tmo` timeout duration when using the `nvme connect` or `nvme connect-all` commands (option `-l`). Additonally, the NVMe/TCP controllers don't experience timeouts in the event of a path failure and remain connected indefinitely.

Validate NVMe-oF

Use the following procedure to validate NVMe-oF for a SUSE Linux Enterprise Server 15 SP6 with ONTAP configuration.

Steps

1. Verify that in-kernel NVMe multipath is enabled:

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

   The expected value is "Y".

2. Verify that the host has the correct controller model for the ONTAP NVMe namespaces:

   ```
   cat /sys/class/nvme-subsystem/nvme-subsys*/model
   ```

   Example output

   ```
   NetApp ONTAP Controller
   NetApp ONTAP Controller
   ```

3. Verify the NVMe I/O policy for the respective ONTAP NVMe I/O controller:

   ```
   cat /sys/class/nvme-subsystem/nvme-subsys*/iopolicy
   ```

   Example output

   ```
   round-robin
   round-robin
   ```

4. Verify that the ONTAP namespaces are visible to the host:

   ```
   nvme list -v
   ```

Show example output

```
Subsystem        Subsystem-NQN
Controllers
----------------
------------------------------------------------------------------------
----------------- ---------------------
nvme-subsys0     nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.unidir_d
hcha p  nvme0, nvme1, nvme2, nvme3

Device   SN                   MN
FR       TxPort Asdress        Subsystem    Namespaces
-------- --------------------
---------------------------------------- --------
---------------------------------------------
nvme0    81LGgBUqsI3EAAAAAAAE NetApp ONTAP Controller   FFFFFFFF tcp
traddr=192.168.111.66,trsvcid=4420,host_traddr=192.168.111.79 nvme-
subsys0 nvme0n1
nvme1    81LGgBUqsI3EAAAAAAAE NetApp ONTAP Controller   FFFFFFFF tcp
traddr=192.168.111.67,trsvcid=4420,host_traddr=192.168.111.79 nvme-
subsys0 nvme0n1
nvme2    81LGgBUqsI3EAAAAAAAE NetApp ONTAP Controller   FFFFFFFF tcp
traddr=192.168.211.66,trsvcid=4420,host_traddr=192.168.211.79 nvme-
subsys0 nvme0n1
nvme3    81LGgBUqsI3EAAAAAAAE NetApp ONTAP Controller   FFFFFFFF tcp
traddr=192.168.211.67,trsvcid=4420,host_traddr=192.168.211.79 nvme-
subsys0 nvme0n1
Device        Generic     NSID       Usage                 Format
Controllers
------------ ------------ ----------
--------------------------------------------------------------
/dev/nvme0n1 /dev/ng0n1   0x1     1.07  GB /   1.07  GB    4 KiB +
0 B   nvme0, nvme1, nvme2, nvme3
```

5. Verify that the controller state of each path is live and has the correct ANA status:

```
nvme list-subsys /dev/<subsystem_name>
```

NVMe/FC

```
nvme list-subsys /dev/nvme2n1
```

Show example output

```
nvme-subsys2 - NQN=nqn.1992-
08.com.netapp:sn.06303c519d8411eea468d039ea36a106:subs
ystem.nvme
 hostnqn=nqn.2014-08.org.nvmexpress:uuid:4c4c4544-
0056-5410-8048-c6c04f425633
 iopolicy=round-robin
\
+- nvme4 fc traddr=nn-0x208fd039ea359e4a:pn-
0x210dd039ea359e4a,host_traddr=nn-0x2000f4c7aa0cd7ab:pn-
0x2100f4c7aa0cd7ab live optimized
+- nvme6 fc traddr=nn-0x208fd039ea359e4a:pn-
0x210ad039ea359e4a,host_traddr=nn-0x2000f4c7aa0cd7aa:pn-
0x2100f4c7aa0cd7aa live optimized
```

NVMe/TCP

```
nvme list-subsys
```

Show example output

```
nvme-subsys1 - NQN=nqn.1992-
08.com.netapp:sn.8b5ee9199ff411eea468d039ea36a106:subsystem.nvme
_tcp_1
 hostnqn=nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0035-5910-
804b-b2c04f444d33
 iopolicy=round-robin
\
+- nvme4 tcp
traddr=192.168.111.66,trsvcid=4420,host_traddr=192.168.111.79,sr
c_addr=192.168.111.79 live
+- nvme3 tcp
traddr=192.168.211.66,trsvcid=4420,host_traddr=192.168.211.79,sr
c_addr=192.168.111.79 live
+- nvme2 tcp
traddr=192.168.111.67,trsvcid=4420,host_traddr=192.168.111.79,sr
c_addr=192.168.111.79 live
+- nvme1 tcp
traddr=192.168.211.67,trsvcid=4420,host_traddr=192.168.211.79,sr
c_addr=192.168.111.79 live
```

6. Verify that the NetApp plug-in displays the correct values for each ONTAP namespace device:

Column

```
nvme netapp ontapdevices -o column
```

Example output

```
Device           Vserver    Namespace Path
NSID UUID                                   Size
---------------- ---------- ------------------------------------
-------------------------------------------- --------
/dev/nvme0n1     vs_192     /vol/fcnvme_vol_1_1_0/fcnvme_ns      1
c6586535-da8a-40fa-8c20-759ea0d69d33   20GB
```

JSON

```
nvme netapp ontapdevices -o json
```

Show example output

```
{
"ONTAPdevices":[
{
"Device":"/dev/nvme0n1",
"Vserver":"vs_192",
"Namespace_Path":"/vol/fcnvme_vol_1_1_0/fcnvme_ns",
"NSID":1,
"UUID":"c6586535-da8a-40fa-8c20-759ea0d69d33",
"Size":"20GB",
"LBA_Data_Size":4096,
"Namespace_Size":262144
}
]
}
```

Create a persistent discovery controller

Beginning with ONTAP 9.11.1, you can create a persistent discovery controller (PDC) for a SUSE Linux Enterprise Server 15 SP6 host. A PDC is required to automatically detect an NVMe subsystem add or remove operation and changes to the discovery log page data.

Steps

1. Verify that the discovery log page data is available and can be retrieved through the initiator port and target LIF combination:

```
nvme discover -t <trtype> -w <host-traddr> -a <traddr>
```

Show example output

```
Discovery Log Number of Records 8, Generation counter 18
=====Discovery Log Entry 0======
trtype: tcp
adrfam: ipv4
subtype: current discovery subsystem
treq: not specified
portid: 4
trsvcid: 8009
subnqn: nqn.1992-
08.com.netapp:sn.8b5ee9199ff411eea468d039ea36a106:discovery
traddr: 192.168.211.67
eflags: explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 1======
trtype: tcp
adrfam: ipv4
subtype: current discovery subsystem
treq: not specified
portid: 2
trsvcid: 8009
subnqn: nqn.1992-
08.com.netapp:sn.8b5ee9199ff411eea468d039ea36a106:discovery
traddr: 192.168.111.67
eflags: explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 2======
trtype: tcp
adrfam: ipv4
subtype: current discovery subsystem
treq: not specified
portid: 3
trsvcid: 8009
subnqn: nqn.1992-
08.com.netapp:sn.8b5ee9199ff411eea468d039ea36a106:discovery
traddr: 192.168.211.66
eflags: explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 3======
trtype: tcp
adrfam: ipv4
subtype: current discovery subsystem
```

```
treq: not specified
portid: 1
trsvcid: 8009
subnqn: nqn.1992-
08.com.netapp:sn.8b5ee9199ff411eea468d039ea36a106:discovery
traddr: 192.168.111.66
eflags: explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 4======
trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 4
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.8b5ee9199ff411eea468d039ea36a106:subsystem.nvme_tcp
_1
traddr: 192.168.211.67
eflags: none
sectype: none
=====Discovery Log Entry 5======
trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 2
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.8b5ee9199ff411eea468d039ea36a106:subsystem.nvme_tcp
_1
traddr: 192.168.111.67
eflags: none
sectype: none
=====Discovery Log Entry 6======
trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 3
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.8b5ee9199ff411eea468d039ea36a106:subsystem.nvme_tcp
_1
traddr: 192.168.211.66
```

```
eflags: none
sectype: none
=====Discovery Log Entry 7======
trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 1
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.8b5ee9199ff411eea468d039ea36a106:subsystem.nvme_tcp
_1
traddr: 192.168.111.66
eflags: none
sectype: none
```

2. Create a PDC for the discovery subsystem:

```
nvme discover -t <trtype> -w <host-traddr> -a <traddr> -p
```

Example output

```
nvme discover -t tcp -w 192.168.111.79 -a 192.168.111.666 -p
```

3. From the ONTAP controller, verify that the PDC has been created:

```
vserver nvme show-discovery-controller -instance -vserver <vserver_name>
```

Show example output

```
vserver nvme show-discovery-controller -instance -vserver vs_nvme79
Vserver Name: vs_CLIENT116 Controller ID: 00C0h
Discovery Subsystem NQN: nqn.1992-
08.com.netapp:sn.48391d66c0a611ecaaa5d039ea165514:discovery Logical
Interface UUID: d23cbb0a-c0a6-11ec-9731-d039ea165abc Logical
Interface:
CLIENT116_lif_4a_1
Node: A400-14-124
Host NQN: nqn.2014-08.org.nvmexpress:uuid:12372496-59c4-4d1b-be09-
74362c0c1afc
Transport Protocol: nvme-tcp
Initiator Transport Address: 192.168.1.16
Host Identifier: 59de25be738348f08a79df4bce9573f3 Admin Queue Depth:
32
Header Digest Enabled: false Data Digest Enabled: false
Vserver UUID: 48391d66-c0a6-11ec-aaa5-d039ea165514
```

Set up secure in-band authentication

Beginning with ONTAP 9.12.1, secure in-band authentication is supported over NVMe/TCP and NVMe/FC between a SUSE Linux Enterprise Server 15 SP6 host and an ONTAP controller.

To set up secure authentication, each host or controller must be associated with a `DH-HMAC-CHAP` key, which is a combination of the NQN of the NVMe host or controller and an authentication secret configured by the administrator. To authenticate its peer, an NVMe host or controller must recognize the key associated with the peer.

You can set up secure in-band authentication using the CLI or a config JSON file. If you need to specify different dhchap keys for different subsystems, you must use a config JSON file.

CLI

Set up secure in-band authentication using the CLI.

Steps

1. Obtain the host NQN:

```
cat /etc/nvme/hostnqn
```

2. Generate the dhchap key for the SUSE Linux Enterprise Server 15 SP6 host.

   The following output describes the `gen-dhchap-key` command paramters:

```
nvme gen-dhchap-key -s optional_secret -l key_length {32|48|64} -m
HMAC_function {0|1|2|3} -n host_nqn
•   -s secret key in hexadecimal characters to be used to initialize
the host key
•   -l length of the resulting key in bytes
•   -m HMAC function to use for key transformation
0 = none, 1- SHA-256, 2 = SHA-384, 3=SHA-512
•   -n host NQN to use for key transformation
```

   In the following example, a random dhchap key with HMAC set to 3 (SHA-512) is generated.

```
nvme gen-dhchap-key -m 3 -n nqn.2014-
08.org.nvmexpress:uuid:d3ca725a- ac8d-4d88-b46a-174ac235139b
DHHC-
1:03:J2UJQfj9f0pLnpF/ASDJRTyILKJRr5CougGpGdQSysPrLu6RW1fGl5VSjbeDF1n
1DEh3nVBe19nQ/LxreSBeH/bx/pU=:
```

3. On the ONTAP controller, add the host and specify both dhchap keys:

```
vserver nvme subsystem host add -vserver <svm_name> -subsystem
<subsystem> -host-nqn <host_nqn> -dhchap-host-secret
<authentication_host_secret> -dhchap-controller-secret
<authentication_controller_secret> -dhchap-hash-function {sha-
256|sha-512} -dhchap-group {none|2048-bit|3072-bit|4096-bit|6144-
bit|8192-bit}
```

4. A host supports two types of authentication methods, unidirectional and bidirectional. On the host, connect to the ONTAP controller and specify dhchap keys based on the chosen authentication method:

```
nvme connect -t tcp -w <host-traddr> -a <tr-addr> -n <host_nqn> -S
<authentication_host_secret> -C <authentication_controller_secret>
```

5. Validate the `nvme connect authentication` command by verifying the host and controller dhchap keys:

   a. Verify the host dhchap keys:

   ```
   cat /sys/class/nvme-subsystem/<nvme-subsysX>/nvme*/dhchap_secret
   ```

   Show example output for a unidirectional configuration

   ```
   cat /sys/class/nvme-subsystem/nvme-subsys1/nvme*/dhchap_secret
   DHHC-
   1:03:je1nQCmjJLUKD62mpYbzlpuw0OIws86NB96uNO/t3jbvhp7fjyR9bIRjO
   Hg8wQtye1JCFSMkBQH3pTKGdYR1OV9gx00=:
   DHHC-
   1:03:je1nQCmjJLUKD62mpYbzlpuw0OIws86NB96uNO/t3jbvhp7fjyR9bIRjO
   Hg8wQtye1JCFSMkBQH3pTKGdYR1OV9gx00=:
   DHHC-
   1:03:je1nQCmjJLUKD62mpYbzlpuw0OIws86NB96uNO/t3jbvhp7fjyR9bIRjO
   Hg8wQtye1JCFSMkBQH3pTKGdYR1OV9gx00=:
   DHHC-
   1:03:je1nQCmjJLUKD62mpYbzlpuw0OIws86NB96uNO/t3jbvhp7fjyR9bIRjO
   Hg8wQtye1JCFSMkBQH3pTKGdYR1OV9gx00=:
   ```

   b. Verify the controller dhchap keys:

   ```
   cat /sys/class/nvme-subsystem/<nvme-
   subsysX>/nvme*/dhchap_ctrl_secret
   ```

Show example output for a bidirectional configuration

```
cat /sys/class/nvme-subsystem/nvme-
subsys6/nvme*/dhchap_ctrl_secret
DHHC-
1:03:WorVEV83eYO53kV4Iel5OpphbX5LAphO3F8fgH3913tlrkSGDBJTt3crX
eTUB8fCwGbPsEyz6CXxdQJi6kbn4IzmkFU=:
DHHC-
1:03:WorVEV83eYO53kV4Iel5OpphbX5LAphO3F8fgH3913tlrkSGDBJTt3crX
eTUB8fCwGbPsEyz6CXxdQJi6kbn4IzmkFU=:
DHHC-
1:03:WorVEV83eYO53kV4Iel5OpphbX5LAphO3F8fgH3913tlrkSGDBJTt3crX
eTUB8fCwGbPsEyz6CXxdQJi6kbn4IzmkFU=:
DHHC-
1:03:WorVEV83eYO53kV4Iel5OpphbX5LAphO3F8fgH3913tlrkSGDBJTt3crX
eTUB8fCwGbPsEyz6CXxdQJi6kbn4IzmkFU=:
```

JSON file

When multiple NVMe subsystems are available on the ONTAP controller configuration, you can use the `/etc/nvme/config.json` file with the `nvme connect-all` command.

To generate the JSON file, you can use the `-o` option. See the NVMe connect-all manual pages for more syntax options.

Steps

1. Configure the JSON file:

Show example output

```
cat /etc/nvme/config.json
[
 {
    "hostnqn":"nqn.2014-08.org.nvmexpress:uuid:12372496-59c4-
4d1b-be09-74362c0c1afc",
    "hostid":"3ae10b42-21af-48ce-a40b-cfb5bad81839",
    "dhchap_key":"DHHC-
1:03:Cu3ZZfIz1WMlqZFnCMqpAgn/T6EVOcIFHez215U+Pow8jTgBF2UbNk3DK4w
fk2EptWpna1rpwG5CndpOgxpRxh9m41w=:"
 },
 {
    "hostnqn":"nqn.2014-08.org.nvmexpress:uuid:12372496-59c4-
4d1b-be09-74362c0c1afc",
    "subsystems":[
        {
            "nqn":"nqn.1992-
08.com.netapp:sn.48391d66c0a611ecaaa5d039ea165514:subsystem.subs
ys_CLIENT116",
            "ports":[
               {
                    "transport":"tcp",
                    "traddr":" 192.168.111.66 ",
                    "host_traddr":" 192.168.111.79",
                    "trsvcid":"4420",
                    "dhchap_ctrl_key":"DHHC-
1:01:0h58bcT/uu0rCpGsDYU6ZHZvRuVqsYKuBRS0Nu0VPx5HEwaZ:"
               },
               {
                    "transport":"tcp",
                    "traddr":" 192.168.111.66 ",
                    "host_traddr":" 192.168.111.79",
                    "trsvcid":"4420",
                    "dhchap_ctrl_key":"DHHC-
1:01:0h58bcT/uu0rCpGsDYU6ZHZvRuVqsYKuBRS0Nu0VPx5HEwaZ:"
               },
               {
                    "transport":"tcp",
                   "traddr":" 192.168.111.66 ",
                    "host_traddr":" 192.168.111.79",
                    "trsvcid":"4420",
                    "dhchap_ctrl_key":"DHHC-
1:01:0h58bcT/uu0rCpGsDYU6ZHZvRuVqsYKuBRS0Nu0VPx5HEwaZ:"
               },
               {
```

```
                    "transport":"tcp",
                    "traddr":" 192.168.111.66 ",
                    "host_traddr":" 192.168.111.79",
                    "trsvcid":"4420",
                    "dhchap_ctrl_key":"DHHC-
1:01:0h58bcT/uu0rCpGsDYU6ZHZvRuVqsYKuBRS0Nu0VPx5HEwaZ:"
               }
           ]
       }
   ]
 }
]
```

+

In the preceding example, `dhchap_key` corresponds to `dhchap_secret` and `dhchap_ctrl_key` corresponds to `dhchap_ctrl_secret`.

2. Connect to the ONTAP controller using the config JSON file:

```
nvme connect-all -J /etc/nvme/config.json
```

Show example output

```
traddr=192.168.111.66 is already connected
traddr=192.168.211.66 is already connected
traddr=192.168.111.66 is already connected
traddr=192.168.211.66 is already connected
traddr=192.168.111.66 is already connected
traddr=192.168.211.66 is already connected
traddr=192.168.111.67 is already connected
traddr=192.168.211.67 is already connected
traddr=192.168.111.67 is already connected
traddr=192.168.211.67 is already connected
traddr=192.168.111.67 is already connected
traddr=192.168.111.67 is already connected
```

3. Verify that the dhchap secrets have been enabled for the respective controllers for each subsystem:
   a. Verify the host dhchap keys:

```
cat /sys/class/nvme-subsystem/nvme-subsys0/nvme0/dhchap_secret
```

Example output

```
DHHC-1:01:NunEWY7AZlXqxITGheByarwZdQvU4ebZg9HOjIr6nOHEkxJg:
```

b. Verify the controller dhchap keys:

```
cat /sys/class/nvme-subsystem/nvme-
subsys0/nvme0/dhchap_ctrl_secret
```

Example output

```
DHHC-
1:03:2YJinsxa2v3+m8qqCiTnmgBZoH6mIT6G/6f0aGO8viVZB4VLNLH4z8CvK7pV
YxN6S5fOAtaU3DNi12rieRMfdbg3704=:
```

Configure Transport Layer Security

Transport Layer Security (TLS) provides secure end-to-end encryption for NVMe connections between NVMe-oF hosts and an ONTAP array. Beginning with ONTAP 9.16.1, you can configure TLS 1.3 using the CLI and a configured pre-shared key (PSK).

About this task

You perform the steps in this procedure on the SUSE Linux Enterprise Server host, except where it specifies that you perform a step on the ONTAP controller.

Steps

1. Check that you have the following ktls-utils, openssl, and libopenssl packages installed on the host:

   a. `rpm -qa | grep ktls`

      Example output

      ```
      ktls-utils-0.10+12.gc3923f7-150600.1.2.x86_64
      ```

   b. `rpm -qa | grep ssl`

      Example output

      ```
      openssl-3-3.1.4-150600.5.7.1.x86_64
      libopenssl1_1-1.1.1w-150600.5.3.1.x86_64
      libopenssl3-3.1.4-150600.5.7.1.x86_64
      ```

2. Verify that you have the correct setup for `/etc/tlshd.conf`:

```
cat /etc/tlshd.conf
```

Show example output

```
[debug]
loglevel=0
tls=0
nl=0
[authenticate]
keyrings=.nvme
[authenticate.client]
#x509.truststore= <pathname>
#x509.certificate= <pathname>
#x509.private_key= <pathname>
[authenticate.server]
#x509.truststore= <pathname>
#x509.certificate= <pathname>
#x509.private_key= <pathname>
```

3. Enable `tlshd` to start at system boot:

```
systemctl enable tlshd
```

4. Verify that the `tlshd` daemon is running:

```
systemctl status tlshd
```

Show example output

```
tlshd.service - Handshake service for kernel TLS consumers
   Loaded: loaded (/usr/lib/systemd/system/tlshd.service; enabled;
preset: disabled)
   Active: active (running) since Wed 2024-08-21 15:46:53 IST; 4h
57min ago
     Docs: man:tlshd(8)
Main PID: 961 (tlshd)
   Tasks: 1
     CPU: 46ms
   CGroup: /system.slice/tlshd.service
       └─961 /usr/sbin/tlshd
Aug 21 15:46:54 RX2530-M4-17-153 tlshd[961]: Built from ktls-utils
0.11-dev on Mar 21 2024 12:00:00
```

5. Generate the TLS PSK by using the `nvme gen-tls-key`:

   a. `cat /etc/nvme/hostnqn`

      Example output

      ```
      nqn.2014-08.org.nvmexpress:uuid:e58eca24-faff-11ea-8fee-3a68dd3b5c5f
      ```

   b. `nvme gen-tls-key --hmac=1 --identity=1 --subsysnqn=nqn.1992-08.com.netapp:sn.1d59a6b2416b11ef9ed5d039ea50acb3:subsystem.sles15`

      Example output

      ```
      NVMeTLSkey-1:01:dNcby017axByCko8GivzOO9zGlgHDXJCN6KLzvYoA+NpT1uD:
      ```

6. On the ONTAP array, add the TLS PSK to the ONTAP subsystem:

   ```
   vserver nvme subsystem host add -vserver sles15_tls -subsystem sles15
   -host-nqn nqn.2014-08.org.nvmexpress:uuid:e58eca24-faff-11ea-8fee-
   3a68dd3b5c5f -tls-configured-psk NVMeTLSkey-
   1:01:dNcby017axByCko8GivzOO9zGlgHDXJCN6KLzvYoA+NpT1uD:
   ```

7. On the SUSE Linux Enterprise Server host, insert the TLS PSK into the host kernel keyring:

```
nvme check-tls-key --identity=1 --subsysnqn =nqn.1992-
08.com.netapp:sn.1d59a6b2416b11ef9ed5d039ea50acb3:subsystem.sles15
--keydata=NVMeTLSkey
-1:01:dNcby017axByCko8GivzOO9zGlgHDXJCN6KLzvYoA+NpT1uD: --insert
```

Example output

```
Inserted TLS key 22152a7e
```

> The PSK shows as "NVMe1R01" because it uses "identity v1" from the TLS handshake algorithm. Identity v1 is the only version that ONTAP supports.

8. Verify that the TLS PSK is inserted correctly:

```
cat /proc/keys | grep NVMe
```

Example output

```
22152a7e I--Q---     1 perm 3b010000     0     0 psk       NVMe1R01
nqn.2014-08.org.nvmexpress:uuid:ffa0c815-e28b-4bb1-8d4c-7c6d5e610bfc
nqn.1992-
08.com.netapp:sn.1d59a6b2416b11ef9ed5d039ea50acb3:subsystem.sles15
UoP9dEfvuCUzzpS0DYxnshKDapZYmvA0/RJJ8JAqmAo=: 32
```

9. On the SUSE Linux Enterprise Server host, connect to the ONTAP subsystem using the inserted TLS PSK:

   a. `nvme connect -t tcp -w 20.20.10.80 -a 20.20.10.14 -n nqn.1992-08.com.netapp:sn.1d59a6b2416b11ef9ed5d039ea50acb3:subsystem.sles15 --tls_key=0x22152a7e --tls`

      Example output

      ```
      connecting to device: nvme0
      ```

   b. `nvme list-subsys`

Example output

```
nvme-subsys0 - NQN=nqn.1992-
08.com.netapp:sn.1d59a6b2416b11ef9ed5d039ea50acb3:subsystem.sles15
               hostnqn=nqn.2014-08.org.nvmexpress:uuid:ffa0c815-e28b-
4bb1-8d4c-7c6d5e610bfc
               iopolicy=round-robin
\
 +- nvme0 tcp
traddr=20.20.10.14,trsvcid=4420,host_traddr=20.20.10.80,src_addr=20.2
0.10.80 live
```

10. Add the target, and verify the TLS connection to the specified ONTAP subsystem:

```
nvme subsystem controller show -vserver sles15_tls -subsystem sles15 -instance
```

Show example output

```
  (vserver nvme subsystem controller show)
                      Vserver Name: sles15_tls
                         Subsystem: sles15
                     Controller ID: 0040h
                 Logical Interface: sles15t_e1a_1
                              Node: A900-17-174
                          Host NQN: nqn.2014-
08.org.nvmexpress:uuid:ffa0c815-e28b-4bb1-8d4c-7c6d5e610bfc
                Transport Protocol: nvme-tcp
       Initiator Transport Address: 20.20.10.80
                   Host Identifier:
ffa0c815e28b4bb18d4c7c6d5e610bfc
              Number of I/O Queues: 4
                  I/O Queue Depths: 128, 128, 128, 128
                 Admin Queue Depth: 32
             Max I/O Size in Bytes: 1048576
         Keep-Alive Timeout (msec): 5000
                      Vserver UUID: 1d59a6b2-416b-11ef-9ed5-
d039ea50acb3
                    Subsystem UUID: 9b81e3c5-5037-11ef-8a90-
d039ea50ac83
            Logical Interface UUID: 8185dcac-5035-11ef-8abb-
d039ea50acb3
             Header Digest Enabled: false
               Data Digest Enabled: false
      Authentication Hash Function: -
Authentication Diffie-Hellman Group: -
               Authentication Mode: none
      Transport Service Identifier: 4420
                      TLS Key Type: configured
                  TLS PSK Identity: NVMe1R01 nqn.2014-
08.org.nvmexpress:uuid:ffa0c815-e28b-4bb1-8d4c-7c6d5e610bfc
nqn.1992-
08.com.netapp:sn.1d59a6b2416b11ef9ed5d039ea50acb3:subsystem.sles15
UoP9dEfvuCUzzpS0DYxnshKDapZYmvA0/RJJ8JAqmAo=
                        TLS Cipher: TLS-AES-128-GCM-SHA256
```

Known issues

There are no known issues.

NVMe-oF host configuration for SUSE Linux Enterprise Server 15 SP5 with ONTAP

NVMe over Fabrics (NVMe-oF), including NVMe over Fibre Channel (NVMe/FC) and other transports, is supported for SUSE Linux Enterprise Server 15 SP5 with Asymmetric Namespace Access (ANA). In NVMe-oF environments, ANA is the equivalent of ALUA multipathing in iSCSI and FCP environments and is implemented with in-kernel NVMe multipath.

The following support is available for the NVMe-oF host configuration for SUSE Linux Enterprise Server 15 SP5 with ONTAP:

- Both NVMe and SCSI traffic can be run on the same host. Therefore, for SCSI LUNs, you can configure dm-multipath for SCSI mpath devices, whereas you might use NVMe multipath to configure NVMe-oF namespace devices on the host.
- Support for NVMe over TCP (NVMe/TCP) in addition to NVMe/FC. The NetApp plug-in in the native `nvme-cli` package displays ONTAP details for both NVMe/FC and NVMe/TCP namespaces.

For additional details on supported configurations, see the Interoperability Matrix Tool.

Features

- Support for NVMe secure, in-band authentication
- Support for persistent discovery controllers (PDCs) using a unique discovery NQN

Known limitations

- SAN booting using the NVMe-oF protocol is currently not supported.
- There's no `sanlun` support for NVMe-oF. Therefore, the host utility support isn't available for NVMe-oF on an SUSE Linux Enterprise Server 15 SP5 host. You can use the NetApp plug-in included in the native nvme-cli package for all NVMe-oF transports.

Configure NVMe/FC

You can configure NVMe/FC for Broadcom/Emulex FC or Marvell/Qlogic FC adapters.

Broadcom/Emulex

Steps

1. Verify that you are using the recommended adapter model:

   ```
   cat /sys/class/scsi_host/host*/modelname
   ```

   Example output:

   ```
   LPe32002 M2
   LPe32002-M2
   ```

2. Verify the adapter model description:

   ```
   cat /sys/class/scsi_host/host*/modeldesc
   ```

   Example output:

   ```
   Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
   Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
   ```

3. Verify that you are using the recommended Emulex host bus adapter (HBA) firmware versions:

   ```
   cat /sys/class/scsi_host/host*/fwrev
   ```

   Example output:

   ```
   14.0.639.20, sli-4:2:c
   14.0.639.20, sli-4:2:c
   ```

4. Verify that you are using the recommended LPFC driver version:

   ```
   cat /sys/module/lpfc/version
   ```

   Example output:

   ```
   0:14.2.0.13
   ```

5. Verify that you can view your initiator ports:

```
cat /sys/class/fc_host/host*/port_name
```

Example output:

```
0x100000109b579d5e
0x100000109b579d5f
```

6. Verify that your initiator ports are online:

```
cat /sys/class/fc_host/host*/port_state
```

Example output:

```
Online
Online
```

7. Verify that the NVMe/FC initiator ports are enabled and that the target ports are visible:

```
cat /sys/class/scsi_host/host*/nvme_info
```

Example output:

In the following example, one initiator port is enabled and connected with two target LIFs.

```
NVME Initiator Enabled
XRI Dist lpfc0 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc0 WWPN x100000109b579d5e WWNN x200000109b579d5e DID
x011c00 ONLINE
NVME RPORT WWPN x208400a098dfdd91 WWNN x208100a098dfdd91 DID x011503
TARGET DISCSRVC ONLINE
NVME RPORT WWPN x208500a098dfdd91 WWNN x208100a098dfdd91 DID x010003
TARGET DISCSRVC *ONLINE

NVME Statistics
LS: Xmt 0000000e49 Cmpl 0000000e49 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000003ceb594f Issue 000000003ce65dbe OutIO
fffffffffffb046f
abort 00000bd2 noxri 00000000 nondlp 00000000 qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 000014f4 Err 00012abd

NVME Initiator Enabled
XRI Dist lpfc1 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc1 WWPN x100000109b579d5f WWNN x200000109b579d5f DID
x011b00 ONLINE
NVME RPORT WWPN x208300a098dfdd91 WWNN x208100a098dfdd91 DID x010c03
TARGET DISCSRVC ONLINE
NVME RPORT WWPN x208200a098dfdd91 WWNN x208100a098dfdd91 DID x012a03
TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 0000000e50 Cmpl 0000000e50 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000003c9859ca Issue 000000003c93515e OutIO
fffffffffffaf794
abort 00000b73 noxri 00000000 nondlp 00000000 qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 0000159d Err 000135c3
```

8. Reboot the host.

Marvell/QLogic

The native inbox qla2xxx driver included in the SUSE Linux Enterprise Server 15 SP5 kernel has the latest fixes. These fixes are essential for ONTAP support.

Steps

1. Verify that you are running the supported adapter driver and firmware versions:

```
cat /sys/class/fc_host/host*/symbolic_name
```

Example output:

```
QLE2742 FW:v9.12.01 DVR: v10.02.08.300-k
QLE2742 FW:v9.12.01 DVR: v10.02.08.300-k
```

2. Verify that the `ql2xnvmeenable` parameter is set to 1:

```
cat /sys/module/qla2xxx/parameters/ql2xnvmeenable
1
```

Enable 1MB I/O size (Optional)

ONTAP reports an MDTS (Max Data Transfer Size) of 8 in the Identify Controller data. This means the maximum I/O request size can be up to 1MB. To issue I/O requests of size 1MB for a Broadcom NVMe/FC host, you should increase the `lpfc` value of the `lpfc_sg_seg_cnt` parameter to 256 from the default value of 64.

> These steps don't apply to Qlogic NVMe/FC hosts.

Steps

1. Set the `lpfc_sg_seg_cnt` parameter to 256:

```
cat /etc/modprobe.d/lpfc.conf
```

You should see an output similar to the following example:

```
options lpfc lpfc_sg_seg_cnt=256
```

2. Run the `dracut -f` command, and reboot the host.
3. Verify that the value for `lpfc_sg_seg_cnt` is 256:

```
cat /sys/module/lpfc/parameters/lpfc_sg_seg_cnt
```

Enable NVMe services

There are two NVMe/FC boot services included in the `nvme-cli` package, however, *only* `nvmefc-boot-connections.service` is enabled to start during system boot; `nvmf-autoconnect.service` is not enabled. Therefore, you need to manually enable `nvmf-autoconnect.service` to start during system boot.

Steps

1. Enable `nvmf-autoconnect.service`:

```
# systemctl enable nvmf-autoconnect.service
Created symlink /etc/systemd/system/default.target.wants/nvmf-
autoconnect.service → /usr/lib/systemd/system/nvmf-autoconnect.service.
```

2. Reboot the host.
3. Verify that `nvmf-autoconnect.service` and `nvmefc-boot-connections.service` are running after the system boot:

   Example output:

```
# systemctl status nvmf-autoconnect.service
nvmf-autoconnect.service - Connect NVMe-oF subsystems automatically
during boot
Loaded: loaded (/usr/lib/systemd/system/nvmf-autoconnect.service;
enabled; vendor preset: disabled)
Active: inactive (dead) since Thu 2023-05-25 14:55:00 IST; 11min
ago
Process: 2108 ExecStartPre=/sbin/modprobe nvme-fabrics (code=exited,
status=0/SUCCESS)
Process: 2114 ExecStart=/usr/sbin/nvme connect-all (code=exited,
status=0/SUCCESS)
Main PID: 2114 (code=exited, status=0/SUCCESS)

systemd[1]: Starting Connect NVMe-oF subsystems automatically during
boot...
nvme[2114]: traddr=nn-0x201700a098fd4ca6:pn-0x201800a098fd4ca6 is
already connected
systemd[1]: nvmf-autoconnect.service: Deactivated successfully.
systemd[1]: Finished Connect NVMe-oF subsystems automatically during
boot.

# systemctl status nvmefc-boot-connections.service
nvmefc-boot-connections.service - Auto-connect to subsystems on FC-NVME
devices found during boot
Loaded: loaded (/usr/lib/systemd/system/nvmefc-boot-
connections.service; enabled; vendor preset: enabled)
Active: inactive (dead) since Thu 2023-05-25 14:55:00 IST; 11min ago
Main PID: 1647 (code=exited, status=0/SUCCESS)

systemd[1]: Starting Auto-connect to subsystems on FC-NVME devices found
during boot...
systemd[1]: nvmefc-boot-connections.service: Succeeded.
systemd[1]: Finished Auto-connect to subsystems on FC-NVME devices found
during boot.
```

Configure NVMe/TCP

You can use the following procedure to configure NVMe/TCP.

Steps

1. Verify that the initiator port can fetch the discovery log page data across the supported NVMe/TCP LIFs:

```
nvme discover -t tcp -w <host-traddr> -a <traddr>
```

Example output:

```
# nvme discover -t tcp -w 192.168.1.4 -a 192.168.1.31

Discovery Log Number of Records 8, Generation counter 18
=====Discovery Log Entry 0====== trtype: tcp
adrfam: ipv4
subtype: current discovery subsystem treq: not specified
portid: 0
trsvcid: 8009 subnqn: nqn.1992-
08.com.netapp:sn.48391d66c0a611ecaaa5d039ea165514:discovery traddr:
192.168.2.117
eflags: explicit discovery connections, duplicate discovery information
sectype: none
=====Discovery Log Entry 1====== trtype: tcp
adrfam: ipv4
subtype: current discovery subsystem treq: not specified
portid: 1
trsvcid: 8009 subnqn: nqn.1992-
08.com.netapp:sn.48391d66c0a611ecaaa5d039ea165514:discovery traddr:
192.168.1.117
eflags: explicit discovery connections, duplicate discovery information
sectype: none
=====Discovery Log Entry 2====== trtype: tcp
adrfam: ipv4
subtype: current discovery subsystem treq: not specified
portid: 2
trsvcid: 8009 subnqn: nqn.1992-
08.com.netapp:sn.48391d66c0a611ecaaa5d039ea165514:discovery traddr:
192.168.2.116
eflags: explicit discovery connections, duplicate discovery information
sectype: none
=====Discovery Log Entry 3====== trtype: tcp
adrfam: ipv4
subtype: current discovery subsystem treq: not specified
portid: 3
trsvcid: 8009 subnqn: nqn.1992-
08.com.netapp:sn.48391d66c0a611ecaaa5d039ea165514:discovery traddr:
192.168.1.116
eflags: explicit discovery connections, duplicate discovery information
sectype: none
=====Discovery Log Entry 4====== trtype: tcp
adrfam: ipv4
subtype: nvme subsystem treq: not specified portid: 0
trsvcid: 4420 subnqn: nqn.1992-
08.com.netapp:sn.48391d66c0a611ecaaa5d039ea165514:subsystem.subsys_CLIEN
```

```
T116
traddr: 192.168.2.117 eflags: not specified sectype: none
=====Discovery Log Entry 5====== trtype: tcp
adrfam: ipv4
subtype: nvme subsystem treq: not specified portid: 1
trsvcid: 4420 subnqn: nqn.1992-
08.com.netapp:sn.48391d66c0a611ecaaa5d039ea165514:subsystem.subsys_CLIEN
T116
traddr: 192.168.1.117 eflags: not specified sectype: none
=====Discovery Log Entry 6====== trtype: tcp
adrfam: ipv4
subtype: nvme subsystem treq: not specified portid: 2
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.48391d66c0a611ecaaa5d039ea165514:subsystem.subsys_CLIEN
T116
traddr: 192.168.2.116 eflags: not specified sectype: none
=====Discovery Log Entry 7====== trtype: tcp
adrfam: ipv4
subtype: nvme subsystem treq: not specified portid: 3
trsvcid: 4420 subnqn: nqn.1992-
08.com.netapp:sn.48391d66c0a611ecaaa5d039ea165514:subsystem.subsys_CLIEN
T116
traddr: 192.168.1.116 eflags: not specified sectype: none
```

2. Verify that all other NVMe/TCP initiator-target LIF combinations can successfully fetch discovery log page data:

```
nvme discover -t tcp -w <host-traddr> -a <traddr>
```

Example output:

```
# nvme discover -t tcp -w 192.168.1.4 -a 192.168.1.32
# nvme discover -t tcp -w 192.168.2.5 -a 192.168.2.36
# nvme discover -t tcp -w 192.168.2.5 -a 192.168.2.37
```

3. Run the `nvme connect-all` command across all the supported NVMe/TCP initiator-target LIFs across the nodes:

```
nvme connect-all -t tcp -w host-traddr -a traddr -l
<ctrl_loss_timeout_in_seconds>
```

Example output:

```
# nvme connect-all -t tcp -w 192.168.1.4 -a 192.168.1.31 -l -1
# nvme connect-all -t tcp -w 192.168.1.4 -a 192.168.1.32 -l -1
# nvme connect-all -t tcp -w 192.168.2.5 -a 192.168.1.36 -l -1
# nvme connect-all -t tcp -w 192.168.2.5 -a 192.168.1.37 -l -1
```

> NetApp recommends setting the `ctrl-loss-tmo` option to `-1` so that the NVMe/TCP initiator attempts to reconnect indefinitely in the event of a path loss.

Validate NVMe-oF

You can use the following procedure to validate NVMe-oF.

Steps

1. Verify that in-kernel NVMe multipath is enabled:

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

2. Verify that the host has the correct controller model for the ONTAP NVMe namespaces:

```
cat /sys/class/nvme-subsystem/nvme-subsys*/model
```

Example output:

```
NetApp ONTAP Controller
NetApp ONTAP Controller
```

3. Verify the NVMe I/O policy for the respective ONTAP NVMe I/O controller:

```
cat /sys/class/nvme-subsystem/nvme-subsys*/iopolicy
```

Example output:

```
round-robin
round-robin
```

4. Verify that the ONTAP namespaces are visible to the host:

```
nvme list -v
```

Example output:

```
Subsystem        Subsystem-NQN
Controllers
----------------
--------------------------------------------------------------------------------
------------ -----------------------
nvme-subsys0     nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.unidir_dhcha
p   nvme0, nvme1, nvme2, nvme3


Device   SN                   MN
FR       TxPort Asdress        Subsystem    Namespaces
-------- -------------------- ----------------------------------------
-------- ---------------------------------------------
nvme0    81LGgBUqsI3EAAAAAAAE NetApp ONTAP Controller   FFFFFFFF tcp
traddr=192.168.2.214,trsvcid=4420,host_traddr=192.168.2.14 nvme-subsys0
nvme0n1
nvme1    81LGgBUqsI3EAAAAAAAE NetApp ONTAP Controller   FFFFFFFF tcp
traddr=192.168.2.215,trsvcid=4420,host_traddr=192.168.2.14 nvme-subsys0
nvme0n1
nvme2    81LGgBUqsI3EAAAAAAAE NetApp ONTAP Controller   FFFFFFFF tcp
traddr=192.168.1.214,trsvcid=4420,host_traddr=192.168.1.14 nvme-subsys0
nvme0n1
nvme3    81LGgBUqsI3EAAAAAAAE NetApp ONTAP Controller   FFFFFFFF tcp
traddr=192.168.1.215,trsvcid=4420,host_traddr=192.168.1.14 nvme-subsys0
nvme0n1


Device       Generic      NSID       Usage                 Format
Controllers
------------ ------------ ----------
-------------------------------------------------------------
/dev/nvme0n1 /dev/ng0n1   0x1     1.07  GB /   1.07  GB    4 KiB +  0 B
nvme0, nvme1, nvme2, nvme3
```

5. Verify that the controller state of each path is live and has the correct ANA status:

```
nvme list-subsys /dev/<subsystem_name>
```

NVMe/FC

Example output

```
# nvme list-subsys /dev/nvme1n1
nvme-subsys1 - NQN=nqn.1992-
08.com.netapp:sn.04ba0732530911ea8e8300a098dfdd91:subsystem.nvme_145
_1
\
+- nvme2 fc traddr=nn-0x208100a098dfdd91:pn-
0x208200a098dfdd91,host_traddr=nn-0x200000109b579d5f:pn-
0x100000109b579d5f live optimized
+- nvme3 fc traddr=nn-0x208100a098dfdd91:pn-
0x208500a098dfdd91,host_traddr=nn-0x200000109b579d5e:pn-
0x100000109b579d5e live optimized
+- nvme4 fc traddr=nn-0x208100a098dfdd91:pn-
0x208400a098dfdd91,host_traddr=nn-0x200000109b579d5e:pn-
0x100000109b579d5e live non-optimized
+- nvme6 fc traddr=nn-0x208100a098dfdd91:pn-
0x208300a098dfdd91,host_traddr=nn-0x200000109b579d5f:pn-
0x100000109b579d5f live non-optimized
```

NVMe/TCP

Example output

```
# nvme list-subsys
nvme-subsys0 - NQN=nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.unidir_d
hchap
hostnqn=nqn.2014-08.org.nvmexpress:uuid:e58eca24-faff-11ea-8fee-
3a68dd3b5c5f
iopolicy=round-robin

 +- nvme0 tcp
traddr=192.168.2.214,trsvcid=4420,host_traddr=192.168.2.14 live
 +- nvme1 tcp
traddr=192.168.2.215,trsvcid=4420,host_traddr=192.168.2.14 live
 +- nvme2 tcp
traddr=192.168.1.214,trsvcid=4420,host_traddr=192.168.1.14 live
 +- nvme3 tcp
traddr=192.168.1.215,trsvcid=4420,host_traddr=192.168.1.14 live
```

6. Verify that the NetApp plug-in displays the correct values for each ONTAP namespace device:

Column

```
nvme netapp ontapdevices -o column
```

Example output:

```
Device           Vserver                   Namespace Path
NSID UUID                                   Size
---------------- -------------------------
---------------------------------------------------------------------
---------------------------
/dev/nvme0n1     vs_CLIENT114
/vol/CLIENT114_vol_0_10/CLIENT114_ns10       1    c6586535-da8a-
40fa-8c20-759ea0d69d33   1.07GB
```

JSON

```
nvme netapp ontapdevices -o json
```

Example output:

```
{
"ONTAPdevices":[
{
"Device":"/dev/nvme0n1",
"Vserver":"vs_CLIENT114",
"Namespace_Path":"/vol/CLIENT114_vol_0_10/CLIENT114_ns10",
"NSID":1,
"UUID":"c6586535-da8a-40fa-8c20-759ea0d69d33",
"Size":"1.07GB",
"LBA_Data_Size":4096,
"Namespace_Size":262144
}
]
}
```

Create a persistent discovery controller

Beginning with ONTAP 9.11.1, you can create a persistent discovery controller (PDC) for your SUSE Linux Enterprise Server 15 SP5 host. A PDC is required to automatically detect an NVMe subsystem add or remove scenario and changes to the discovery log page data.

Steps

1. Verify that the discovery log page data is available and can be retrieved through the initiator port and target LIF combination:

```
nvme discover -t <trtype> -w <host-traddr> -a <traddr>
```

Show example output:

```
Discovery Log Number of Records 16, Generation counter 14
=====Discovery Log Entry 0======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  0
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:discovery
traddr:  192.168.1.214
eflags:  explicit discovery connections, duplicate discovery
information sectype: none
=====Discovery Log Entry 1======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  0
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:discovery
traddr:  192.168.1.215
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 2======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  0
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:discovery
traddr:  192.168.2.215
eflags:  explicit discovery connections, duplicate discovery
information sectype: none
=====Discovery Log Entry 3======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  0
```

```
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:discovery
traddr:  192.168.2.214
eflags:  explicit discovery connections, duplicate discovery
information sectype: none
=====Discovery Log Entry 4======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  0
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.unidir_n
one
traddr:  192.168.1.214
eflags:  none
sectype: none
=====Discovery Log Entry 5======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  0
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.unidir_n
one
traddr:  192.168.1.215
eflags:  none
sectype: none
=====Discovery Log Entry 6======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  0
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.unidir_n
one
traddr:  192.168.2.215
eflags:  none
sectype: none
=====Discovery Log Entry 7======
```

```
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  0
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.unidir_n
one
traddr:  192.168.2.214
eflags:  none
sectype: none
=====Discovery Log Entry 8======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  0
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.subsys_C
LIENT114
traddr:  192.168.1.214
eflags:  none
sectype: none
=====Discovery Log Entry 9======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  0
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.subsys_C
LIENT114
traddr:  192.168.1.215
eflags:  none
sectype: none
=====Discovery Log Entry 10======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  0
trsvcid: 4420
subnqn:  nqn.1992-
```

```
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.subsys_C
LIENT114
traddr:  192.168.2.215
eflags:  none
sectype: none
=====Discovery Log Entry 11======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  0
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.subsys_C
LIENT114
traddr:  192.168.2.214
eflags:  none
sectype: none
=====Discovery Log Entry 12======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  0
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.unidir_d
hchap
traddr:  192.168.1.214
eflags:  none
sectype: none
=====Discovery Log Entry 13======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  0
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.unidir_d
hchap
traddr:  192.168.1.215
eflags:  none
sectype: none
=====Discovery Log Entry 14======
trtype:  tcp
```

```
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  0
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.unidir_d
hchap
traddr:  192.168.2.215
eflags:  none
sectype: none
=====Discovery Log Entry 15======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  0
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.unidir_d
hchap
traddr:  192.168.2.214
eflags:  none
sectype: none
```

2. Create a PDC for the discovery subsystem:

```
nvme discover -t <trtype> -w <host-traddr> -a <traddr> -p
```

Example output:

```
nvme discover -t tcp -w 192.168.1.16 -a 192.168.1.116 -p
```

3. From the ONTAP controller, verify that the PDC has been created:

```
vserver nvme show-discovery-controller -instance -vserver vserver_name
```

Example output:

```
vserver nvme show-discovery-controller -instance -vserver vs_nvme175
Vserver Name: vs_CLIENT116 Controller ID: 00C0h
Discovery Subsystem NQN: nqn.1992-
08.com.netapp:sn.48391d66c0a611ecaaa5d039ea165514:discovery Logical
Interface UUID: d23cbb0a-c0a6-11ec-9731-d039ea165abc Logical Interface:
CLIENT116_lif_4a_1
Node: A400-14-124
Host NQN: nqn.2014-08.org.nvmexpress:uuid:12372496-59c4-4d1b-be09-
74362c0c1afc
Transport Protocol: nvme-tcp
Initiator Transport Address: 192.168.1.16
Host Identifier: 59de25be738348f08a79df4bce9573f3 Admin Queue Depth: 32
Header Digest Enabled: false Data Digest Enabled: false
Vserver UUID: 48391d66-c0a6-11ec-aaa5-d039ea165514
```

Set up secure in-band authentication

Beginning with ONTAP 9.12.1, secure in-band authentication is supported over NVMe/TCP and NVMe/FC between your SUSE Linux Enterprise Server 15 SP5 host and your ONTAP controller.

To set up secure authentication, each host or controller must be associated with a `DH-HMAC-CHAP` key, which is a combination of the NQN of the NVMe host or controller and an authentication secret configured by the administrator. To authenticate its peer, an NVMe host or controller must recognize the key associated with the peer.

You can set up secure in-band authentication using the CLI or a config JSON file. If you need to specify different dhchap keys for different subsystems, you must use a config JSON file.

CLI

Steps

1. Obtain the host NQN:

```
cat /etc/nvme/hostnqn
```

2. Generate the dhchap key for the SUSE Linux Enterprise Server 15 SP5 host:

```
nvme gen-dhchap-key -s optional_secret -l key_length {32|48|64} -m
HMAC_function {0|1|2|3} -n host_nqn

•   -s secret key in hexadecimal characters to be used to initialize
the host key
•   -l length of the resulting key in bytes
•   -m HMAC function to use for key transformation
0 = none, 1- SHA-256, 2 = SHA-384, 3=SHA-512
•   -n host NQN to use for key transformation
```

In the following example, a random dhchap key with HMAC set to 3 (SHA-512) is generated.

```
# nvme gen-dhchap-key -m 3 -n nqn.2014-
08.org.nvmexpress:uuid:d3ca725a- ac8d-4d88-b46a-174ac235139b
DHHC-
1:03:J2UJQfj9f0pLnpF/ASDJRTyILKJRr5CougGpGdQSysPrLu6RW1fGl5VSjbeDF1n
1DEh3nVBe19nQ/LxreSBeH/bx/pU=:
```

3. On the ONTAP controller, add the host and specify both dhchap keys:

```
vserver nvme subsystem host add -vserver <svm_name> -subsystem
<subsystem> -host-nqn <host_nqn> -dhchap-host-secret
<authentication_host_secret> -dhchap-controller-secret
<authentication_controller_secret> -dhchap-hash-function {sha-
256|sha-512} -dhchap-group {none|2048-bit|3072-bit|4096-bit|6144-
bit|8192-bit}
```

4. A host supports two types of authentication methods, unidirectional and bidirectional. On the host, connect to the ONTAP controller and specify dhchap keys based on the chosen authentication method:

```
nvme connect -t tcp -w <host-traddr> -a <tr-addr> -n <host_nqn> -S
<authentication_host_secret> -C <authentication_controller_secret>
```

5. Validate the `nvme connect authentication` command by verifying the host and controller dhchap keys:

    a. Verify the host dhchap keys:

    ```
    $cat /sys/class/nvme-subsystem/<nvme-subsysX>/nvme*/dhchap_secret
    ```

    Example output for unidirectional configuration:

    ```
    # cat /sys/class/nvme-subsystem/nvme-subsys1/nvme*/dhchap_secret
    DHHC-
    1:03:je1nQCmjJLUKD62mpYbzlpuw0OIws86NB96uNO/t3jbvhp7fjyR9bIRjOHg8
    wQtye1JCFSMkBQH3pTKGdYR1OV9gx00=:
    DHHC-
    1:03:je1nQCmjJLUKD62mpYbzlpuw0OIws86NB96uNO/t3jbvhp7fjyR9bIRjOHg8
    wQtye1JCFSMkBQH3pTKGdYR1OV9gx00=:
    DHHC-
    1:03:je1nQCmjJLUKD62mpYbzlpuw0OIws86NB96uNO/t3jbvhp7fjyR9bIRjOHg8
    wQtye1JCFSMkBQH3pTKGdYR1OV9gx00=:
    DHHC-
    1:03:je1nQCmjJLUKD62mpYbzlpuw0OIws86NB96uNO/t3jbvhp7fjyR9bIRjOHg8
    wQtye1JCFSMkBQH3pTKGdYR1OV9gx00=:
    ```

    b. Verify the controller dhchap keys:

    ```
    $cat /sys/class/nvme-subsystem/<nvme-
    subsysX>/nvme*/dhchap_ctrl_secret
    ```

    Example output for bidirectional configuration:

```
# cat /sys/class/nvme-subsystem/nvme-
subsys6/nvme*/dhchap_ctrl_secret
DHHC-
1:03:WorVEV83eYO53kV4Iel5OpphbX5LAphO3F8fgH3913tlrkSGDBJTt3crXeTU
B8fCwGbPsEyz6CXxdQJi6kbn4IzmkFU=:
DHHC-
1:03:WorVEV83eYO53kV4Iel5OpphbX5LAphO3F8fgH3913tlrkSGDBJTt3crXeTU
B8fCwGbPsEyz6CXxdQJi6kbn4IzmkFU=:
DHHC-
1:03:WorVEV83eYO53kV4Iel5OpphbX5LAphO3F8fgH3913tlrkSGDBJTt3crXeTU
B8fCwGbPsEyz6CXxdQJi6kbn4IzmkFU=:
DHHC-
1:03:WorVEV83eYO53kV4Iel5OpphbX5LAphO3F8fgH3913tlrkSGDBJTt3crXeTU
B8fCwGbPsEyz6CXxdQJi6kbn4IzmkFU=:
```

JSON file

You can use the `/etc/nvme/config.json` file with the `nvme connect-all` command when multiple NVMe subsystems are available on the ONTAP controller configuration.

You can generate the JSON file using `-o` option. Refer to the NVMe connect-all man pages for more syntax options.

Steps

1. Configure the JSON file:

```
# cat /etc/nvme/config.json
[
 {
    "hostnqn":"nqn.2014-08.org.nvmexpress:uuid:12372496-59c4-4d1b-
be09-74362c0c1afc",
    "hostid":"3ae10b42-21af-48ce-a40b-cfb5bad81839",
    "dhchap_key":"DHHC-
1:03:Cu3ZZfIz1WMlqZFnCMqpAgn/T6EVOcIFHez215U+Pow8jTgBF2UbNk3DK4wfk2E
ptWpna1rpwG5CndpOgxpRxh9m41w=:"
 },

 {
    "hostnqn":"nqn.2014-08.org.nvmexpress:uuid:12372496-59c4-4d1b-
be09-74362c0c1afc",
    "subsystems":[
        {
            "nqn":"nqn.1992-
08.com.netapp:sn.48391d66c0a611ecaaa5d039ea165514:subsystem.subsys_C
LIENT116",
            "ports":[
```

```
                {
                        "transport":"tcp",
                        "traddr":"192.168.1.117",
                        "host_traddr":"192.168.1.16",
                        "trsvcid":"4420",
                        "dhchap_ctrl_key":"DHHC-
1:01:0h58bcT/uu0rCpGsDYU6ZHZvRuVqsYKuBRS0Nu0VPx5HEwaZ:"
                },
                {
                        "transport":"tcp",
                        "traddr":"192.168.1.116",
                        "host_traddr":"192.168.1.16",
                        "trsvcid":"4420",
                        "dhchap_ctrl_key":"DHHC-
1:01:0h58bcT/uu0rCpGsDYU6ZHZvRuVqsYKuBRS0Nu0VPx5HEwaZ:"
                },
                {
                        "transport":"tcp",
                        "traddr":"192.168.2.117",
                        "host_traddr":"192.168.2.16",
                        "trsvcid":"4420",
                        "dhchap_ctrl_key":"DHHC-
1:01:0h58bcT/uu0rCpGsDYU6ZHZvRuVqsYKuBRS0Nu0VPx5HEwaZ:"
                },
                {
                        "transport":"tcp",
                        "traddr":"192.168.2.116",
                        "host_traddr":"192.168.2.16",
                        "trsvcid":"4420",
                        "dhchap_ctrl_key":"DHHC-
1:01:0h58bcT/uu0rCpGsDYU6ZHZvRuVqsYKuBRS0Nu0VPx5HEwaZ:"
                }
            ]
        }
    ]
 }
]

[NOTE]
In the preceding example, `dhchap_key` corresponds to
`dhchap_secret` and `dhchap_ctrl_key` corresponds to
`dhchap_ctrl_secret`.
```

2. Connect to the ONTAP controller using the config JSON file:

```
nvme connect-all -J /etc/nvme/config.json
```

Example output:

```
traddr=192.168.2.116 is already connected
traddr=192.168.1.116 is already connected
traddr=192.168.2.117 is already connected
traddr=192.168.1.117 is already connected
traddr=192.168.2.117 is already connected
traddr=192.168.1.117 is already connected
traddr=192.168.2.116 is already connected
traddr=192.168.1.116 is already connected
traddr=192.168.2.116 is already connected
traddr=192.168.1.116 is already connected
traddr=192.168.2.117 is already connected
traddr=192.168.1.117 is already connected
```

3. Verify that the dhchap secrets have been enabled for the respective controllers for each subsystem:
   a. Verify the host dhchap keys:

   ```
   # cat /sys/class/nvme-subsystem/nvme-subsys0/nvme0/dhchap_secret
   ```

   Example output:

   ```
   DHHC-1:01:NunEWY7AZlXqxITGheByarwZdQvU4ebZg9HOjIr6nOHEkxJg:
   ```

   b. Verify the controller dhchap keys:

   ```
   # cat /sys/class/nvme-subsystem/nvme-
   subsys0/nvme0/dhchap_ctrl_secret
   ```

   Example output:

   ```
   DHHC-
   1:03:2YJinsxa2v3+m8qqCiTnmgBZoH6mIT6G/6f0aGO8viVZB4VLNLH4z8CvK7pV
   YxN6S5fOAtaU3DNi12rieRMfdbg3704=:
   ```

Known issues

There are no known issues for the SUSE Linux Enterprise Server 15 SP5 with ONTAP release.

NVMe-oF host configuration for SUSE Linux Enterprise Server 15 SP4 with ONTAP

NVMe over Fabrics (NVMe-oF), including NVMe over Fibre Channel (NVMe/FC) and other transports, is supported with SUSE Linux Enterprise Server (SLES) 15 SP4 with Asymmetric Namespace Access (ANA). In NVMe-oF environments, ANA is the equivalent of ALUA multipathing in iSCSI and FCP environments and is implemented with in-kernel NVMe multipath.

The following support is available for the NVMe-oF host configuration for SUSE Linux Enterprise Server 15 SP4 with ONTAP:

- Both NVMe and SCSI traffic can be run on the same host. Therefore, for SCSI LUNs, you can configure dm-multipath for SCSI mpath devices, whereas you might use NVMe multipath to configure NVMe-oF namespace devices on the host.
- Support for NVMe over TCP (NVMe/TCP) in addition to NVMe/FC. The NetApp plug-in in the native nvme-cli package displays ONTAP details for both NVMe/FC and NVMe/TCP namespaces.

For additional details on supported configurations, see the Interoperability Matrix Tool.

Features

- Support for NVMe secure, in-band authentication
- Support for persistent discovery controllers (PDCs) using a unique discovery NQN

Known limitations

- SAN booting using the NVMe-oF protocol is currently not supported.
- There's no sanlun support for NVMe-oF. Therefore, the host utility support isn't available for NVMe-oF on an SUSE Linux Enterprise Server 15 SP5 host. You can rely on the NetApp plug-in included in the native nvme-cli package for all NVMe-oF transports.

Configure NVMe/FC

You can configure NVMe/FC for Broadcom/Emulex FC adapters or Marvell/Qlogic FC adapters.

Broadcom/Emulex

Steps

1. Verify that you are using the recommended adapter model:

   ```
   cat /sys/class/scsi_host/host*/modelname
   ```

   Example output:

   ```
   LPe32002 M2
   LPe32002-M2
   ```

2. Verify the adapter model description:

   ```
   cat /sys/class/scsi_host/host*/modeldesc
   ```

   Example output:

   ```
   Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
   Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
   ```

3. Verify that you are using the recommended Emulex host bus adapter (HBA) firmware versions:

   ```
   cat /sys/class/scsi_host/host*/fwrev
   ```

   Example output:

   ```
   12.8.351.47, sli-4:2:c
   12.8.351.47, sli-4:2:c
   ```

4. Verify that you are using the recommended LPFC driver version:

   ```
   cat /sys/module/lpfc/version
   ```

   Example output:

   ```
   0:14.2.0.6
   ```

5. Verify that you can view your initiator ports:

```
cat /sys/class/fc_host/host*/port_name
```

Example output:

```
0x100000109b579d5e
0x100000109b579d5f
```

6. Verify that your initiator ports are online:

```
cat /sys/class/fc_host/host*/port_state
```

Example output:

```
Online
Online
```

7. Verify that the NVMe/FC initiator ports are enabled and that the target ports are visible:

```
cat /sys/class/scsi_host/host*/nvme_info
```

Example output:

In the following example, one initiator port is enabled and connected with two target LIFs.

```
NVME Initiator Enabled
XRI Dist lpfc0 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc0 WWPN x100000109b579d5e WWNN x200000109b579d5e DID
x011c00 ONLINE
NVME RPORT WWPN x208400a098dfdd91 WWNN x208100a098dfdd91 DID x011503
TARGET DISCSRVC ONLINE
NVME RPORT WWPN x208500a098dfdd91 WWNN x208100a098dfdd91 DID x010003
TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 0000000e49 Cmpl 0000000e49 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000003ceb594f Issue 000000003ce65dbe OutIO
fffffffffffb046f
abort 00000bd2 noxri 00000000 nondlp 00000000 qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 000014f4 Err 00012abd

NVME Initiator Enabled
XRI Dist lpfc1 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc1 WWPN x100000109b579d5f WWNN x200000109b579d5f DID
x011b00 ONLINE
NVME RPORT WWPN x208300a098dfdd91 WWNN x208100a098dfdd91 DID x010c03
TARGET DISCSRVC ONLINE
NVME RPORT WWPN x208200a098dfdd91 WWNN x208100a098dfdd91 DID x012a03
TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 0000000e50 Cmpl 0000000e50 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000003c9859ca Issue 000000003c93515e OutIO
fffffffffffaf794
abort 00000b73 noxri 00000000 nondlp 00000000 qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 0000159d Err 000135c3
```

8. Reboot the host.

Marvell/QLogic

The native inbox qla2xxx driver included in the SUSE Linux Enterprise Server 15 SP4 kernel has the latest fixes. These fixes are essential for ONTAP support.

Steps

1. Verify that you are running the supported adapter driver and firmware versions:

```
cat /sys/class/fc_host/host*/symbolic_name
```

Example output:

```
QLE2742 FW:v9.08.02 DVR:v10.02.07.800-k QLE2742 FW:v9.08.02
DVR:v10.02.07.800-k
```

2. Verify that the `ql2xnvmeenable` parameter is set to 1:

```
cat /sys/module/qla2xxx/parameters/ql2xnvmeenable
1
```

Enable 1MB I/O size (Optional)

ONTAP reports an MDTS (Max Data Transfer Size) of 8 in the Identify Controller data. This means the maximum I/O request size can be up to 1MB. To issue I/O requests of size 1MB for a Broadcom NVMe/FC host, you should increase the `lpfc` value of the `lpfc_sg_seg_cnt` parameter to 256 from the default value of 64.

> These steps don't apply to Qlogic NVMe/FC hosts.

Steps

1. Set the `lpfc_sg_seg_cnt` parameter to 256:

```
cat /etc/modprobe.d/lpfc.conf
```

You should see an output similar to the following example:

```
options lpfc lpfc_sg_seg_cnt=256
```

2. Run the `dracut -f` command, and reboot the host.
3. Verify that the value for `lpfc_sg_seg_cnt` is 256:

```
cat /sys/module/lpfc/parameters/lpfc_sg_seg_cnt
```

Enable NVMe services

There are two NVMe/FC boot services included in the `nvme-cli` package, however, *only* `nvmefc-boot-connections.service` is enabled to start during system boot; `nvmf-autoconnect.service` is not enabled. Therefore, you need to manually enable `nvmf-autoconnect.service` to start during system boot.

Steps

1. Enable `nvmf-autoconnect.service`:

```
# systemctl enable nvmf-autoconnect.service
Created symlink /etc/systemd/system/default.target.wants/nvmf-
autoconnect.service → /usr/lib/systemd/system/nvmf-autoconnect.service.
```

2. Reboot the host.
3. Verify that `nvmf-autoconnect.service` and `nvmefc-boot-connections.service` are running after the system boot:

   Example output:

```
# systemctl status nvmf-autoconnect.service
   nvmf-autoconnect.service - Connect NVMe-oF subsystems automatically
during boot
     Loaded: loaded (/usr/lib/systemd/system/nvmf-autoconnect.service;
enabled; vendor preset: disabled)
     Active: inactive (dead) since Thu 2023-05-25 14:55:00 IST; 11min
ago
    Process: 2108 ExecStartPre=/sbin/modprobe nvme-fabrics (code=exited,
status=0/SUCCESS)
    Process: 2114 ExecStart=/usr/sbin/nvme connect-all (code=exited,
status=0/SUCCESS)
   Main PID: 2114 (code=exited, status=0/SUCCESS)

   systemd[1]: Starting Connect NVMe-oF subsystems automatically during
boot...
   nvme[2114]: traddr=nn-0x201700a098fd4ca6:pn-0x201800a098fd4ca6 is
already connected
   systemd[1]: nvmf-autoconnect.service: Deactivated successfully.
   systemd[1]: Finished Connect NVMe-oF subsystems automatically during
boot.

# systemctl status nvmefc-boot-connections.service
nvmefc-boot-connections.service - Auto-connect to subsystems on FC-NVME
devices found during boot
   Loaded: loaded (/usr/lib/systemd/system/nvmefc-boot-
connections.service; enabled; vendor preset: enabled)
   Active: inactive (dead) since Thu 2023-05-25 14:55:00 IST; 11min ago
 Main PID: 1647 (code=exited, status=0/SUCCESS)

systemd[1]: Starting Auto-connect to subsystems on FC-NVME devices found
during boot...
systemd[1]: nvmefc-boot-connections.service: Succeeded.
systemd[1]: Finished Auto-connect to subsystems on FC-NVME devices found
during boot.
```

Configure NVMe/TCP

You can use the following procedure to configure NVMe/TCP.

Steps

1. Verify that the initiator port can fetch the discovery log page data across the supported NVMe/TCP LIFs:

```
nvme discover -t tcp -w <host-traddr> -a <traddr>
```

Example output:

```
# nvme discover -t tcp -w 192.168.1.4 -a 192.168.1.31

Discovery Log Number of Records 8, Generation counter 18
=====Discovery Log Entry 0====== trtype: tcp
adrfam: ipv4
subtype: current discovery subsystem treq: not specified
portid: 0
trsvcid: 8009 subnqn: nqn.1992-
08.com.netapp:sn.48391d66c0a611ecaaa5d039ea165514:discovery traddr:
192.168.2.117
eflags: explicit discovery connections, duplicate discovery information
sectype: none
=====Discovery Log Entry 1====== trtype: tcp
adrfam: ipv4
subtype: current discovery subsystem treq: not specified
portid: 1
trsvcid: 8009 subnqn: nqn.1992-
08.com.netapp:sn.48391d66c0a611ecaaa5d039ea165514:discovery traddr:
192.168.1.117
eflags: explicit discovery connections, duplicate discovery information
sectype: none
=====Discovery Log Entry 2====== trtype: tcp
adrfam: ipv4
subtype: current discovery subsystem treq: not specified
portid: 2
trsvcid: 8009 subnqn: nqn.1992-
08.com.netapp:sn.48391d66c0a611ecaaa5d039ea165514:discovery traddr:
192.168.2.116
eflags: explicit discovery connections, duplicate discovery information
sectype: none
=====Discovery Log Entry 3====== trtype: tcp
adrfam: ipv4
subtype: current discovery subsystem treq: not specified
portid: 3
trsvcid: 8009 subnqn: nqn.1992-
08.com.netapp:sn.48391d66c0a611ecaaa5d039ea165514:discovery traddr:
192.168.1.116
eflags: explicit discovery connections, duplicate discovery information
sectype: none
=====Discovery Log Entry 4====== trtype: tcp
adrfam: ipv4
subtype: nvme subsystem treq: not specified portid: 0
trsvcid: 4420 subnqn: nqn.1992-
08.com.netapp:sn.48391d66c0a611ecaaa5d039ea165514:subsystem.subsys_CLIEN
```

```
T116
traddr: 192.168.2.117 eflags: not specified sectype: none
=====Discovery Log Entry 5====== trtype: tcp
adrfam: ipv4
subtype: nvme subsystem treq: not specified portid: 1
trsvcid: 4420 subnqn: nqn.1992-
08.com.netapp:sn.48391d66c0a611ecaaa5d039ea165514:subsystem.subsys_CLIEN
T116
traddr: 192.168.1.117 eflags: not specified sectype: none
=====Discovery Log Entry 6====== trtype: tcp
adrfam: ipv4
subtype: nvme subsystem treq: not specified portid: 2
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.48391d66c0a611ecaaa5d039ea165514:subsystem.subsys_CLIEN
T116
traddr: 192.168.2.116 eflags: not specified sectype: none
=====Discovery Log Entry 7====== trtype: tcp
adrfam: ipv4
subtype: nvme subsystem treq: not specified portid: 3
trsvcid: 4420 subnqn: nqn.1992-
08.com.netapp:sn.48391d66c0a611ecaaa5d039ea165514:subsystem.subsys_CLIEN
T116
traddr: 192.168.1.116 eflags: not specified sectype: none
```

2. Verify that all other NVMe/TCP initiator-target LIF combinations can successfully fetch discovery log page data:

```
nvme discover -t tcp -w <host-traddr> -a <traddr>
```

Example output:

```
# nvme discover -t tcp -w 192.168.1.4 -a 192.168.1.32
# nvme discover -t tcp -w 192.168.2.5 -a 192.168.2.36
# nvme discover -t tcp -w 192.168.2.5 -a 192.168.2.37
```

3. Run the `nvme connect-all` command across all the supported NVMe/TCP initiator-target LIFs across the nodes:

```
nvme connect-all -t tcp -w host-traddr -a traddr -l
<ctrl_loss_timeout_in_seconds>
```

Example output:

```
# nvme connect-all -t tcp -w 192.168.1.4 -a 192.168.1.31 -l -1
# nvme connect-all -t tcp -w 192.168.1.4 -a 192.168.1.32 -l -1
# nvme connect-all -t tcp -w 192.168.2.5 -a 192.168.1.36 -l -1
# nvme connect-all -t tcp -w 192.168.2.5 -a 192.168.1.37 -l -1
```

NetApp recommends setting the `ctrl-loss-tmo` option to `-1` so that the NVMe/TCP initiator attempts to reconnect indefinitely in the event of a path loss.

Validate NVMe-oF

You can use the following procedure to validate NVMe-oF.

Steps

1. Verify that in-kernel NVMe multipath is enabled:

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

2. Verify that the host has the correct controller model for the ONTAP NVMe namespaces:

```
cat /sys/class/nvme-subsystem/nvme-subsys*/model
```

Example output:

```
NetApp ONTAP Controller
NetApp ONTAP Controller
```

3. Verify the NVMe I/O policy for the respective ONTAP NVMe I/O controller:

```
cat /sys/class/nvme-subsystem/nvme-subsys*/iopolicy
```

Example output:

```
round-robin
round-robin
```

4. Verify that the ONTAP namespaces are visible to the host:

```
nvme list -v
```

Example output:

```
Subsystem        Subsystem-NQN
Controllers
----------------
------------------------------------------------------------------------
------------ -----------------------
nvme-subsys0     nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.unidir_dhcha
p    nvme0, nvme1, nvme2, nvme3


Device   SN                   MN
FR       TxPort Asdress        Subsystem    Namespaces
-------- -------------------- ----------------------------------------
-------- ---------------------------------------------
nvme0    81LGgBUqsI3EAAAAAAAE NetApp ONTAP Controller   FFFFFFFF tcp
traddr=192.168.2.214,trsvcid=4420,host_traddr=192.168.2.14 nvme-subsys0
nvme0n1
nvme1    81LGgBUqsI3EAAAAAAAE NetApp ONTAP Controller   FFFFFFFF tcp
traddr=192.168.2.215,trsvcid=4420,host_traddr=192.168.2.14 nvme-subsys0
nvme0n1
nvme2    81LGgBUqsI3EAAAAAAAE NetApp ONTAP Controller   FFFFFFFF tcp
traddr=192.168.1.214,trsvcid=4420,host_traddr=192.168.1.14 nvme-subsys0
nvme0n1
nvme3    81LGgBUqsI3EAAAAAAAE NetApp ONTAP Controller   FFFFFFFF tcp
traddr=192.168.1.215,trsvcid=4420,host_traddr=192.168.1.14 nvme-subsys0
nvme0n1


Device       Generic      NSID       Usage                 Format
Controllers
------------ ------------ ----------
-------------------------------------------------------------
/dev/nvme0n1 /dev/ng0n1   0x1     1.07  GB /   1.07  GB    4 KiB +  0 B
nvme0, nvme1, nvme2, nvme3
```

5. Verify that the controller state of each path is live and has the correct ANA status:

```
nvme list-subsys /dev/<subsystem_name>
```

NVMe/FC

```
# nvme list-subsys /dev/nvme1n1
nvme-subsys1 - NQN=nqn.1992-
08.com.netapp:sn.04ba0732530911ea8e8300a098dfdd91:subsystem.nvme_145
_1
\
+- nvme2 fc traddr=nn-0x208100a098dfdd91:pn-
0x208200a098dfdd91,host_traddr=nn-0x200000109b579d5f:pn-
0x100000109b579d5f live optimized
+- nvme3 fc traddr=nn-0x208100a098dfdd91:pn-
0x208500a098dfdd91,host_traddr=nn-0x200000109b579d5e:pn-
0x100000109b579d5e live optimized
+- nvme4 fc traddr=nn-0x208100a098dfdd91:pn-
0x208400a098dfdd91,host_traddr=nn-0x200000109b579d5e:pn-
0x100000109b579d5e live non-optimized
+- nvme6 fc traddr=nn-0x208100a098dfdd91:pn-
0x208300a098dfdd91,host_traddr=nn-0x200000109b579d5f:pn-
0x100000109b579d5f live non-optimized
```

NVMe/TCP

```
# nvme list-subsys
nvme-subsys0 - NQN=nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.unidir_d
hchap
hostnqn=nqn.2014-08.org.nvmexpress:uuid:e58eca24-faff-11ea-8fee-
3a68dd3b5c5f
iopolicy=round-robin

 +- nvme0 tcp
traddr=192.168.2.214,trsvcid=4420,host_traddr=192.168.2.14 live
 +- nvme1 tcp
traddr=192.168.2.215,trsvcid=4420,host_traddr=192.168.2.14 live
 +- nvme2 tcp
traddr=192.168.1.214,trsvcid=4420,host_traddr=192.168.1.14 live
 +- nvme3 tcp
traddr=192.168.1.215,trsvcid=4420,host_traddr=192.168.1.14 live
```

6. Verify that the NetApp plug-in displays the correct values for each ONTAP namespace device:

Column

```
nvme netapp ontapdevices -o column
```

Example output:

```
Device           Vserver                   Namespace Path
NSID UUID                                   Size
---------------- -------------------------
--------------------------------------------------------------------
---------------------------
/dev/nvme0n1     vs_CLIENT114
/vol/CLIENT114_vol_0_10/CLIENT114_ns10       1    c6586535-da8a-
40fa-8c20-759ea0d69d33   1.07GB
```

JSON

```
nvme netapp ontapdevices -o json
```

Example output:


```
{
  "ONTAPdevices":[
    {
      "Device":"/dev/nvme0n1",
      "Vserver":"vs_CLIENT114",
      "Namespace_Path":"/vol/CLIENT114_vol_0_10/CLIENT114_ns10",
      "NSID":1,
      "UUID":"c6586535-da8a-40fa-8c20-759ea0d69d33",
      "Size":"1.07GB",
      "LBA_Data_Size":4096,
      "Namespace_Size":262144
    }
  ]
}
```


Create a persistent discovery controller

Beginning with ONTAP 9.11.1, you can create a persistent discovery controller (PDC) for your SUSE Linux Enterprise Server 15 SP4 host. A PDC is required to automatically detect an NVMe subsystem add or remove scenario and changes to the discovery log page data.

Steps

1. Verify that the discovery log page data is available and can be retrieved through the initiator port and target LIF combination:

```
nvme discover -t <trtype> -w <host-traddr> -a <traddr>
```

Show example output:

```
Discovery Log Number of Records 16, Generation counter 14
=====Discovery Log Entry 0======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  0
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:discovery
traddr:  192.168.1.214
eflags:  explicit discovery connections, duplicate discovery
information sectype: none
=====Discovery Log Entry 1======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  0
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:discovery
traddr:  192.168.1.215
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 2======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  0
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:discovery
traddr:  192.168.2.215
eflags:  explicit discovery connections, duplicate discovery
information sectype: none
=====Discovery Log Entry 3======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  0
```

```
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:discovery
traddr:  192.168.2.214
eflags:  explicit discovery connections, duplicate discovery
information sectype: none
=====Discovery Log Entry 4======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  0
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.unidir_n
one
traddr:  192.168.1.214
eflags:  none
sectype: none
=====Discovery Log Entry 5======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  0
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.unidir_n
one
traddr:  192.168.1.215
eflags:  none
sectype: none
=====Discovery Log Entry 6======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  0
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.unidir_n
one
traddr:  192.168.2.215
eflags:  none
sectype: none
=====Discovery Log Entry 7======
```

```
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  0
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.unidir_n
one
traddr:  192.168.2.214
eflags:  none
sectype: none
=====Discovery Log Entry 8======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  0
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.subsys_C
LIENT114
traddr:  192.168.1.214
eflags:  none
sectype: none
=====Discovery Log Entry 9======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  0
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.subsys_C
LIENT114
traddr:  192.168.1.215
eflags:  none
sectype: none
=====Discovery Log Entry 10======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  0
trsvcid: 4420
subnqn:  nqn.1992-
```

```
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.subsys_C
LIENT114
traddr:  192.168.2.215
eflags:  none
sectype: none
=====Discovery Log Entry 11======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  0
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.subsys_C
LIENT114
traddr:  192.168.2.214
eflags:  none
sectype: none
=====Discovery Log Entry 12======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  0
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.unidir_d
hchap
traddr:  192.168.1.214
eflags:  none
sectype: none
=====Discovery Log Entry 13======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  0
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.unidir_d
hchap
traddr:  192.168.1.215
eflags:  none
sectype: none
=====Discovery Log Entry 14======
trtype:  tcp
```

```
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  0
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.unidir_d
hchap
traddr:  192.168.2.215
eflags:  none
sectype: none
=====Discovery Log Entry 15======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  0
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.unidir_d
hchap
traddr:  192.168.2.214
eflags:  none
sectype: none
```

2. Create a PDC for the discovery subsystem:

```
nvme discover -t <trtype> -w <host-traddr> -a <traddr> -p
```

Example output:

```
nvme discover -t tcp -w 192.168.1.16 -a 192.168.1.116 -p
```

3. From the ONTAP controller, verify that the PDC has been created:

```
vserver nvme show-discovery-controller -instance -vserver vserver_name
```

Example output:

```
vserver nvme show-discovery-controller -instance -vserver vs_nvme175
Vserver Name: vs_CLIENT116 Controller ID: 00C0h
Discovery Subsystem NQN: nqn.1992-
08.com.netapp:sn.48391d66c0a611ecaaa5d039ea165514:discovery Logical
Interface UUID: d23cbb0a-c0a6-11ec-9731-d039ea165abc Logical Interface:
CLIENT116_lif_4a_1
Node: A400-14-124
Host NQN: nqn.2014-08.org.nvmexpress:uuid:12372496-59c4-4d1b-be09-
74362c0c1afc
Transport Protocol: nvme-tcp
Initiator Transport Address: 192.168.1.16
Host Identifier: 59de25be738348f08a79df4bce9573f3 Admin Queue Depth: 32
Header Digest Enabled: false Data Digest Enabled: false
Vserver UUID: 48391d66-c0a6-11ec-aaa5-d039ea165514
```

Set up secure in-band authentication

Beginning with ONTAP 9.12.1, secure, in-band authentication is supported over NVMe/TCP and NVMe/FC between your SUSE Linux Enterprise Server 15 SP4 host and your ONTAP controller.

To set up secure authentication, each host or controller must be associated with a `DH-HMAC-CHAP` key, which is a combination of the NQN of the NVMe host or controller and an authentication secret configured by the administrator. To authenticate its peer, an NVMe host or controller must recognize the key associated with the peer.

You can set up secure in-band authentication using the CLI or a config JSON file. If you need to specify different dhchap keys for different subsystems, you must use a config JSON file.

CLI

Steps

1. Obtain the host NQN:

```
cat /etc/nvme/hostnqn
```

2. Generate the dhchap key for the SUSE Linux Enterprise Server 15 SP4 host:

```
nvme gen-dhchap-key -s optional_secret -l key_length {32|48|64} -m
HMAC_function {0|1|2|3} -n host_nqn

•   -s secret key in hexadecimal characters to be used to initialize
the host key
•   -l length of the resulting key in bytes
•   -m HMAC function to use for key transformation
0 = none, 1- SHA-256, 2 = SHA-384, 3=SHA-512
•   -n host NQN to use for key transformation
```

+
In the following example, a random dhchap key with HMAC set to 3 (SHA-512) is generated.

```
# nvme gen-dhchap-key -m 3 -n nqn.2014-08.org.nvmexpress:uuid:d3ca725a-
ac8d-4d88-b46a-174ac235139b
DHHC-
1:03:J2UJQfj9f0pLnpF/ASDJRTyILKJRr5CougGpGdQSysPrLu6RW1fGl5VSjbeDF1n1DE
h3nVBe19nQ/LxreSBeH/bx/pU=:
```

1. On the ONTAP controller, add the host and specify both dhchap keys:

```
vserver nvme subsystem host add -vserver <svm_name> -subsystem
<subsystem> -host-nqn <host_nqn> -dhchap-host-secret
<authentication_host_secret> -dhchap-controller-secret
<authentication_controller_secret> -dhchap-hash-function {sha-
256|sha-512} -dhchap-group {none|2048-bit|3072-bit|4096-bit|6144-
bit|8192-bit}
```

2. A host supports two types of authentication methods, unidirectional and bidirectional. On the host, connect to the ONTAP controller and specify dhchap keys based on the chosen authentication method:

```
nvme connect -t tcp -w <host-traddr> -a <tr-addr> -n <host_nqn> -S
<authentication_host_secret> -C <authentication_controller_secret>
```

3. Validate the `nvme connect authentication` command by verifying the host and controller dhchap keys:

   a. Verify the host dhchap keys:

   ```
   $cat /sys/class/nvme-subsystem/<nvme-subsysX>/nvme*/dhchap_secret
   ```

   Example output for unidirectional configuration:

   ```
   SR650-14-114:~ # cat /sys/class/nvme-subsystem/nvme-
   subsys1/nvme*/dhchap_secret
   DHHC-
   1:03:je1nQCmjJLUKD62mpYbzlpuw0OIws86NB96uNO/t3jbvhp7fjyR9bIRjOHg8
   wQtye1JCFSMkBQH3pTKGdYR1OV9gx00=:
   DHHC-
   1:03:je1nQCmjJLUKD62mpYbzlpuw0OIws86NB96uNO/t3jbvhp7fjyR9bIRjOHg8
   wQtye1JCFSMkBQH3pTKGdYR1OV9gx00=:
   DHHC-
   1:03:je1nQCmjJLUKD62mpYbzlpuw0OIws86NB96uNO/t3jbvhp7fjyR9bIRjOHg8
   wQtye1JCFSMkBQH3pTKGdYR1OV9gx00=:
   DHHC-
   1:03:je1nQCmjJLUKD62mpYbzlpuw0OIws86NB96uNO/t3jbvhp7fjyR9bIRjOHg8
   wQtye1JCFSMkBQH3pTKGdYR1OV9gx00=:
   ```

   b. Verify the controller dhchap keys:

   ```
   $cat /sys/class/nvme-subsystem/<nvme-
   subsysX>/nvme*/dhchap_ctrl_secret
   ```

   Example output for bidirectional configuration:

```
SR650-14-114:~ # cat /sys/class/nvme-subsystem/nvme-
subsys6/nvme*/dhchap_ctrl_secret
DHHC-
1:03:WorVEV83eYO53kV4Iel5OpphbX5LAphO3F8fgH3913tlrkSGDBJTt3crXeTU
B8fCwGbPsEyz6CXxdQJi6kbn4IzmkFU=:
DHHC-
1:03:WorVEV83eYO53kV4Iel5OpphbX5LAphO3F8fgH3913tlrkSGDBJTt3crXeTU
B8fCwGbPsEyz6CXxdQJi6kbn4IzmkFU=:
DHHC-
1:03:WorVEV83eYO53kV4Iel5OpphbX5LAphO3F8fgH3913tlrkSGDBJTt3crXeTU
B8fCwGbPsEyz6CXxdQJi6kbn4IzmkFU=:
DHHC-
1:03:WorVEV83eYO53kV4Iel5OpphbX5LAphO3F8fgH3913tlrkSGDBJTt3crXeTU
B8fCwGbPsEyz6CXxdQJi6kbn4IzmkFU=:
```

JSON file

You can use the `/etc/nvme/config.json` file with the `nvme connect-all` command when multiple NVMe subsystems are available on the ONTAP controller configuration.

You can generate the JSON file using `-o` option. Refer to the NVMe connect-all man pages for more syntax options.

Steps

1. Configure the JSON file:

```
# cat /etc/nvme/config.json
[
 {
    "hostnqn":"nqn.2014-08.org.nvmexpress:uuid:12372496-59c4-4d1b-
be09-74362c0c1afc",
    "hostid":"3ae10b42-21af-48ce-a40b-cfb5bad81839",
    "dhchap_key":"DHHC-
1:03:Cu3ZZfIz1WMlqZFnCMqpAgn/T6EVOcIFHez215U+Pow8jTgBF2UbNk3DK4wfk2E
ptWpna1rpwG5CndpOgxpRxh9m41w=:"
 },

 {
    "hostnqn":"nqn.2014-08.org.nvmexpress:uuid:12372496-59c4-4d1b-
be09-74362c0c1afc",
    "subsystems":[
        {
            "nqn":"nqn.1992-
08.com.netapp:sn.48391d66c0a611ecaaa5d039ea165514:subsystem.subsys_C
LIENT116",
            "ports":[
```

```
                {
                     "transport":"tcp",
                     "traddr":"192.168.1.117",
                     "host_traddr":"192.168.1.16",
                     "trsvcid":"4420",
                     "dhchap_ctrl_key":"DHHC-
1:01:0h58bcT/uu0rCpGsDYU6ZHZvRuVqsYKuBRS0Nu0VPx5HEwaZ:"
                },
                {
                     "transport":"tcp",
                     "traddr":"192.168.1.116",
                     "host_traddr":"192.168.1.16",
                     "trsvcid":"4420",
                     "dhchap_ctrl_key":"DHHC-
1:01:0h58bcT/uu0rCpGsDYU6ZHZvRuVqsYKuBRS0Nu0VPx5HEwaZ:"
                },
                {
                     "transport":"tcp",
                     "traddr":"192.168.2.117",
                     "host_traddr":"192.168.2.16",
                     "trsvcid":"4420",
                     "dhchap_ctrl_key":"DHHC-
1:01:0h58bcT/uu0rCpGsDYU6ZHZvRuVqsYKuBRS0Nu0VPx5HEwaZ:"
                },
                {
                     "transport":"tcp",
                     "traddr":"192.168.2.116",
                     "host_traddr":"192.168.2.16",
                     "trsvcid":"4420",
                     "dhchap_ctrl_key":"DHHC-
1:01:0h58bcT/uu0rCpGsDYU6ZHZvRuVqsYKuBRS0Nu0VPx5HEwaZ:"
                }
            ]
        }
    ]
 }
]

[NOTE]
In the preceding example, `dhchap_key` corresponds to
`dhchap_secret` and `dhchap_ctrl_key` corresponds to
`dhchap_ctrl_secret`.
```

2. Connect to the ONTAP controller using the config JSON file:

```
nvme connect-all -J /etc/nvme/config.json
```

Example output:

```
traddr=192.168.2.116 is already connected
traddr=192.168.1.116 is already connected
traddr=192.168.2.117 is already connected
traddr=192.168.1.117 is already connected
traddr=192.168.2.117 is already connected
traddr=192.168.1.117 is already connected
traddr=192.168.2.116 is already connected
traddr=192.168.1.116 is already connected
traddr=192.168.2.116 is already connected
traddr=192.168.1.116 is already connected
traddr=192.168.2.117 is already connected
traddr=192.168.1.117 is already connected
```

3. Verify that the dhchap secrets have been enabled for the respective controllers for each subsystem:
   a. Verify the host dhchap keys:

      ```
      # cat /sys/class/nvme-subsystem/nvme-subsys0/nvme0/dhchap_secret
      ```

      Example output:

      ```
      DHHC-1:01:NunEWY7AZlXqxITGheByarwZdQvU4ebZg9HOjIr6nOHEkxJg:
      ```

   b. Verify the controller dhchap keys:

      ```
      # cat /sys/class/nvme-subsystem/nvme-
      subsys0/nvme0/dhchap_ctrl_secret
      ```

      Example output:

      ```
      DHHC-
      1:03:2YJinsxa2v3+m8qqCiTnmgBZoH6mIT6G/6f0aGO8viVZB4VLNLH4z8CvK7pV
      YxN6S5fOAtaU3DNi12rieRMfdbg3704=:
      ```

Known issues

There are no known issues for the SUSE Linux Enterprise Server 15 SP4 with ONTAP release.

NVMe-oF Host Configuration for SUSE Linux Enterprise Server 15 SP3 with ONTAP

NVMe over Fabrics or NVMe-oF (including NVMe/FC and other transports) is supported for SUSE Linux Enterprise Server 15 SP3 with ANA (Asymmetric Namespace Access). ANA is the ALUA equivalent in NVMe-oF environments, and is currently implemented with in-kernel NVMe Multipath. Using this procedure, you can enable NVMe-oF with in-kernel NVMe Multipath using ANA on SUSE Linux Enterprise Server 15 SP3 and ONTAP as the target.

Refer to the Interoperability Matrix Tool for accurate details regarding supported configurations.

Features

- SUSE Linux Enterprise Server 15 SP3 supports NVMe/FC and other transports.
- There is no sanlun support for NVMe-oF. Therefore, there is no Linux Host Utilities support for NVMe-oF on SUSE Linux Enterprise Server 15 SP3. You can rely on the NetApp plug-in included in the native nvme-cli package for NVMe-oF. This should support all NVMe-oF transports.
- Both NVMe and SCSI traffic can be run on the same host. In fact, that is expected to be the commonly deployed host config for customers. Therefore, for SCSI, you might configure `dm-multipath` as usual for SCSI LUNs resulting in mpath devices, whereas NVMe multipath might be used to configure NVMe-oF multipath devices on the host.

Known limitations

SAN booting using the NVMe-oF protocol is currently not supported.

Enable in-kernel NVMe Multipath

In-kernel NVMe multipath is already enabled by default on SUSE Linux Enterprise Server hosts, such as SUSE Linux Enterprise Server 15 SP3. Therefore, no additional setting is required here. Refer to the Interoperability Matrix Tool for accurate details regarding supported configurations.

NVMe-oF initiator packages

Refer to the Interoperability Matrix Tool for accurate details regarding supported configurations.

1. Verify that you have the requisite kernel and nvme-cli MU packages installed on the SUSE Linux Enterprise Server 15 SP3 MU host.

   Example:

   ```
   # uname -r
   5.3.18-59.5-default

   # rpm -qa|grep nvme-cli
   nvme-cli-1.13-3.3.1.x86_64
   ```

The above nvme-cli MU package now includes the following:

- **NVMe/FC auto-connect scripts** - Required for NVMe/FC auto-(re)connect when underlying paths to the namespaces are restored as well as during the host reboot:

```
# rpm -ql nvme-cli-1.13-3.3.1.x86_64
/etc/nvme
/etc/nvme/hostid
/etc/nvme/hostnqn
/usr/lib/systemd/system/nvmefc-boot-connections.service
/usr/lib/systemd/system/nvmefc-connect.target
/usr/lib/systemd/system/nvmefc-connect@.service
...
```

- **ONTAP udev rule** - New udev rule to ensure NVMe multipath round-robin loadbalancer default applies to all ONTAP namespaces:

```
# rpm -ql nvme-cli-1.13-3.3.1.x86_64
/etc/nvme
/etc/nvme/hostid
/etc/nvme/hostnqn
/usr/lib/systemd/system/nvmefc-boot-connections.service
/usr/lib/systemd/system/nvmf-autoconnect.service
/usr/lib/systemd/system/nvmf-connect.target
/usr/lib/systemd/system/nvmf-connect@.service
/usr/lib/udev/rules.d/70-nvmf-autoconnect.rules
/usr/lib/udev/rules.d/71-nvmf-iopolicy-netapp.rules
...
# cat /usr/lib/udev/rules.d/71-nvmf-iopolicy-netapp.rules
# Enable round-robin for NetApp ONTAP and NetApp E-Series
ACTION=="add", SUBSYSTEM=="nvme-subsystem", ATTR{model}=="NetApp
ONTAP Controller", ATTR{iopolicy}="round-robin"
ACTION=="add", SUBSYSTEM=="nvme-subsystem", ATTR{model}=="NetApp E-
Series", ATTR{iopolicy}="round-robin"
```

- **NetApp plug-in for ONTAP devices** - The existing NetApp plug-in has now been modified to handle ONTAP namespaces as well.

2. Check the hostnqn string at `/etc/nvme/hostnqn` on the host and ensure that it properly matches with the hostnqn string for the corresponding subsystem on the ONTAP array. For example,

```
# cat /etc/nvme/hostnqn
nqn.2014-08.org.nvmexpress:uuid:3ca559e1-5588-4fc4-b7d6-5ccfb0b9f054
::> vserver nvme subsystem host show -vserver vs_fcnvme_145
Vserver     Subsystem      Host NQN
-------     ---------      ----------------------------------
vs_nvme_145 nvme_145_1 nqn.2014-08.org.nvmexpress:uuid:c7b07b16-a22e-
41a6-a1fd-cf8262c8713f
            nvme_145_2 nqn.2014-08.org.nvmexpress:uuid:c7b07b16-a22e-
41a6-a1fd-cf8262c8713f
            nvme_145_3 nqn.2014-08.org.nvmexpress:uuid:c7b07b16-a22e-
41a6-a1fd-cf8262c8713f
            nvme_145_4 nqn.2014-08.org.nvmexpress:uuid:c7b07b16-a22e-
41a6-a1fd-cf8262c8713f
            nvme_145_5 nqn.2014-08.org.nvmexpress:uuid:c7b07b16-a22e-
41a6-a1fd-cf8262c8713f
5 entries were displayed.
```

Proceed with the below steps depending on the FC adapter being used on the host.

Configure NVMe/FC

Broadcom/Emulex

1. Verify that you have the recommended adapter and firmware versions. For example,

```
# cat /sys/class/scsi_host/host*/modelname
LPe32002-M2
LPe32002-M2
# cat /sys/class/scsi_host/host*/modeldesc
Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
# cat /sys/class/scsi_host/host*/fwrev
12.8.340.8, sli-4:2:c
12.8.840.8, sli-4:2:c
```

   - The newer lpfc drivers (both inbox and outbox) already have lpfc_enable_fc4_type default set to 3, therefore, you no longer need to set this explicitly in the `/etc/modprobe.d/lpfc.conf`, and recreate the `initrd`. The `lpfc nvme` support is already enabled by default:

   ```
   # cat /sys/module/lpfc/parameters/lpfc_enable_fc4_type
   3
   ```

   - The existing native inbox lpfc driver is already the latest and compatible with NVMe/FC. Therefore, you do not need to install the lpfc oob driver.

```
# cat /sys/module/lpfc/version
0:12.8.0.10
```

2. Verify that the initiator ports are up and running:

```
# cat /sys/class/fc_host/host*/port_name
0x100000109b579d5e
0x100000109b579d5f
# cat /sys/class/fc_host/host*/port_state
Online
Online
```

3. Verify that the NVMe/FC initiator ports are enabled, you are able to see the target ports, and all ports are up and running.
   In the following example, only one initiator port is enabled and connected with two target LIFs:

```
# cat /sys/class/scsi_host/host*/nvme_info
NVME Initiator Enabled
XRI Dist lpfc0 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc0 WWPN x100000109b579d5e WWNN x200000109b579d5e DID
x011c00 ONLINE
NVME RPORT WWPN x208400a098dfdd91 WWNN x208100a098dfdd91 DID x011503
TARGET DISCSRVC ONLINE
NVME RPORT WWPN x208500a098dfdd91 WWNN x208100a098dfdd91 DID x010003
TARGET DISCSRVC ONLINE
NVME Statistics
LS: Xmt 0000000e49 Cmpl 0000000e49 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000003ceb594f Issue 000000003ce65dbe OutIO
fffffffffffb046f
abort 00000bd2 noxri 00000000 nondlp 00000000 qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 000014f4 Err 00012abd
NVME Initiator Enabled
XRI Dist lpfc1 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc1 WWPN x100000109b579d5f WWNN x200000109b579d5f DID
x011b00 ONLINE
NVME RPORT WWPN x208300a098dfdd91 WWNN x208100a098dfdd91 DID x010c03
TARGET DISCSRVC ONLINE
NVME RPORT WWPN x208200a098dfdd91 WWNN x208100a098dfdd91 DID x012a03
TARGET DISCSRVC ONLINE
NVME Statistics
LS: Xmt 0000000e50 Cmpl 0000000e50 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000003c9859ca Issue 000000003c93515e OutIO
fffffffffffaf794
abort 00000b73 noxri 00000000 nondlp 00000000 qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 0000159d Err 000135c3
```

4. Reboot the host.

Enable 1MB I/O Size (Optional)

ONTAP reports an MDTS (Max Data Transfer Size) of 8 in the Identify Controller data which means the maximum I/O request size should be up to 1MB. However, to issue I/O requests of size 1MB for the Broadcom NVMe/FC host, the lpfc parameter `lpfc_sg_seg_cnt` should also be bumped up to 256 from the default value of 64. Use the following instructions to do so:

1. Append the value 256 in the respective `modprobe lpfc.conf` file:

```
# cat /etc/modprobe.d/lpfc.conf
options lpfc lpfc_sg_seg_cnt=256
```

2. Run the `dracut -f` command, and reboot the host.
3. After reboot, verify that the above setting has been applied by checking the corresponding sysfs value:

```
# cat /sys/module/lpfc/parameters/lpfc_sg_seg_cnt
256
```

Now the Broadcom NVMe/FC host should be able to send up 1MB I/O requests on the ONTAP namespace devices.

Marvell/QLogic

The native inbox qla2xxx driver included in the newer SUSE Linux Enterprise Server 15 SP3 MU kernel has the latest upstream fixes. These fixes are essential for ONTAP support.

1. Verify that you are running the supported adapter driver and firmware versions, for example:

```
# cat /sys/class/fc_host/host*/symbolic_name
QLE2742 FW:v9.06.02 DVR:v10.02.00.106-k
QLE2742 FW:v9.06.02 DVR:v10.02.00.106-k
```

2. Verify `ql2xnvmeenable` is set which enables the Marvell adapter to function as a NVMe/FC initiator:

```
# cat /sys/module/qla2xxx/parameters/ql2xnvmeenable
1
```

Configure NVMe/TCP

Unlike NVMe/FC, NVMe/TCP has no auto-connect functionality. This manifests two major limitations on the Linux NVMe/TCP host:

- **No auto-reconnect after paths get reinstated** NVMe/TCP cannot automatically reconnect to a path that is reinstated beyond the default `ctrl-loss-tmo` timer of 10 minutes following a path down.
- **No auto-connect during host bootup** NVMe/TCP cannot automatically connect during host bootup as well.

You should set the retry period for failover events to at least 30 minutes to prevent timeouts. You can increase the retry period by increasing the value of the ctrl_loss_tmo timer. Following are the details:

Steps

1. Verify whether the initiator port can fetch the discovery log page data across the supported NVMe/TCP LIFs:

```
# nvme discover -t tcp -w 192.168.1.8 -a 192.168.1.51
Discovery Log Number of Records 10, Generation counter 119
=====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.56e362e9bb4f11ebbaded039ea165abc:subsystem.nvme_118_tcp
_1
traddr: 192.168.2.56
sectype: none
=====Discovery Log Entry 1======
trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 1
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.56e362e9bb4f11ebbaded039ea165abc:subsystem.nvme_118_tcp
_1
traddr: 192.168.1.51
sectype: none
=====Discovery Log Entry 2======
trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 0
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.56e362e9bb4f11ebbaded039ea165abc:subsystem.nvme_118_tcp
_2
traddr: 192.168.2.56
sectype: none
...
```

2. Verify that other NVMe/TCP initiator-target LIF combos are able to successfully fetch discovery log page data. For example,

```
# nvme discover -t tcp -w 192.168.1.8 -a 192.168.1.52
# nvme discover -t tcp -w 192.168.2.9 -a 192.168.2.56
# nvme discover -t tcp -w 192.168.2.9 -a 192.168.2.57
```

3. Run `nvme connect-all` command across all the supported NVMe/TCP initiator-target LIFs across the nodes. Ensure you set a longer `ctrl_loss_tmo` timer retry period (for example, 30 minutes, which can be set through `-l 1800`) during the connect-all so that it would retry for a longer period of time in the event of a path loss. For example,

```
# nvme connect-all -t tcp -w 192.168.1.8 -a 192.168.1.51 -l 1800
# nvme connect-all -t tcp -w 192.168.1.8 -a 192.168.1.52 -l 1800
# nvme connect-all -t tcp -w 192.168.2.9 -a 192.168.2.56 -l 1800
# nvme connect-all -t tcp -w 192.168.2.9 -a 192.168.2.57 -l 1800
```

Validate NVMe-oF

1. Verify that in-kernel NVMe multipath is indeed enabled by checking:

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

2. Verify that the appropriate NVMe-oF settings (such as, `model` set to `NetApp ONTAP Controller` and `load balancing iopolicy` set to `round-robin`) for the respective ONTAP namespaces properly reflect on the host:

```
# cat /sys/class/nvme-subsystem/nvme-subsys*/model
NetApp ONTAP Controller
NetApp ONTAP Controller

# cat /sys/class/nvme-subsystem/nvme-subsys*/iopolicy
round-robin
round-robin
```

3. Verify that the ONTAP namespaces properly reflect on the host. For example,

```
# nvme list
Node           SN                    Model                           Namespace
------------   --------------------- ---------------------------------
/dev/nvme0n1   81CZ5BQuUNfGAAAAAAAB  NetApp ONTAP Controller   1

Usage                 Format         FW Rev
-------------------  -----------    --------
85.90 GB / 85.90 GB  4 KiB + 0 B    FFFFFFFF
```

Another example:

```
# nvme list
Node           SN                    Model                           Namespace
------------   --------------------- ---------------------------------
/dev/nvme0n1   81CYrBQuTHQFAAAAAAAC  NetApp ONTAP Controller   1

Usage                 Format         FW Rev
-------------------  -----------    --------
85.90 GB / 85.90 GB  4 KiB + 0 B    FFFFFFFF
```

4. Verify that the controller state of each path is live and has proper ANA status. For example,

```
# nvme list-subsys /dev/nvme1n1
nvme-subsys1 - NQN=nqn.1992-
08.com.netapp:sn.04ba0732530911ea8e8300a098dfdd91:subsystem.nvme_145_1
\
+- nvme2 fc traddr=nn-0x208100a098dfdd91:pn-0x208200a098dfdd91
host_traddr=nn-0x200000109b579d5f:pn-0x100000109b579d5f live non-
optimized
+- nvme3 fc traddr=nn-0x208100a098dfdd91:pn-0x208500a098dfdd91
host_traddr=nn-0x200000109b579d5e:pn-0x100000109b579d5e live non-
optimized
+- nvme4 fc traddr=nn-0x208100a098dfdd91:pn-0x208400a098dfdd91
host_traddr=nn-0x200000109b579d5e:pn-0x100000109b579d5e live optimized
+- nvme6 fc traddr=nn-0x208100a098dfdd91:pn-0x208300a098dfdd91
host_traddr=nn-0x200000109b579d5f:pn-0x100000109b579d5f live optimized
```

Another example:

```
#nvme list-subsys /dev/nvme0n1
nvme-subsys0 - NQN=nqn.1992-
08.com.netapp:sn.37ba7d9cbfba11eba35dd039ea165514:subsystem.nvme_114_tcp
_1
\
+- nvme0 tcp traddr=192.168.2.36 trsvcid=4420 host_traddr=192.168.1.4
live optimized
+- nvme1 tcp traddr=192.168.1.31 trsvcid=4420 host_traddr=192.168.1.4
live optimized
+- nvme10 tcp traddr=192.168.2.37 trsvcid=4420 host_traddr=192.168.1.4
live non-optimized
+- nvme11 tcp traddr=192.168.1.32 trsvcid=4420 host_traddr=192.168.1.4
live non-optimized
+- nvme20 tcp traddr=192.168.2.36 trsvcid=4420 host_traddr=192.168.2.5
live optimized
+- nvme21 tcp traddr=192.168.1.31 trsvcid=4420 host_traddr=192.168.2.5
live optimized
+- nvme30 tcp traddr=192.168.2.37 trsvcid=4420 host_traddr=192.168.2.5
live non-optimized
+- nvme31 tcp traddr=192.168.1.32 trsvcid=4420 host_traddr=192.168.2.5
live non-optimized
```

5. Verify that the NetApp plug-in displays proper values for each ONTAP namespace device. For example,

```
# nvme netapp ontapdevices -o column
Device       Vserver          Namespace Path
---------    -------
--------------------------------------------------
/dev/nvme1n1 vserver_fcnvme_145 /vol/fcnvme_145_vol_1_0_0/fcnvme_145_ns

NSID  UUID                                  Size
----  ------------------------------        ------
1      23766b68-e261-444e-b378-2e84dbe0e5e1  85.90GB


# nvme netapp ontapdevices -o json
{
"ONTAPdevices" : [
     {
       "Device" : "/dev/nvme1n1",
       "Vserver" : "vserver_fcnvme_145",
       "Namespace_Path" : "/vol/fcnvme_145_vol_1_0_0/fcnvme_145_ns",
       "NSID" : 1,
       "UUID" : "23766b68-e261-444e-b378-2e84dbe0e5e1",
       "Size" : "85.90GB",
       "LBA_Data_Size" : 4096,
       "Namespace_Size" : 20971520
     }
  ]
}
```

Another example:

```
# nvme netapp ontapdevices -o column
Device        Vserver          Namespace Path
---------     -------
--------------------------------------------------
/dev/nvme0n1 vs_tcp_114       /vol/tcpnvme_114_1_0_1/tcpnvme_114_ns

NSID  UUID                                   Size
----  ------------------------------         ------
1      a6aee036-e12f-4b07-8e79-4d38a9165686  85.90GB


# nvme netapp ontapdevices -o json
{
     "ONTAPdevices" : [
     {
          "Device" : "/dev/nvme0n1",
           "Vserver" : "vs_tcp_114",
          "Namespace_Path" : "/vol/tcpnvme_114_1_0_1/tcpnvme_114_ns",
          "NSID" : 1,
          "UUID" : "a6aee036-e12f-4b07-8e79-4d38a9165686",
          "Size" : "85.90GB",
          "LBA_Data_Size" : 4096,
          "Namespace_Size" : 20971520
       }
  ]

}
```

Known issues

There are no known issues.

NVMe/FC Host Configuration for SUSE Linux Enterprise Server 15 SP2 with ONTAP

NVMe/FC is supported on ONTAP 9.6 and above with SUSE Linux Enterprise Server 15 SP2. The SUSE Linux Enterprise Server 15 SP2 host can run both NVMe/FC, and FCP traffic through the same fibre channel initiator adapter ports. See the Hardware Universe for a list of supported FC adapters and controllers.

For the current list of supported configurations and versions, see the Interoperability Matrix Tool.

> ⓘ You can use the configuration settings provided in this procedure to configure cloud clients connected to Cloud Volumes ONTAP and Amazon FSx for ONTAP.

Known limitations

SAN booting using the NVMe-oF protocol is currently not supported.

Enable NVMe/FC on SUSE Linux Enterprise Server 15 SP2

1. Upgrade to the recommended SUSE Linux Enterprise Server 15 SP2 MU kernel version.
2. Upgrade the native nvme-cli package.

   This native nvme-cli package contains the NVMe/FC auto-connect scripts, ONTAP udev rule which enables round-robin load balancing for NVMe Multipath as well as the NetApp plug-in for ONTAP namespaces.

   ```
   # rpm -qa|grep nvme-cli
   nvme-cli-1.10-2.38.x86_64
   ```

3. On the SUSE Linux Enterprise Server 15 SP2 host, check the host NQN string at `/etc/nvme/hostnqn` and verify that it matches the host NQN string for the corresponding subsystem on the ONTAP array. For example:

   ```
   # cat /etc/nvme/hostnqn
   nqn.2014-08.org.nvmexpress:uuid:3ca559e1-5588-4fc4-b7d6-5ccfb0b9f054
   ```

   ```
   ::> vserver nvme subsystem host show -vserver vs_fcnvme_145
   Vserver Subsystem Host NQN
   ------- ---------
   ----------------------------------------------------------
   vs_fcnvme_145
   nvme_145_1
   nqn.2014-08.org.nvmexpress:uuid:c7b07b16-a22e-41a6-a1fd-cf8262c8713f
   nvme_145_2
   nqn.2014-08.org.nvmexpress:uuid:c7b07b16-a22e-41a6-a1fd-cf8262c8713f
   nvme_145_3
   nqn.2014-08.org.nvmexpress:uuid:c7b07b16-a22e-41a6-a1fd-cf8262c8713f
   nvme_145_4
   nqn.2014-08.org.nvmexpress:uuid:c7b07b16-a22e-41a6-a1fd-cf8262c8713f
   nvme_145_5
   nqn.2014-08.org.nvmexpress:uuid:c7b07b16-a22e-41a6-a1fd-cf8262c8713f
   5 entries were displayed.
   ```

4. Reboot the host.

Configure the Broadcom FC Adapter for NVMe/FC

1. Verify that you are using the supported adapter. For the current list of supported adapters see the Interoperability Matrix Tool.

```
# cat /sys/class/scsi_host/host*/modelname
LPe32002-M2
LPe32002-M2
```

```
# cat /sys/class/scsi_host/host*/modeldesc
Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
```

2. Verify that you are using the recommended Broadcom lpfc firmware and native inbox driver versions.

```
# cat /sys/class/scsi_host/host*/fwrev
12.6.240.40, sli-4:2:c
12.6.240.40, sli-4:2:c
```

```
# cat /sys/module/lpfc/version
0:12.8.0.2
```

3. Verify that lpfc_enable_fc4_type is set to 3.

```
# cat /sys/module/lpfc/parameters/lpfc_enable_fc4_type
3
```

4. Verify that the initiator ports are up and running.

```
# cat /sys/class/fc_host/host*/port_name
0x100000109b579d5e
0x100000109b579d5f
```

```
# cat /sys/class/fc_host/host*/port_state
Online
Online
```

5. Verify that the NVMe/FC initiator ports are enabled, running and able to see the target LIFs.

```
# cat /sys/class/scsi_host/host*/nvme_info
NVME Initiator Enabled
XRI Dist lpfc0 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc0 WWPN x100000109b579d5e WWNN x200000109b579d5e DID
x011c00 ONLINE
NVME RPORT WWPN x208400a098dfdd91 WWNN x208100a098dfdd91 DID x011503
TARGET DISCSRVC ONLINE
NVME RPORT WWPN x208500a098dfdd91 WWNN x208100a098dfdd91 DID x010003
TARGET DISCSRVC ONLINE
NVME Statistics
LS: Xmt 0000000e49 Cmpl 0000000e49 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000003ceb594f Issue 000000003ce65dbe OutIO
fffffffffffb046f
abort 00000bd2 noxri 00000000 nondlp 00000000 qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 000014f4 Err 00012abd
NVME Initiator Enabled
XRI Dist lpfc1 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc1 WWPN x100000109b579d5f WWNN x200000109b579d5f DID
x011b00 ONLINE
NVME RPORT WWPN x208300a098dfdd91 WWNN x208100a098dfdd91 DID x010c03
TARGET DISCSRVC ONLINE
NVME RPORT WWPN x208200a098dfdd91 WWNN x208100a098dfdd91 DID x012a03
TARGET DISCSRVC ONLINE
NVME Statistics
LS: Xmt 0000000e50 Cmpl 0000000e50 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000003c9859ca Issue 000000003c93515e OutIO
ffffffffffffaf794
abort 00000b73 noxri 00000000 nondlp 00000000 qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 0000159d Err 000135c3
```

Validate NVMe/FC

1. Verify the following NVMe/FC settings.

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

```
# cat /sys/class/nvme-subsystem/nvme-subsys*/model
NetApp ONTAP Controller
```

```
# cat /sys/class/nvme-subsystem/nvme-subsys*/iopolicy
round-robin
```

2. Verify that the namespaces are created.

```
# nvme list
Node SN Model Namespace Usage Format FW Rev
---------------- --------------------
---------------------------------------- ---------
-------------------------- ---------------- --------
/dev/nvme1n1 814vWBNRwfBGAAAAAAAB NetApp ONTAP Controller 1 85.90 GB /
85.90 GB 4 KiB + 0 B FFFFFFFF
```

3. Verify the status of the ANA paths.

```
# nvme list-subsys /dev/nvme1n1
nvme-subsys1 - NQN=nqn.1992-
08.com.netapp:sn.04ba0732530911ea8e8300a098dfdd91:subsystem.nvme_145_1
\
+- nvme2 fc traddr=nn-0x208100a098dfdd91:pn-0x208200a098dfdd91
host_traddr=nn-0x200000109b579d5f:pn-0x100000109b579d5f live
inaccessible
+- nvme3 fc traddr=nn-0x208100a098dfdd91:pn-0x208500a098dfdd91
host_traddr=nn-0x200000109b579d5e:pn-0x100000109b579d5e live
inaccessible
+- nvme4 fc traddr=nn-0x208100a098dfdd91:pn-0x208400a098dfdd91
host_traddr=nn-0x200000109b579d5e:pn-0x100000109b579d5e live optimized
+- nvme6 fc traddr=nn-0x208100a098dfdd91:pn-0x208300a098dfdd91
host_traddr=nn-0x200000109b579d5f:pn-0x100000109b579d5f live optimized
```

4. Verify the NetApp plug-in for ONTAP devices.

```
# nvme netapp ontapdevices -o column
Device Vserver Namespace Path NSID UUID Size
---------------- -------------------------
-------------------------------------------------- ----
-------------------------------------- ---------
/dev/nvme1n1 vserver_fcnvme_145 /vol/fcnvme_145_vol_1_0_0/fcnvme_145_ns
1 23766b68-e261-444e-b378-2e84dbe0e5e1 85.90GB

# nvme netapp ontapdevices -o json
{
"ONTAPdevices" : [
     {
       "Device" : "/dev/nvme1n1",
       "Vserver" : "vserver_fcnvme_145",
       "Namespace_Path" : "/vol/fcnvme_145_vol_1_0_0/fcnvme_145_ns",
       "NSID" : 1,
       "UUID" : "23766b68-e261-444e-b378-2e84dbe0e5e1",
       "Size" : "85.90GB",
       "LBA_Data_Size" : 4096,
       "Namespace_Size" : 20971520
     },
  ]
}
```

Known issues

There are no known issues.

Enable 1MB I/O Size for Broadcom NVMe/FC

ONTAP reports an MDTS (Max Data Transfer Size) of 8 in the Identify Controller data. This means the maximum I/O request size can be up to 1MB. To issue I/O requests of size 1MB for a Broadcom NVMe/FC host, you should increase the `lpfc` value of the `lpfc_sg_seg_cnt` parameter to 256 from the default value of 64.

> These steps don't apply to Qlogic NVMe/FC hosts.

Steps

1. Set the `lpfc_sg_seg_cnt` parameter to 256:

   ```
   cat /etc/modprobe.d/lpfc.conf
   ```

   You should see an output similar to the following example:

```
options lpfc lpfc_sg_seg_cnt=256
```

2. Run the `dracut -f` command, and reboot the host.
3. Verify that the value for `lpfc_sg_seg_cnt` is 256:

```
cat /sys/module/lpfc/parameters/lpfc_sg_seg_cnt
```

LPFC Verbose Logging

Set the lpfc driver for NVMe/FC.

Steps

1. Set the `lpfc_log_verbose` driver setting to any of the following values to log NVMe/FC events.

```
#define LOG_NVME 0x00100000 /* NVME general events. */
#define LOG_NVME_DISC 0x00200000 /* NVME Discovery/Connect events. */
#define LOG_NVME_ABTS 0x00400000 /* NVME ABTS events. */
#define LOG_NVME_IOERR 0x00800000 /* NVME IO Error events. */
```

2. After setting the values, run the `dracut-f` command and reboot the host.
3. Verify the settings.

```
# cat /etc/modprobe.d/lpfc.conf options lpfc lpfc_log_verbose=0xf00083

# cat /sys/module/lpfc/parameters/lpfc_log_verbose 15728771
```

NVMe/FC Host Configuration for SUSE Linux Enterprise Server 15 SP1 with ONTAP

You can configure NVMe over Fibre Channel (NVMe/FC) on hosts running SUSE Linux Enterprise Server 15 SP1 and ONTAP as the target.

Beginning with ONTAP 9.6, NVMe/FC is supported for the following versions of SUSE Linux Enterprise Server:

- SUSE Linux Enterprise Server 15 SP1

  The SUSE Linux Enterprise Server 15 SP1 host can run both NVMe/FC and FCP traffic through the same fibre channel initiator adapter ports. See the Hardware Universe for a list of supported FC adapters and controllers.

  For the current list of supported configurations and versions, see the Interoperability Matrix Tool.

- Native NVMe/FC auto-connect scripts are included in the nvme-cli package. You can use the native inbox lpfc driver on SUSE Linux Enterprise Server 15 SP1.

Known limitations

SAN booting using the NVMe-oF protocol is currently not supported.

Enable NVMe/FC on SUSE Linux Enterprise Server 15 SP1

1. Upgrade to the recommended SUSE Linux Enterprise Server 15 SP2 MU kernel
2. Upgrade to the recommended nvme-cli MU version.

   This nvme-cli package contains the native NVMe/FC auto-connect scripts, so you do not need to install the external NVMe/FC auto-connect scripts provided by Broadcom on the SUSE Linux Enterprise Server 15 SP1 host. This package also includes the ONTAP udev rule which enables round-robin load balancing for NVMe multipath, and the NetApp plug-in for ONTAP devices.

   ```
   # rpm -qa | grep nvme-cli
   nvme-cli-1.8.1-6.9.1.x86_64
   ```

3. On the SUSE Linux Enterprise Server 15 SP1 host, check the host NQN string at `/etc/nvme/hostnqn` and verify that it matches the host NQN string for the corresponding subsystem on the ONTAP array. For example:

   ```
   # cat /etc/nvme/hostnqn
   nqn.2014-08.org.nvmexpress:uuid:75953f3b-77fe-4e03-bf3c-09d5a156fbcd
   ```

   ```
   *> vserver nvme subsystem host show -vserver vs_nvme_10
   Vserver Subsystem Host NQN
   ------- --------- -------------------------------------- -----------
   sles_117_nvme_ss_10_0
   nqn.2014-08.org.nvmexpress:uuid:75953f3b-77fe-4e03-bf3c-09d5a156fbcd
   ```

4. Reboot the host.

Configure the Broadcom FC Adapter for NVMe/FC

1. Verify that you are using the supported adapter. For the current list of supported adapters see the Interoperability Matrix Tool.

   ```
   # cat /sys/class/scsi_host/host*/modelname
   LPe32002-M2
   LPe32002-M2
   ```

   ```
   # cat /sys/class/scsi_host/host*/modeldesc
   Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
   Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
   ```

2. Verify that you are using the recommended Broadcom lpfc firmware and native inbox driver versions.

```
# cat /sys/class/scsi_host/host*/fwrev
12.4.243.17, sil-4.2.c
12.4.243.17, sil-4.2.c
```

```
# cat /sys/module/lpfc/version
0:12.6.0.0
```

3. Verify that lpfc_enable_fc4_type is set to 3.

```
# cat /sys/module/lpfc/parameters/lpfc_enable_fc4_type
3
```

4. Verify that the initiator ports are up and running.

```
# cat /sys/class/fc_host/host*/port_name
0x10000090fae0ec61
0x10000090fae0ec62
```

```
# cat /sys/class/fc_host/host*/port_state
Online
Online
```

5. Verify that the NVMe/FC initiator ports are enabled, running and able to see the target LIFs.

```
# cat /sys/class/scsi_host/host*/nvme_info
NVME Initiator Enabled
XRI Dist lpfc0 Total 6144 NVME 2947 SCSI 2977 ELS 250
NVME LPORT lpfc0 WWPN x10000090fae0ec61 WWNN x20000090fae0ec61 DID
x012000 ONLINE
NVME RPORT WWPN x202d00a098c80f09 WWNN x202c00a098c80f09 DID x010201
TARGET DISCSRVC ONLINE
NVME RPORT WWPN x203100a098c80f09 WWNN x202c00a098c80f09 DID x010601
TARGET DISCSRVC ONLINE
NVME Statistics
…
```

Validate NVMe/FC

1. Verify the following NVMe/FC settings.

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

```
# cat /sys/class/nvme-subsystem/nvme-subsys*/model
NetApp ONTAP Controller
NetApp ONTAP Controller
```

```
# cat /sys/class/nvme-subsystem/nvme-subsys*/iopolicy
round-robin
round-robin
```

2. Verify that the namespaces are created.

```
# nvme list
Node SN Model Namespace Usage Format FW Rev
---------------- -------------------- -----------------------
/dev/nvme0n1 80BADBKnB/JvAAAAAAAC NetApp ONTAP Controller 1 53.69 GB /
53.69 GB 4 KiB + 0 B FFFFFFFF
```

3. Verify the status of the ANA paths.

```
# nvme list-subsys/dev/nvme0n1
Nvme-subsysf0 – NQN=nqn.1992-
08.com.netapp:sn.341541339b9511e8a9b500a098c80f09:subsystem.sles_117_nvm
e_ss_10_0
\
+- nvme0 fc traddr=nn-0x202c00a098c80f09:pn-0x202d00a098c80f09
host_traddr=nn-0x20000090fae0ec61:pn-0x10000090fae0ec61 live optimized
+- nvme1 fc traddr=nn-0x207300a098dfdd91:pn-0x207600a098dfdd91
host_traddr=nn-0x200000109b1c1204:pn-0x100000109b1c1204 live
inaccessible
+- nvme2 fc traddr=nn-0x207300a098dfdd91:pn-0x207500a098dfdd91
host_traddr=nn-0x200000109b1c1205:pn-0x100000109b1c1205 live optimized
+- nvme3 fc traddr=nn-0x207300a098dfdd91:pn-0x207700a098dfdd91 host
traddr=nn-0x200000109b1c1205:pn-0x100000109b1c1205 live inaccessible
```

4. Verify the NetApp plug-in for ONTAP devices.

```
# nvme netapp ontapdevices -o column
Device    Vserver   Namespace Path             NSID   UUID   Size
-------  --------- -------------------------  ------ ----- -----
/dev/nvme0n1   vs_nvme_10       /vol/sles_117_vol_10_0/sles_117_ns_10_0
1        55baf453-f629-4a18-9364-b6aee3f50dad   53.69GB

# nvme netapp ontapdevices -o json
{
   "ONTAPdevices" : [
   {
        Device" : "/dev/nvme0n1",
        "Vserver" : "vs_nvme_10",
        "Namespace_Path" : "/vol/sles_117_vol_10_0/sles_117_ns_10_0",
         "NSID" : 1,
         "UUID" : "55baf453-f629-4a18-9364-b6aee3f50dad",
         "Size" : "53.69GB",
         "LBA_Data_Size" : 4096,
         "Namespace_Size" : 13107200
    }
]
```

Known issues

There are no known issues.

Enable 1MB I/O Size for Broadcom NVMe/FC

ONTAP reports an MDTS (Max Data Transfer Size) of 8 in the Identify Controller data. This means the maximum I/O request size can be up to 1MB. To issue I/O requests of size 1MB for a Broadcom NVMe/FC host, you should increase the `lpfc` value of the `lpfc_sg_seg_cnt` parameter to 256 from the default value of 64.

These steps don't apply to Qlogic NVMe/FC hosts.

Steps

1. Set the `lpfc_sg_seg_cnt` parameter to 256:

   ```
   cat /etc/modprobe.d/lpfc.conf
   ```

   You should see an output similar to the following example:

   ```
   options lpfc lpfc_sg_seg_cnt=256
   ```

2. Run the `dracut -f` command, and reboot the host.

3. Verify that the value for `lpfc_sg_seg_cnt` is 256:

```
cat /sys/module/lpfc/parameters/lpfc_sg_seg_cnt
```

LPFC Verbose Logging

Set the lpfc driver for NVMe/FC.

Steps

1. Set the `lpfc_log_verbose` driver setting to any of the following values to log NVMe/FC events.

```
#define LOG_NVME 0x00100000 /* NVME general events. */
#define LOG_NVME_DISC 0x00200000 /* NVME Discovery/Connect events. */
#define LOG_NVME_ABTS 0x00400000 /* NVME ABTS events. */
#define LOG_NVME_IOERR 0x00800000 /* NVME IO Error events. */
```

2. After setting the values, run the `dracut-f` command and reboot the host.
3. Verify the settings.

```
# cat /etc/modprobe.d/lpfc.conf options lpfc lpfc_log_verbose=0xf00083

# cat /sys/module/lpfc/parameters/lpfc_log_verbose 15728771
```

Ubuntu

NVMe-oF host configuration for Ubuntu 24.04 with ONTAP

NVMe over Fabrics (NVMe-oF), including NVMe over Fibre Channel (NVMe/FC) and other transports, is supported with Ubuntu 24.04 and Asymmetric Namespace Access (ANA). In NVMe-oF environments, ANA is the equivalent of ALUA multipathing in iSCSI and FC environments and is implemented with in-kernel NVMe multipath.

The following support is available for the NVMe-oF host configuration for Ubuntu 24.04 with ONTAP:

- The NetApp plug-in in the native nvme-cli package displays ONTAP details for NVMe/FC namespaces.
- Use of NVMe and SCSI co-existent traffic on the same host on a given host bus adapter (HBA), without the explicit dm-multipath settings to prevent claiming NVMe namespaces.

For additional details on supported configurations, see the Interoperability Matrix Tool.

Features

Ubuntu 24.04 has in-kernel NVMe multipath enabled for NVMe namespaces by default. This means you don’t need explicit settings.

Known limitations

SAN booting using the NVMe-oF protocol is currently not supported for Ubuntu 24.04 with ONTAP.

Validate software versions

You can use the following procedure to validate the minimum supported Ubuntu 24.04 software versions.

Steps

1. Install Ubuntu 24.04 on the server. After the installation is complete, verify that you are running the specified Ubuntu 24.04 kernel:

   ```
   uname -r
   ```

   ```
   6.8.0-31-generic
   ```

2. Install the `nvme-cli` package:

   ```
   apt list | grep nvme
   ```

   ```
   nvme-cli/noble-updates 2.8-1ubuntu0.1 amd64
   ```

3. On the Ubuntu 24.04 host, check the hostnqn string at `/etc/nvme/hostnqn`:

   ```
   cat /etc/nvme/hostnqn
   ```

   ```
   nqn.2014-08.org.nvmexpress:uuid:ace18dd8-1f5a-11ec-b0c3-3a68dd61a6ff
   ```

4. Verify that the `hostnqn` string from `/etc/nvme/hostnqn` matches the `hostnqn` string for the corresponding subsystem on the ONTAP array:

   ```
   vserver nvme subsystem host show -vserver vs_106_fc_nvme
   ```

   ```
   Vserver     Subsystem          Host NQN
   ----------- ---------------
   ----------------------------------------------------------
   vs_106_fc_nvme ub_106 nqn.2014-08.org.nvmexpress:uuid:c04702c8-e91e-
   4353-9995-ba4536214631
   ```

> If the `hostnqn` strings don't match, use the `vserver modify` command to update the `hostnqn` string on your corresponding ONTAP array subsystem to match the `hostnqn` string from `/etc/nvme/hostnqn` on the host.

Configure NVMe/FC

You can configure NVMe/FC for Broadcom/Emulex or Marvell/Qlogic adapters.

Broadcom/Emulex

Configure NVMe/FC for a Broadcom/Emulex adapter.

Steps

1. Verify that you're using the supported adapter model:

   a. `cat /sys/class/scsi_host/host*/modelname`

   ```
   LPe36002-M64
   LPe36002-M64
   ```

   b. `cat /sys/class/scsi_host/host*/modeldesc`

   ```
   Emulex LPe36002-M64 2-Port 64Gb Fibre Channel Adapter
   Emulex LPe36002-M64 2-Port 64Gb Fibre Channel Adapter
   ```

2. Verify that you're using the recommended Broadcom `lpfc` firmware and inbox driver.

   a. `cat /sys/class/scsi_host/host*/fwrev`

   ```
   14.4.317.10, sli-4:6:d
   14.4.317.10, sli-4:6:d
   ```

   b. `cat /sys/module/lpfc/version`

   ```
   0:14.2.0.17
   ```

   For the current list of supported adapter driver and firmware versions, see the Interoperability Matrix Tool.

3. Verify that `lpfc_enable_fc4_type` is set to `3`:

   ```
   cat /sys/module/lpfc/parameters/lpfc_enable_fc4_type
   ```

   The expected output is 3.

4. Verify that the initiator ports are up and running and that you can see the target LIFs:

   a. `cat /sys/class/fc_host/host*/port_name`

   ```
   0x100000109bf0447b
   0x100000109bf0447c
   ```

b. `cat /sys/class/fc_host/host*/port_state`

```
Online
Online
```

c. `cat /sys/class/scsi_host/host*/nvme_info`

Show example output

```
NVME Initiator Enabled
XRI Dist lpfc0 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc0 WWPN x100000109bf0447b WWNN x200000109bf0447b
DID x022600 ONLINE
NVME RPORT       WWPN x200fd039eaa8138b WWNN x200ad039eaa8138b
DID x021006 TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 0000000187 Cmpl 0000000187 Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 0000000014096514 Issue 000000001407fcd6 OutIO
fffffffffffe97c2
        abort 00000048 noxri 00000000 nondlp 0000001c qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 00000048 Err 00000077

NVME Initiator Enabled
XRI Dist lpfc1 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc1 WWPN x100000109bf0447c WWNN x200000109bf0447c
DID x022300 ONLINE
NVME RPORT       WWPN x2010d039eaa8138b WWNN x200ad039eaa8138b
DID x021106 TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 0000000187 Cmpl 0000000187 Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 00000000140970ed Issue 00000000140813da OutIO
fffffffffffea2ed
        abort 00000047 noxri 00000000 nondlp 0000002b qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 00000047 Err 00000075
```

Marvell/QLogic

The native inbox qla2xxx driver included in the Ubuntu 24.04 GA kernel has the latest upstream fixes.

These fixes are essential for ONTAP support.

Configure NVMe/FC for a Marvell/QLogic adapter.

Steps

1. Verify that you are running the supported adapter driver and firmware versions:

```
cat /sys/class/fc_host/host*/symbolic_name
```

```
QLE2872 FW: v9.15.00 DVR: v10.02.09.100-k
QLE2872 FW: v9.15.00 DVR: v10.02.09.100-k
```

2. Verify that `ql2xnvmeenable` is set. This enables the Marvell adapter to function as an NVMe/FC initiator:

```
cat /sys/module/qla2xxx/parameters/ql2xnvmeenable
```

The expected ouptut is 1.

Enable 1MB I/O (Optional)

ONTAP reports an MDTS (Max Data Transfer Size) of 8 in the Identify Controller data. This means the maximum I/O request size can be up to 1MB. To issue I/O requests of size 1MB for a Broadcom NVMe/FC host, you should increase the `lpfc` value of the `lpfc_sg_seg_cnt` parameter to 256 from the default value of 64.

> These steps don't apply to Qlogic NVMe/FC hosts.

Steps

1. Set the `lpfc_sg_seg_cnt` parameter to 256:

```
cat /etc/modprobe.d/lpfc.conf
```

You should see an output similar to the following example:

```
options lpfc lpfc_sg_seg_cnt=256
```

2. Run the `dracut -f` command, and reboot the host.
3. Verify that the value for `lpfc_sg_seg_cnt` is 256:

```
cat /sys/module/lpfc/parameters/lpfc_sg_seg_cnt
```

Configure NVMe/TCP

NVMe/TCP doesn't support the auto-connect functionality. Instead, you can manually discover the NVMe/TCP subsystems and namespaces by using the `connect` or `connect-all` commands.

Steps

1. Verify that the initiator port can fetch the discovery log page data across the supported NVMe/TCP LIFs:

   ```
   nvme discover -t tcp -w <host-traddr> -a <traddr>
   ```

Show example

```
# nvme discover -t tcp -w 192.168.167.150 -a 192.168.167.155
Discovery Log Number of Records 8, Generation counter 10
=====Discovery Log Entry 0======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  4
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.9b7d42b764ff11efb8fed039eabac370:discovery
traddr:  192.168.167.156
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 1======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  2
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.9b7d42b764ff11efb8fed039eabac370:discovery
traddr:  192.168.166.156
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 2======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  3
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.9b7d42b764ff11efb8fed039eabac370:discovery
traddr:  192.168.167.155
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 3======
trtype:  tcp
adrfam:  ipv4
```

```
subtype: current discovery subsystem
treq:    not specified
portid:  1
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.9b7d42b764ff11efb8fed039eabac370:discovery
traddr:  192.168.166.155
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 4======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.9b7d42b764ff11efb8fed039eabac370:subsystem.ubuntu_2
4.04_tcp_211
traddr:  192.168.167.156
eflags:  none
sectype: none
=====Discovery Log Entry 5======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.9b7d42b764ff11efb8fed039eabac370:subsystem.ubuntu_2
4.04_tcp_211
traddr:  192.168.166.156
eflags:  none
sectype: none
=====Discovery Log Entry 6======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  3
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.9b7d42b764ff11efb8fed039eabac370:subsystem.ubuntu_2
4.04_tcp_211
```

```
traddr:  192.168.167.155
eflags:  none
sectype: none
=====Discovery Log Entry 7======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.9b7d42b764ff11efb8fed039eabac370:subsystem.ubuntu_2
4.04_tcp_211
traddr:  192.168.166.155
eflags:  none
sectype: none
```

2. Verify that the other NVMe/TCP initiator-target LIF combinations can fetch discovery log page data:

```
nvme discover -t tcp -w <host-traddr> -a <traddr>
```

Show example output

```
#nvme discover -t tcp -w 192.168.167.150 -a 192.168.167.155
#nvme discover -t tcp -w 192.168.167.150 -a 192.168.167.156
#nvme discover -t tcp -w 192.168.166.150 -a 192.168.166.155
#nvme discover -t tcp -w 192.168.166.150 -a 192.168.166.156
```

3. Run the `nvme connect-all` command across all the supported NVMe/TCP initiator-target LIFs across the nodes:

```
nvme connect-all -t tcp -w <host-traddr> -a <traddr>
```

Show example output

```
#nvme connect-all -t tcp -w 192.168.167.150 -a 192.168.167.155
#nvme connect-all -t tcp -w 192.168.167.150 -a 192.168.167.156
#nvme connect-all -t tcp -w 192.168.166.150 -a 192.168.166.155
#nvme connect-all -t tcp -w 192.168.166.150 -a 192.168.166.156
```

Beginning with Ubuntu 24.04, the ctrl_loss_tmo timeout default setting for NVMe/TCP is turned off. This means there is no limit on the number of retries (indefinite retry) and you don’t need to manually configure a specific ctrl_loss_tmo timeout duration when using the `nvme connect` or `nvme connect-all` commands (option -l). With this default behavior, the NVMe/TCP controllers don’t experience timeouts in the event of a path failure and remain connected indefinitely.

Validate NVMe-oF

You can use the following procedure to validate NVME-oF.

Steps

1. Verify that the in-kernel NVMe multipath is enabled:

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

   The expected output is "Y".

2. Verify that the appropriate NVMe-oF settings (such as, model set to "NetApp ONTAP Controller" and load balancing iopolicy set to "round-robin") for the respective ONTAP namespaces correctly display on the host:

   a. `cat /sys/class/nvme-subsystem/nvme-subsys*/model`

      ```
      NetApp ONTAP Controller
      NetApp ONTAP Controller
      ```

   b. `cat /sys/class/nvme-subsystem/nvme-subsys*/iopolicy`

      ```
      round-robin
      round-robin
      ```

3. Verify that the namespaces are created and correctly discovered on the host:

   ```
   nvme list
   ```

Show example output

```
Node          SN                         Model
---------------------------------------------------------
/dev/nvme0n1 81CZ5BQuUNfGAAAAAAAB   NetApp ONTAP Controller


Namespace Usage   Format                FW            Rev
-----------------------------------------------------------
1                 21.47 GB / 21.47 GB   4 KiB + 0 B   FFFFFFFF
```

4. Verify that the controller state of each path is live and has the correct ANA status:

NVMe/FC

```
nvme list-subsys /dev/nvme0n1
```

Show example output

```
nvme-subsys4 - NQN=nqn.1992-
08.com.netapp:sn.8763d311b2ac11ed950ed039ea951c46:subsystem.
ubuntu_24.04 \
+- nvme1 fc traddr=nn-0x20a6d039ea954d17:pn-
0x20a7d039ea954d17,host_traddr=nn-0x200000109b1b95ef:pn-
0x100000109b1b95ef live optimized
+- nvme2 fc traddr=nn-0x20a6d039ea954d17:pn-
0x20a8d039ea954d17,host_traddr=nn-0x200000109b1b95f0:pn-
0x100000109b1b95f0 live optimized
+- nvme3 fc traddr=nn-0x20a6d039ea954d17:pn-
0x20aad039ea954d17,host_traddr=nn-0x200000109b1b95f0:pn-
0x100000109b1b95f0 live non-optimized
+- nvme5 fc traddr=nn-0x20a6d039ea954d17:pn-
0x20a9d039ea954d17,host_traddr=nn-0x200000109b1b95ef:pn-
0x100000109b1b95ef live non-optimized
```

NVMe/TCP

```
nvme list-subsys /dev/nvme1n1
```

Show example output

```
nvme-subsys0 - NQN=nqn.1992-
08.com.netapp:sn.9b7d42b764ff11efb8fed039eabac370:subsystem.ubun
tu_24.04_tcp_211
               hostnqn=nqn.2014-08.org.nvmexpress:uuid:4c4c4544-
0050-3410-8035-c3c04f4a5933
               iopolicy=round-robin
 +- nvme0 tcp
traddr=192.168.166.155,trsvcid=4420,host_traddr=192.168.166.150,
src_addr=192.168.166.150 live optimized
 +- nvme1 tcp
traddr=192.168.167.155,trsvcid=4420,host_traddr=192.168.167.150,
src_addr=192.168.167.150 live optimized
 +- nvme2 tcp
traddr=192.168.166.156,trsvcid=4420,host_traddr=192.168.166.150,
src_addr=192.168.166.150 live non-optimized
 +- nvme3 tcp
traddr=192.168.167.156,trsvcid=4420,host_traddr=192.168.167.150,
src_addr=192.168.167.150 live non-optimized
```

5. Verify that the NetApp plug-in displays the correct values for each ONTAP namespace device:

Column

```
nvme netapp ontapdevices -o column
```

Show example output

```
Device         Vserver        Namespace Path          NSID UUID
Size
------------- -------------  ----------------------- ----
-------------------------------------- ---------
/dev/nvme0n1   vs_211_tcp    /vol/tcpvol1/ns1        1
1cc7bc78-8d7b-4d8e-a3c4-750f9461a6e9   21.47GB
```

JSON

```
nvme netapp ontapdevices -o json
```

Show example output

```
{

"ONTAPdevices" : [
{
      "Device":"/dev/nvme0n9",
      "Vserver":"vs_211_tcp",
      "Namespace_Path":"/vol/tcpvol9/ns9",
      "NSID":9,
      "UUID":"99640dd9-8463-4c12-8282-b525b39fc10b",
      "Size":"21.47GB",
      "LBA_Data_Size":4096,
      "Namespace_Size":5242880
    }
  ]
}
```

Known issues

There are no known issues for the NVMe-oF host configuration for Ubuntu 24.04 with ONTAP release.

NVMe-oF host configuration for Ubuntu 24.04 with ONTAP

NVMe over Fabrics (NVMe-oF), including NVMe over Fibre Channel (NVMe/FC) and other transports, is supported with Ubuntu 22.04 with Asymmetric Namespace Access (ANA). In NVMe-oF environments, ANA is the equivalent of ALUA multipathing in iSCSI and FC environments and is implemented with in-kernel NVMe multipath.

The following support is available for the NVMe-oF host configuration for Ubuntu 22.04 with ONTAP:

- The NetApp plug-in in the native nvme-cli package displays ONTAP details for NVMe/FC namespaces.
- Use of NVMe and SCSI co-existent traffic on the same host on a given host bus adapter (HBA), without the explicit dm-multipath settings to prevent claiming NVMe namespaces.

For additional details on supported configurations, see the Interoperability Matrix Tool.

Features

Ubuntu 22.04 has in-kernel NVMe multipath enabled for NVMe namespaces by default. Therefore, there is no need for explicit settings.

Known limitations

SAN booting using the NVMe-oF protocol is currently not supported.

Validate software versions

You can use the following procedure to validate the minimum supported Ubuntu 22.04 software versions.

Steps

1. Install Ubuntu 22.04 on the server. After the installation is complete, verify that you are running the specified Ubuntu 22.04 kernel:

   ```
   # uname -r
   ```

   Example output:

   ```
   5.15.0-101-generic
   ```

2. Install the `nvme-cli` package:

   ```
   # apt list | grep nvme
   ```

   Example output:

   ```
   nvme-cli/jammy-updates,now 1.16-3ubuntu0.1 amd64
   ```

3. On the Ubuntu 22.04 host, check the hostnqn string at `/etc/nvme/hostnqn`:

```
# cat /etc/nvme/hostnqn
```

Example output

```
nqn.2014-08.org.nvmexpress:uuid:063a9fa0-438a-4737-b9b4-95a21c66d041
```

4. Verify that the `hostnqn` string matches the `hostnqn` string for the corresponding subsystem on the ONTAP array:

```
::> vserver nvme subsystem host show -vserver vs_106_fc_nvme
```

Example output:

```
Vserver     Subsystem          Host NQN
----------- ---------------
----------------------------------------------------------
vs_106_fc_nvme ub_106 nqn.2014-08.org.nvmexpress:uuid:c04702c8-e91e-
4353-9995-ba4536214631
```

> If the `hostnqn` strings do not match, use the `vserver modify` command to update the `hostnqn` string on your corresponding ONTAP array subsystem to match the `hostnqn` string from `/etc/nvme/hostnqn` on the host.

Configure NVMe/FC

You can configure NVMe/FC for Broadcom/Emulex or Marvell/Qlogic adapters.

Broadcom/Emulex

Steps

1. Verify that you are using the supported adapter model.

```
# cat /sys/class/scsi_host/host*/modelname
```

Example output:

```
LPe36002-M64
LPe36002-M64
```

```
# cat /sys/class/scsi_host/host*/modeldesc
```

Example output:

```
Emulex LPe36002-M64 2-Port 64Gb Fibre Channel Adapter
Emulex LPe36002-M64 2-Port 64Gb Fibre Channel Adapter
```

2. Verify that you are using the recommended Broadcom `lpfc` firmware and inbox driver.

```
# cat /sys/class/scsi_host/host*/fwrev

14.2.673.40, sli-4:6:d
14.2.673.40, sli-4:6:d

# cat /sys/module/lpfc/version
0: 14.0.0.4
```

For the current list of supported adapter driver and firmware versions, see the Interoperability Matrix Tool.

3. Verify that `lpfc_enable_fc4_type` is set to `3`:

```
# cat /sys/module/lpfc/parameters/lpfc_enable_fc4_type
3
```

4. Verify that the initiator ports are up and running and that you can see the target LIFs:

```
# cat /sys/class/fc_host/host*/port_name
0x100000109bf0447c
0x100000109bf0447b
# cat /sys/class/fc_host/host*/port_state
Online
Online
# cat /sys/class/scsi_host/host*/nvme_info
        NVME Initiator Enabled
XRI Dist lpfc1 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc1 WWPN x100000109bf0447c WWNN x200000109bf0447c DID
x022300 ONLINE
NVME RPORT       WWPN x200cd039eaa8138b WWNN x200ad039eaa8138b DID
x021509 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2010d039eaa8138b WWNN x200ad039eaa8138b DID
x021108 TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 000000000e Cmpl 000000000e Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 0000000000005238 Issue 000000000000523a OutIO
0000000000000002
        abort 00000000 noxri 00000000 nondlp 00000000 qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 00000000 Err 00000000

NVME Initiator Enabled
XRI Dist lpfc0 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc0 WWPN x100000109bf0447b WWNN x200000109bf0447b DID
x022600 ONLINE
NVME RPORT       WWPN x200bd039eaa8138b WWNN x200ad039eaa8138b DID
x021409 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x200fd039eaa8138b WWNN x200ad039eaa8138b DID
x021008 TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 000000000e Cmpl 000000000e Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000000000523c Issue 000000000000523e OutIO
0000000000000002
        abort 00000000 noxri 00000000 nondlp 00000000 qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 00000000 Err 00000000
```

Marvell/QLogic FC Adapter for NVMe/FC

The native inbox qla2xxx driver included in the Ubuntu 22.04 GA kernel has the latest upstream fixes.

These fixes are essential for ONTAP support.

Steps

1. Verify that you are running the supported adapter driver and firmware versions:

```
# cat /sys/class/fc_host/host*/symbolic_name
```

Example output

```
QLE2872 FW: v9.14.02 DVR: v10.02.06.200-k
QLE2872 FW: v9.14.02 DVR: v10.02.06.200-k
```

2. Verify that `ql2xnvmeenable` is set. This enables the Marvell adapter to function as an NVMe/FC initiator:

```
# cat /sys/module/qla2xxx/parameters/ql2xnvmeenable
1
```

Enable 1MB I/O (Optional)

ONTAP reports an MDTS (Max Data Transfer Size) of 8 in the Identify Controller data. This means the maximum I/O request size can be up to 1MB. To issue I/O requests of size 1MB for a Broadcom NVMe/FC host, you should increase the `lpfc` value of the `lpfc_sg_seg_cnt` parameter to 256 from the default value of 64.

> ⓘ These steps don't apply to Qlogic NVMe/FC hosts.

Steps

1. Set the `lpfc_sg_seg_cnt` parameter to 256:

```
cat /etc/modprobe.d/lpfc.conf
```

You should see an output similar to the following example:

```
options lpfc lpfc_sg_seg_cnt=256
```

2. Run the `dracut -f` command, and reboot the host.
3. Verify that the value for `lpfc_sg_seg_cnt` is 256:

```
cat /sys/module/lpfc/parameters/lpfc_sg_seg_cnt
```

Configure NVMe/TCP

NVMe/TCP does not have auto-connect functionality. Therefore, if a path goes down and is not reinstated within the default time out period of 10 minutes, NVMe/TCP cannot automatically reconnect. To prevent a time out, you should set the retry period for failover events to at least 30 minutes.

Steps

1. Verify that the initiator port can fetch the discovery log page data across the supported NVMe/TCP LIFs:

   ```
   nvme discover -t tcp -w host-traddr -a traddr
   ```

   Example output:

   ```
   # nvme discover -t tcp -w 10.10.11.47-a 10.10.10.122

   Discovery Log Number of Records 8, Generation counter 10
   =====Discovery Log Entry 0======
   trtype:  tcp
   adrfam:  ipv4
   subtype: current discovery subsystem
   treq:    not specified
   portid:  0
   trsvcid: 8009
   subnqn:  nqn.1992-
   08.com.netapp:sn.bbfb4ee8dfb611edbd07d039ea165590:discovery
   traddr:  10.10.10.122
   eflags:  explicit discovery connections, duplicate discovery information
   sectype: none
   =====Discovery Log Entry 1======
   trtype:  tcp
   adrfam:  ipv4
   subtype: current discovery subsystem
   treq:    not specified
   portid:  1
   trsvcid: 8009
   subnqn:  nqn.1992
   08.com.netapp:sn.bbfb4ee8dfb611edbd07d039ea165590:discovery
   traddr:  10.10.10.124
   eflags:  explicit discovery connections, duplicate discovery information
   sectype: none
   =====Discovery Log Entry 2======
   trtype:  tcp
   ```

2. Verify that the other NVMe/TCP initiator-target LIF combinations are able to successfully fetch discovery log page data:

```
nvme discover -t tcp -w host-traddr -a traddr
```

Example output:

```
#nvme discover -t tcp -w 10.10.10.47 -a 10.10.10.122
#nvme discover -t tcp -w 10.10.10.47 -a 10.10.10.124
#nvme discover -t tcp -w 10.10.11.47 -a 10.10.11.122
#nvme discover -t tcp -w 10.10.11.47 -a 10.10.11.
```

3. Run the nvme connect-all command across all the supported NVMe/TCP initiator-target LIFs across the nodes, and set the controller loss timeout period for at least 30 minutes or 1800 seconds:

```
nvme connect-all -t tcp -w host-traddr -a traddr -l 1800
```

Example output:

```
#   nvme    connect-all -t  tcp -w  10.10.10.47 -a  10.10.10.122 -l 1800
#   nvme    connect-all -t  tcp -w  10.10.10.47 -a  10.10.10.124 -l 1800
#   nvme    connect-all -t  tcp -w  10.10.11.47 -a  10.10.11.122 -l 1800
#   nvme    connect-all -t  tcp -w  10.10.11.47 -a  10.10.11.124 -l 1800
```

Validate NVMe-oF

You can use the following procedure to validate NVME-oF.

Steps

1. Verify that the in-kernel NVMe multipath is enabled:

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

2. Verify that the appropriate NVMe-oF settings (such as, model set to NetApp ONTAP Controller and load balancing iopolicy set to round-robin) for the respective ONTAP namespaces correctly reflect on the host:

```
# cat /sys/class/nvme-subsystem/nvme-subsys*/model
NetApp ONTAP Controller
NetApp ONTAP Controller
```

```
# cat /sys/class/nvme-subsystem/nvme-subsys*/iopolicy
round-robin
round-robin
```

3. Verify that the namespaces are created and correctly discovered on the host:

```
# nvme list
```

Example output:

```
Node         SN                   Model
---------------------------------------------------------
/dev/nvme0n1 81CZ5BQuUNfGAAAAAAAB   NetApp ONTAP Controller


Namespace Usage    Format             FW            Rev
-----------------------------------------------------------
1                 21.47 GB / 21.47 GB   4 KiB + 0 B   FFFFFFFF
```

4. Verify that the controller state of each path is live and has the correct ANA status:

NVMe/FC

```
# nvme list-subsys /dev/nvme0n1
```

Example output:

```
nvme-subsys4 - NQN=nqn.1992-
08.com.netapp:sn.8763d311b2ac11ed950ed039ea951c46:subsystem. ub_106
\
+- nvme1 fc traddr=nn-0x20a6d039ea954d17:pn-
0x20a7d039ea954d17,host_traddr=nn-0x200000109b1b95ef:pn-
0x100000109b1b95ef live optimized
+- nvme2 fc traddr=nn-0x20a6d039ea954d17:pn-
0x20a8d039ea954d17,host_traddr=nn-0x200000109b1b95f0:pn-
0x100000109b1b95f0 live optimized
+- nvme3 fc traddr=nn-0x20a6d039ea954d17:pn-
0x20aad039ea954d17,host_traddr=nn-0x200000109b1b95f0:pn-
0x100000109b1b95f0 live non-optimized
+- nvme5 fc traddr=nn-0x20a6d039ea954d17:pn-
0x20a9d039ea954d17,host_traddr=nn-0x200000109b1b95ef:pn-
0x100000109b1b95ef live non-optimized
```

NVME/TCP

```
# nvme list-subsys /dev/nvme1n1
```

Example output:

```
nvme-subsys1 - NQN=nqn.1992- 08.com.netapp:sn.
bbfb4ee8dfb611edbd07d039ea165590:subsystem.rhel_tcp_95
+- nvme1 tcp
traddr=10.10.10.122,trsvcid=4420,host_traddr=10.10.10.47,src_addr=10
.10.10.47 live
+- nvme2 tcp
traddr=10.10.10.124,trsvcid=4420,host_traddr=10.10.10.47,src_addr=10
.10.10.47 live
+- nvme3 tcp
traddr=10.10.11.122,trsvcid=4420,host_traddr=10.10.11.47,src_addr=10
.10.11.47 live
+- nvme4 tcp
traddr=10.10.11.124,trsvcid=4420,host_traddr=10.10.11.47,src_addr=10
.10.11.47 live
```

5. Verify that the NetApp plug-in displays the correct values for each ONTAP namespace device:

Column

```
# nvme netapp ontapdevices -o column
```

Example output:

```
Device        Vserver   Namespace Path
----------------------- ------------------------------
/dev/nvme0n1 co_iscsi_tcp_ubuntu /vol/vol1/ns1



NSID       UUID                                   Size
------------------------------------------------------------
1          79c2c569-b7fa-42d5-b870-d9d6d7e5fa84 21.47GB
```

JSON

```
# nvme netapp ontapdevices -o json
```

Example output

```
{

"ONTAPdevices" : [
{

"Device" : "/dev/nvme0n1",
"Vserver" : "co_iscsi_tcp_ubuntu",
"Namespace_Path" : "/vol/nvmevol1/ns1",
"NSID" : 1,
"UUID" : "79c2c569-b7fa-42d5-b870-d9d6d7e5fa84",
"Size" : "21.47GB",
"LBA_Data_Size" : 4096,
"Namespace_Size" : 5242880
},

]
}
```

Known issues

The NVMe-oF host configuration for Ubuntu 22.04 with ONTAP release has the following known issue:

NetApp Bug ID	Title	Description
CONTAPEXT-2037	Ubuntu 22.04 NVMe-oF hosts create duplicate Persistent Discovery Controllers	On NVMe-oF hosts, you can use the "nvme discover -p" command to create Persistent Discovery Controllers (PDCs). This command should only create one PDC for each initiator-target combination. However, if you are running Ubuntu 22.04 on an NVMe-oF host, a duplicate PDC is created each time "nvme discover -p" is executed. This leads to unnecessary usage of resources on both the host and the target.

Windows

Configure Windows Server 2025 with NVMe/FC for ONTAP

You can configure NVMe over Fibre Channel (NVMe/FC) on hosts running Windows Server 2025 for operation with ONTAP LUNs.

About this task

You can use the following support with the NVMe/FC host configuration for Windows 2025. You should also review the known limitations before starting the configuration process.

- Support available:

  Beginning with ONTAP 9.10.1, NVMe/FC is supported for Windows Server 2025.

  For a list of supported FC adapters and controllers, see the Hardware Universe. For the current list of supported configurations and versions, see the Interoperability Matrix Tool.

- Known limitations:

  Windows Failover Cluster isn't supported with NVMe/FC because ONTAP doesn't currently support persistent reservations with NVMe/FC.

  > Broadcom ships an external driver for Windows NVMe/FC that is a translational SCSI □ NVMe driver and not a true NVMe/FC driver. The translational overhead doesn't necessarily impact performance, but it does negate the performance benefits of NVMe/FC. As a result, NVMe/FC and FCP performance is the same on Windows servers, unlike other operating systems such as Linux, where NVMe/FC performance is significantly better than that of FCP.

Enable NVMe/FC

Enable FC/NVMe on the Windows initiator host.

Steps

1. Install the Emulex HBA Manager utility on the Windows host.
2. On each of the HBA initiator ports, set the following HBA driver parameters:
   - EnableNVMe = 1

    ◦ NVMEMode = 0

3. Reboot the host.

Configure the Broadcom FC adapter

The Broadcom initiator can serve both NVMe/FC and FCP traffic through the same 32G FC adapter ports. For FCP and FC/NVMe, you should use the Microsft device-specific module (DSM) as the Microsoft Multipath I/O (MPIO) option.

A `hostnqn` is associated with each host bus adapter (HBA) port for the Broadcom adapter with FC/NVMe in a Windows environment. The `hostnqn` is formatted as shown in the following example:

```
nqn.2017-01.com.broadcom:ecd:nvmf:fc:100000109b1b9765
nqn.2017-01.com.broadcom:ecd:nvmf:fc:100000109b1b9766
```

Enable MPIO for NVMe devices

You need to enable MPIO for NVMe devices to complete the NVMe configuration on the Windows host.

Steps

1. Install Windows Host Utility Kit 7.1 to set the driver parameters that are common to both FC and NVMe.
2. Open the MPIO properties.
3. From the **Discover Multi-Paths** tab, add the device ID listed for NVMe.

   MPIO becomes aware of the NVMe devices, which are visible under disk management.

4. Open **Disk Management** and go to **Disk Properties**.
5. From the **MPIO** tab, select **Details**.
6. Set the following Microsoft DSM settings:
   ◦ PathVerifiedPeriod: **10**
   ◦ PathVerifyEnabled: **Enable**
   ◦ RetryCount: **6**
   ◦ RetryInterval: **1**
   ◦ PDORemovedPeriod: **130**
7. Select the MPIO Policy **Round Robin with Subset**.
8. Change the registry values:

   ```
   HKLM\SYSTEM\CurrentControlSet\Services\mpio\Parameters\PathRecoveryInter
   val DWORD -> 30

   HKLM\SYSTEM\CurrentControlSet\Services\mpio \Parameters\
   UseCustomPathRecoveryInterval  DWORD-> 1
   ```

9. Reboot the host.

Validate the NVMe/FC configuration

Verify that the NVMe subsystems have been discovered and the ONTAP namespaces are correct for the NVMe-oF configuration.

Steps

1. Verify that "Port Type" is `FC+NVMe`:

   ```
   listhba
   ```

   Show example

   ```
   Port WWN       : 10:00:00:10:9b:1b:97:65
   Node WWN       : 20:00:00:10:9b:1b:97:65
   Fabric Name    : 10:00:c4:f5:7c:a5:32:e0
   Flags          : 8000e300
   Host Name      : INTEROP-57-159
   Mfg            : Emulex Corporation
   Serial No.     : FC71367217
   Port Number    : 0
   Mode           : Initiator
   PCI Bus Number : 94
   PCI Function   : 0
   Port Type      : FC+NVMe
   Model          : LPe32002-M2

   Port WWN       : 10:00:00:10:9b:1b:97:66
   Node WWN       : 20:00:00:10:9b:1b:97:66
   Fabric Name    : 10:00:c4:f5:7c:a5:32:e0
   Flags          : 8000e300
   Host Name      : INTEROP-57-159
   Mfg            : Emulex Corporation
   Serial No.     : FC71367217
   Port Number    : 1
   Mode           : Initiator
   PCI Bus Number : 94
   PCI Function   : 1
   Port Type      : FC+NVMe
   Model          : LPe32002-M2
   ```

2. Verify that the NVMe/FC subsystems have been discovered:

   - `nvme-list`

Show example

```
NVMe Qualified Name      :  nqn.1992-
08.com.netapp:sn.a3b74c32db2911eab229d039ea141105:subsystem.win_n
vme_interop-57-159
Port WWN                 :  20:09:d0:39:ea:14:11:04
Node WWN                 :  20:05:d0:39:ea:14:11:04
Controller ID            :  0x0180
Model Number             :  NetApp ONTAP Controller
Serial Number            :  81CGZBPU5T/uAAAAAAAB
Firmware Version         :  FFFFFFFF
Total Capacity           :  Not Available
Unallocated Capacity     :  Not Available

NVMe Qualified Name      :  nqn.1992-
08.com.netapp:sn.a3b74c32db2911eab229d039ea141105:subsystem.win_n
vme_interop-57-159
Port WWN                 :  20:06:d0:39:ea:14:11:04
Node WWN                 :  20:05:d0:39:ea:14:11:04
Controller ID            :  0x0181
Model Number             :  NetApp ONTAP Controller
Serial Number            :  81CGZBPU5T/uAAAAAAAB
Firmware Version         :  FFFFFFFF
Total Capacity           :  Not Available
Unallocated Capacity     :  Not Available
Note: At present Namespace Management is not supported by NetApp
Arrays.
```

◦ `nvme-list`

Show example

```
NVMe Qualified Name     :  nqn.1992-
08.com.netapp:sn.a3b74c32db2911eab229d039ea141105:subsystem.win_n
vme_interop-57-159
Port WWN                :  20:07:d0:39:ea:14:11:04
Node WWN                :  20:05:d0:39:ea:14:11:04
Controller ID           :  0x0140
Model Number            :  NetApp ONTAP Controller
Serial Number           :  81CGZBPU5T/uAAAAAAAB
Firmware Version        :  FFFFFFFF
Total Capacity          :  Not Available
Unallocated Capacity    :  Not Available

NVMe Qualified Name     :  nqn.1992-
08.com.netapp:sn.a3b74c32db2911eab229d039ea141105:subsystem.win_n
vme_interop-57-159
Port WWN                :  20:08:d0:39:ea:14:11:04
Node WWN                :  20:05:d0:39:ea:14:11:04
Controller ID           :  0x0141
Model Number            :  NetApp ONTAP Controller
Serial Number           :  81CGZBPU5T/uAAAAAAAB
Firmware Version        :  FFFFFFFF
Total Capacity          :  Not Available
Unallocated Capacity    :  Not Available

Note: At present Namespace Management is not supported by NetApp
Arrays.
```

3. Verify that the namespaces have been created:

```
nvme-list-ns
```

Show example

```
Active Namespaces (attached to controller 0x0141):

                                                SCSI             SCSI
SCSI
   NSID              DeviceName          Bus Number     Target Number
OS LUN
-----------  --------------------  ------------  ---------------
---------
0x00000001   \\.\PHYSICALDRIVE9          0                1
0
0x00000002   \\.\PHYSICALDRIVE10         0                1
1
0x00000003   \\.\PHYSICALDRIVE11         0                1
2
0x00000004   \\.\PHYSICALDRIVE12         0                1
3
0x00000005   \\.\PHYSICALDRIVE13         0                1
4
0x00000006   \\.\PHYSICALDRIVE14         0                1
5
0x00000007   \\.\PHYSICALDRIVE15         0                1
6
0x00000008   \\.\PHYSICALDRIVE16         0                1
7
```

Configure Windows Server 2022 with NVMe/FC for ONTAP

You can configure NVMe over Fibre Channel (NVMe/FC) on hosts running Windows Server 2022 for operation with ONTAP LUNs.

About this task

You can use the following support with the NVMe/FC host configuration for Windows 2022. You should also review the known limitations before starting the configuration process.

- Support available:

  Beginning with ONTAP 9.7, NVMe/FC is supported for Windows Server 2022.

  For a list of supported FC adapters and controllers, see the Hardware Universe. For the current list of supported configurations and versions, see the Interoperability Matrix Tool.

- Known limitations:

  Windows Failover Cluster isn't supported with NVMe/FC because ONTAP doesn't currently support persistent reservations with NVMe/FC.

> Broadcom ships an external driver for Windows NVMe/FC that is a translational SCSI NVMe driver and not a true NVMe/FC driver. The translational overhead doesn't necessarily impact performance, but it does negate the performance benefits of NVMe/FC. As a result, NVMe/FC and FCP performance is the same on Windows servers, unlike other operating systems such as Linux, where NVMe/FC performance is significantly better than that of FCP.

Enable NVMe/FC

Enable FC/NVMe on the Windows initiator host.

Steps

1. Install the Emulex HBA Manager utility on the Windows host.
2. On each of the HBA initiator ports, set the following HBA driver parameters:
   - EnableNVMe = 1
   - NVMEMode = 0
3. Reboot the host.

Configure the Broadcom FC adapter

The Broadcom initiator can serve both NVMe/FC and FCP traffic through the same 32G FC adapter ports. For FCP and FC/NVMe, you should use the Microsoft device-specific module (DSM) as the Microsoft Multipath I/O (MPIO) option.

A `hostnqn` is associated with each host bus adapter (HBA) port for the Broadcom adapter with FC/NVMe in a Windows environment. The `hostnqn` is formatted as shown in the following example:

```
nqn.2017-01.com.broadcom:ecd:nvmf:fc:100000109b1b9765
nqn.2017-01.com.broadcom:ecd:nvmf:fc:100000109b1b9766
```

Enable MPIO for NVMe devices

You need to enable MPIO for NVMe devices to complete the NVMe configuration on the Windows host.

Steps

1. Install Windows Host Utility Kit 7.1 to set the driver parameters that are common to both FC and NVMe.
2. Open the MPIO properties.
3. From the **Discover Multi-Paths** tab, add the device ID listed for NVMe.

   MPIO becomes aware of the NVMe devices, which are visible under disk management.

4. Open **Disk Management** and go to **Disk Properties**.
5. From the **MPIO** tab, select **Details**.
6. Set the following Microsoft DSM settings:
   - PathVerifiedPeriod: **10**

- PathVerifyEnabled: **Enable**
- RetryCount: **6**
- RetryInterval: **1**
- PDORemovedPeriod: **130**

7. Select the MPIO Policy **Round Robin with Subset**.
8. Change the registry values:

```
HKLM\SYSTEM\CurrentControlSet\Services\mpio\Parameters\PathRecoveryInter
val DWORD -> 30

HKLM\SYSTEM\CurrentControlSet\Services\mpio \Parameters\
UseCustomPathRecoveryInterval  DWORD-> 1
```

9. Reboot the host.

Validate NVMe/FC

Verify that the NVMe subsystems have been discovered and the ONTAP namespaces are correct for the NVMe-oF configuration.

Steps

1. Verify that "Port Type" is `FC+NVMe`:

```
listhba
```

Show example

```
Port WWN        : 10:00:00:10:9b:1b:97:65
Node WWN        : 20:00:00:10:9b:1b:97:65
Fabric Name     : 10:00:c4:f5:7c:a5:32:e0
Flags           : 8000e300
Host Name       : INTEROP-57-159
Mfg             : Emulex Corporation
Serial No.      : FC71367217
Port Number     : 0
Mode            : Initiator
PCI Bus Number  : 94
PCI Function    : 0
Port Type       : FC+NVMe
Model           : LPe32002-M2

Port WWN        : 10:00:00:10:9b:1b:97:66
Node WWN        : 20:00:00:10:9b:1b:97:66
Fabric Name     : 10:00:c4:f5:7c:a5:32:e0
Flags           : 8000e300
Host Name       : INTEROP-57-159
Mfg             : Emulex Corporation
Serial No.      : FC71367217
Port Number     : 1
Mode            : Initiator
PCI Bus Number  : 94
PCI Function    : 1
Port Type       : FC+NVMe
Model           : LPe32002-M2
```

2. Verify that the NVMe/FC subsystems have been discovered:
   - `nvme-list`

Show example

```
NVMe Qualified Name      :  nqn.1992-
08.com.netapp:sn.a3b74c32db2911eab229d039ea141105:subsystem.win_n
vme_interop-57-159
Port WWN                 :  20:09:d0:39:ea:14:11:04
Node WWN                 :  20:05:d0:39:ea:14:11:04
Controller ID            :  0x0180
Model Number             :  NetApp ONTAP Controller
Serial Number            :  81CGZBPU5T/uAAAAAAAB
Firmware Version         :  FFFFFFFF
Total Capacity           :  Not Available
Unallocated Capacity     :  Not Available

NVMe Qualified Name      :  nqn.1992-
08.com.netapp:sn.a3b74c32db2911eab229d039ea141105:subsystem.win_n
vme_interop-57-159
Port WWN                 :  20:06:d0:39:ea:14:11:04
Node WWN                 :  20:05:d0:39:ea:14:11:04
Controller ID            :  0x0181
Model Number             :  NetApp ONTAP Controller
Serial Number            :  81CGZBPU5T/uAAAAAAAB
Firmware Version         :  FFFFFFFF
Total Capacity           :  Not Available
Unallocated Capacity     :  Not Available
Note: At present Namespace Management is not supported by NetApp
Arrays.
```

- `nvme-list`

Show example

```
NVMe Qualified Name     :  nqn.1992-
08.com.netapp:sn.a3b74c32db2911eab229d039ea141105:subsystem.win_n
vme_interop-57-159
Port WWN                :  20:07:d0:39:ea:14:11:04
Node WWN                :  20:05:d0:39:ea:14:11:04
Controller ID           :  0x0140
Model Number            :  NetApp ONTAP Controller
Serial Number           :  81CGZBPU5T/uAAAAAAAB
Firmware Version        :  FFFFFFFF
Total Capacity          :  Not Available
Unallocated Capacity    :  Not Available

NVMe Qualified Name     :  nqn.1992-
08.com.netapp:sn.a3b74c32db2911eab229d039ea141105:subsystem.win_n
vme_interop-57-159
Port WWN                :  20:08:d0:39:ea:14:11:04
Node WWN                :  20:05:d0:39:ea:14:11:04
Controller ID           :  0x0141
Model Number            :  NetApp ONTAP Controller
Serial Number           :  81CGZBPU5T/uAAAAAAAB
Firmware Version        :  FFFFFFFF
Total Capacity          :  Not Available
Unallocated Capacity    :  Not Available

Note: At present Namespace Management is not supported by NetApp
Arrays.
```

3. Verify that the namespaces have been created:

```
nvme-list-ns
```

Show example

```
Active Namespaces (attached to controller 0x0141):

                                        SCSI           SCSI
SCSI
   NSID           DeviceName         Bus Number    Target Number
OS LUN
-----------  --------------------  ------------  ---------------
---------
0x00000001   \\.\PHYSICALDRIVE9         0               1
0
0x00000002   \\.\PHYSICALDRIVE10        0               1
1
0x00000003   \\.\PHYSICALDRIVE11        0               1
2
0x00000004   \\.\PHYSICALDRIVE12        0               1
3
0x00000005   \\.\PHYSICALDRIVE13        0               1
4
0x00000006   \\.\PHYSICALDRIVE14        0               1
5
0x00000007   \\.\PHYSICALDRIVE15        0               1
6
0x00000008   \\.\PHYSICALDRIVE16        0               1
7
```

NVMe/FC Host Configuration for Windows Server 2019 with ONTAP

You can configure NVMe over Fibre Channel (NVMe/FC) on hosts running Windows Server 2019 for operation with ONTAP LUNs..

About this task

You can use the following support with the NVMe/FC host configuration for Windows 2019. You should also review the known limitations before starting the configuration process.

> You can use the configuration settings provided in this procedure to configure cloud clients connected to Cloud Volumes ONTAP and Amazon FSx for ONTAP.

- Support available:

  Beginning with ONTAP 9.7, NVMe/FC is supported for Windows Server 2019.

  For a list of supported FC adapters and controllers, see the Hardware Universe. For the current list of supported configurations and versions, see the Interoperability Matrix Tool.

- Known limitations:

  Windows Failover Cluster isn't supported with NVMe/FC because ONTAP doesn't currently support persistent reservations with NVMe/FC.

> Broadcom ships an external driver for Windows NVMe/FC that is a translational SCSI NVMe driver and not a true NVMe/FC driver. The translational overhead doesn't necessarily impact performance, but it does negate the performance benefits of NVMe/FC. As a result, NVMe/FC and FCP performance is the same on Windows servers, unlike other operating systems such as Linux, where NVMe/FC performance is significantly better than that of FCP.

Enable NVMe/FC

Enable FC/NVMe on the Windows initiator host.

Steps

1. Install the Emulex HBA Manager utility on the Windows host.
2. On each of the HBA initiator ports, set the following HBA driver parameters:
   - EnableNVMe = 1
   - NVMEMode = 0
3. Reboot the host.

Configure the Broadcom FC adapter

The Broadcom initiator can serve both NVMe/FC and FCP traffic through the same 32G FC adapter ports. For FCP and FC/NVMe, you should use the Microsft device-specific module (DSM) as the Microsoft Multipath I/O (MPIO) option.

A `hostnqn` is associated with each host bus adapter (HBA) port for the Broadcom adapter with FC/NVMe in a Windows environment. The `hostnqn` is formatted as shown in the following example:

```
nqn.2017-01.com.broadcom:ecd:nvmf:fc:100000109b1b9765
nqn.2017-01.com.broadcom:ecd:nvmf:fc:100000109b1b9766
```

Enable MPIO for NVMe devices

You need to enable MPIO for NVMe devices to complete the NVMe configuration on the Windows host.

Steps

1. Install Windows Host Utility Kit 7.1 to set the driver parameters that are common to both FC and NVMe.
2. Open the MPIO properties.
3. From the **Discover Multi-Paths** tab, add the device ID listed for NVMe.

   MPIO becomes aware of the NVMe devices, which are visible under disk management.

4. Open **Disk Management** and go to **Disk Properties**.
5. From the **MPIO** tab, select **Details**.

6. Set the following Microsoft DSM settings:
   ◦ PathVerifiedPeriod: **10**
   ◦ PathVerifyEnabled: **Enable**
   ◦ RetryCount: **6**
   ◦ RetryInterval: **1**
   ◦ PDORemovedPeriod: **130**
7. Select the MPIO Policy **Round Robin with Subset**.
8. Change the registry values:

```
HKLM\SYSTEM\CurrentControlSet\Services\mpio\Parameters\PathRecoveryInter
val DWORD -> 30

HKLM\SYSTEM\CurrentControlSet\Services\mpio \Parameters\
UseCustomPathRecoveryInterval  DWORD-> 1
```

9. Reboot the host.

Validate NVMe/FC

Verify that the NVMe subsystems have been discovered and the ONTAP namespaces are correct for the NVMe-oF configuration.

Steps

1. Verify that "Port Type" is `FC+NVMe`:

   ```
   listhba
   ```

Show example

```
Port WWN         : 10:00:00:10:9b:1b:97:65
Node WWN         : 20:00:00:10:9b:1b:97:65
Fabric Name      : 10:00:c4:f5:7c:a5:32:e0
Flags            : 8000e300
Host Name        : INTEROP-57-159
Mfg              : Emulex Corporation
Serial No.       : FC71367217
Port Number      : 0
Mode             : Initiator
PCI Bus Number   : 94
PCI Function     : 0
Port Type        : FC+NVMe
Model            : LPe32002-M2

Port WWN         : 10:00:00:10:9b:1b:97:66
Node WWN         : 20:00:00:10:9b:1b:97:66
Fabric Name      : 10:00:c4:f5:7c:a5:32:e0
Flags            : 8000e300
Host Name        : INTEROP-57-159
Mfg              : Emulex Corporation
Serial No.       : FC71367217
Port Number      : 1
Mode             : Initiator
PCI Bus Number   : 94
PCI Function     : 1
Port Type        : FC+NVMe
Model            : LPe32002-M2
```

2. Verify that the NVMe/FC subsystems have been discovered:
   - `nvme-list`

Show example

```
NVMe Qualified Name     :  nqn.1992-
08.com.netapp:sn.a3b74c32db2911eab229d039ea141105:subsystem.win_n
vme_interop-57-159
Port WWN                :  20:09:d0:39:ea:14:11:04
Node WWN                :  20:05:d0:39:ea:14:11:04
Controller ID           :  0x0180
Model Number            :  NetApp ONTAP Controller
Serial Number           :  81CGZBPU5T/uAAAAAAAB
Firmware Version        :  FFFFFFFF
Total Capacity          :  Not Available
Unallocated Capacity    :  Not Available

NVMe Qualified Name     :  nqn.1992-
08.com.netapp:sn.a3b74c32db2911eab229d039ea141105:subsystem.win_n
vme_interop-57-159
Port WWN                :  20:06:d0:39:ea:14:11:04
Node WWN                :  20:05:d0:39:ea:14:11:04
Controller ID           :  0x0181
Model Number            :  NetApp ONTAP Controller
Serial Number           :  81CGZBPU5T/uAAAAAAAB
Firmware Version        :  FFFFFFFF
Total Capacity          :  Not Available
Unallocated Capacity    :  Not Available
Note: At present Namespace Management is not supported by NetApp
Arrays.
```

- `nvme-list`

Show example

```
NVMe Qualified Name     :  nqn.1992-
08.com.netapp:sn.a3b74c32db2911eab229d039ea141105:subsystem.win_n
vme_interop-57-159
Port WWN                :  20:07:d0:39:ea:14:11:04
Node WWN                :  20:05:d0:39:ea:14:11:04
Controller ID           :  0x0140
Model Number            :  NetApp ONTAP Controller
Serial Number           :  81CGZBPU5T/uAAAAAAAB
Firmware Version        :  FFFFFFFF
Total Capacity          :  Not Available
Unallocated Capacity    :  Not Available

NVMe Qualified Name     :  nqn.1992-
08.com.netapp:sn.a3b74c32db2911eab229d039ea141105:subsystem.win_n
vme_interop-57-159
Port WWN                :  20:08:d0:39:ea:14:11:04
Node WWN                :  20:05:d0:39:ea:14:11:04
Controller ID           :  0x0141
Model Number            :  NetApp ONTAP Controller
Serial Number           :  81CGZBPU5T/uAAAAAAAB
Firmware Version        :  FFFFFFFF
Total Capacity          :  Not Available
Unallocated Capacity    :  Not Available

Note: At present Namespace Management is not supported by NetApp
Arrays.
```

3. Verify that the namespaces have been created:

```
nvme-list-ns
```

Show example

```
Active Namespaces (attached to controller 0x0141):

                                         SCSI           SCSI
SCSI
   NSID           DeviceName        Bus Number    Target Number
OS LUN
-----------  --------------------  ------------  ----------------
---------
0x00000001   \\.\PHYSICALDRIVE9         0               1
0
0x00000002   \\.\PHYSICALDRIVE10        0               1
1
0x00000003   \\.\PHYSICALDRIVE11        0               1
2
0x00000004   \\.\PHYSICALDRIVE12        0               1
3
0x00000005   \\.\PHYSICALDRIVE13        0               1
4
0x00000006   \\.\PHYSICALDRIVE14        0               1
5
0x00000007   \\.\PHYSICALDRIVE15        0               1
6
0x00000008   \\.\PHYSICALDRIVE16        0               1
7
```

Configure Windows Server 2016 with NVMe/FC for ONTAP

You can configure NVMe over Fibre Channel (NVMe/FC) on hosts running Windows Server 2016 for operation with ONTAP LUNs.

About this task

You can use the following support with the NVMe/FC host configuration for Windows 2016. You should also review the known limitations before starting the configuration process.

> You can use the configuration settings provided in this procedure to configure cloud clients connected to Cloud Volumes ONTAP and Amazon FSx for ONTAP.

- Support available:

  Beginning with ONTAP 9.7, NVMe/FC is supported for Windows Server 2016.

  For a list of supported FC adapters and controllers, see the Hardware Universe. For the current list of supported configurations and versions, see the Interoperability Matrix Tool.

- Known limitations:

  Windows Failover Cluster isn't supported with NVMe/FC because ONTAP doesn't currently support persistent reservations with NVMe/FC.

> Broadcom ships an external driver for Windows NVMe/FC that is a translational SCSI NVMe driver and not a true NVMe/FC driver. The translational overhead doesn't necessarily impact performance, but it does negate the performance benefits of NVMe/FC. As a result, NVMe/FC and FCP performance is the same on Windows servers, unlike other operating systems such as Linux, where NVMe/FC performance is significantly better than that of FCP.

Enable NVMe/FC

Enable FC/NVMe on the Windows initiator host.

Steps

1. Install the Emulex HBA Manager utility on the Windows host.
2. On each of the HBA initiator ports, set the following HBA driver parameters:
   - EnableNVMe = 1
   - NVMEMode = 0
3. Reboot the host.

Configure the Broadcom FC adapter

The Broadcom initiator can serve both NVMe/FC and FCP traffic through the same 32G FC adapter ports. For FCP and FC/NVMe, you should use the Microsft device-specific module (DSM) as the Microsoft Multipath I/O (MPIO) option.

A `hostnqn` is associated with each host bus adapter (HBA) port for the Broadcom adapter with FC/NVMe in a Windows environment. The `hostnqn` is formatted as shown in the following example:

```
nqn.2017-01.com.broadcom:ecd:nvmf:fc:100000109b1b9765
nqn.2017-01.com.broadcom:ecd:nvmf:fc:100000109b1b9766
```

Enable MPIO for NVMe devices

You need to enable MPIO for NVMe devices to complete the NVMe configuration on the Windows host.

Steps

1. Install Windows Host Utility Kit 7.1 to set the driver parameters that are common to both FC and NVMe.
2. Open the MPIO properties.
3. From the **Discover Multi-Paths** tab, add the device ID listed for NVMe.

   MPIO becomes aware of the NVMe devices, which are visible under disk management.

4. Open **Disk Management** and go to **Disk Properties**.
5. From the **MPIO** tab, select **Details**.

6. Set the following Microsoft DSM settings:
   - PathVerifiedPeriod: **10**
   - PathVerifyEnabled: **Enable**
   - RetryCount: **6**
   - RetryInterval: **1**
   - PDORemovedPeriod: **130**
7. Select the MPIO Policy **Round Robin with Subset**.
8. Change the registry values:

```
HKLM\SYSTEM\CurrentControlSet\Services\mpio\Parameters\PathRecoveryInter
val DWORD -> 30

HKLM\SYSTEM\CurrentControlSet\Services\mpio \Parameters\
UseCustomPathRecoveryInterval  DWORD-> 1
```

9. Reboot the host.

Validate NVMe/FC

Verify that the NVMe subsystems have been discovered and the ONTAP namespaces are correct for the NVMe-oF configuration.

Steps

1. Verify that "Port Type" is `FC+NVMe`:

   ```
   listhba
   ```

Show example

```
Port WWN        : 10:00:00:10:9b:1b:97:65
Node WWN        : 20:00:00:10:9b:1b:97:65
Fabric Name     : 10:00:c4:f5:7c:a5:32:e0
Flags           : 8000e300
Host Name       : INTEROP-57-159
Mfg             : Emulex Corporation
Serial No.      : FC71367217
Port Number     : 0
Mode            : Initiator
PCI Bus Number  : 94
PCI Function    : 0
Port Type       : FC+NVMe
Model           : LPe32002-M2

Port WWN        : 10:00:00:10:9b:1b:97:66
Node WWN        : 20:00:00:10:9b:1b:97:66
Fabric Name     : 10:00:c4:f5:7c:a5:32:e0
Flags           : 8000e300
Host Name       : INTEROP-57-159
Mfg             : Emulex Corporation
Serial No.      : FC71367217
Port Number     : 1
Mode            : Initiator
PCI Bus Number  : 94
PCI Function    : 1
Port Type       : FC+NVMe
Model           : LPe32002-M2
```

2. Verify that the NVMe/FC subsystems have been discovered:
   - `nvme-list`

Show example

```
NVMe Qualified Name     :  nqn.1992-
08.com.netapp:sn.a3b74c32db2911eab229d039ea141105:subsystem.win_n
vme_interop-57-159
Port WWN                :  20:09:d0:39:ea:14:11:04
Node WWN                :  20:05:d0:39:ea:14:11:04
Controller ID           :  0x0180
Model Number            :  NetApp ONTAP Controller
Serial Number           :  81CGZBPU5T/uAAAAAAAB
Firmware Version        :  FFFFFFFF
Total Capacity          :  Not Available
Unallocated Capacity    :  Not Available

NVMe Qualified Name     :  nqn.1992-
08.com.netapp:sn.a3b74c32db2911eab229d039ea141105:subsystem.win_n
vme_interop-57-159
Port WWN                :  20:06:d0:39:ea:14:11:04
Node WWN                :  20:05:d0:39:ea:14:11:04
Controller ID           :  0x0181
Model Number            :  NetApp ONTAP Controller
Serial Number           :  81CGZBPU5T/uAAAAAAAB
Firmware Version        :  FFFFFFFF
Total Capacity          :  Not Available
Unallocated Capacity    :  Not Available
Note: At present Namespace Management is not supported by NetApp
Arrays.
```

◦ `nvme-list`

Show example

```
NVMe Qualified Name     :  nqn.1992-
08.com.netapp:sn.a3b74c32db2911eab229d039ea141105:subsystem.win_n
vme_interop-57-159
Port WWN                :  20:07:d0:39:ea:14:11:04
Node WWN                :  20:05:d0:39:ea:14:11:04
Controller ID           :  0x0140
Model Number            :  NetApp ONTAP Controller
Serial Number           :  81CGZBPU5T/uAAAAAAAB
Firmware Version        :  FFFFFFFF
Total Capacity          :  Not Available
Unallocated Capacity    :  Not Available

NVMe Qualified Name     :  nqn.1992-
08.com.netapp:sn.a3b74c32db2911eab229d039ea141105:subsystem.win_n
vme_interop-57-159
Port WWN                :  20:08:d0:39:ea:14:11:04
Node WWN                :  20:05:d0:39:ea:14:11:04
Controller ID           :  0x0141
Model Number            :  NetApp ONTAP Controller
Serial Number           :  81CGZBPU5T/uAAAAAAAB
Firmware Version        :  FFFFFFFF
Total Capacity          :  Not Available
Unallocated Capacity    :  Not Available

Note: At present Namespace Management is not supported by NetApp
Arrays.
```

3. Verify that the namespaces have been created:

```
nvme-list-ns
```

Show example

```
Active Namespaces (attached to controller 0x0141):

                                        SCSI          SCSI
SCSI
   NSID           DeviceName          Bus Number    Target Number
OS LUN
-----------  --------------------  ------------  ---------------
---------
0x00000001   \\.\PHYSICALDRIVE9        0              1
0
0x00000002   \\.\PHYSICALDRIVE10       0              1
1
0x00000003   \\.\PHYSICALDRIVE11       0              1
2
0x00000004   \\.\PHYSICALDRIVE12       0              1
3
0x00000005   \\.\PHYSICALDRIVE13       0              1
4
0x00000006   \\.\PHYSICALDRIVE14       0              1
5
0x00000007   \\.\PHYSICALDRIVE15       0              1
6
0x00000008   \\.\PHYSICALDRIVE16       0              1
7
```

Configure Windows Server 2012 R2 with NVMe/FC for ONTAP

You can configure NVMe over Fibre Channel (NVMe/FC) on hosts running Windows Server 2012 R2 for operation with ONTAP LUNs.

About this task

You can use the following support with the NVMe/FC host configuration for Windows 2012 R2. You should also review the known limitations before starting the configuration process.

> ⓘ You can use the configuration settings provided in this procedure to configure cloud clients connected to Cloud Volumes ONTAP and Amazon FSx for ONTAP.

- Support available:

  Beginning with ONTAP 9.7, NVMe/FC is supported for Windows Server 2012 R2.

  For a list of supported FC adapters and controllers, see the Hardware Universe. For the current list of supported configurations and versions, see the Interoperability Matrix Tool.

* Known limitations:

  Windows Failover Cluster isn't supported with NVMe/FC because ONTAP doesn't currently support persistent reservations with NVMe/FC.

> Broadcom ships an external driver for Windows NVMe/FC that is a translational SCSI NVMe driver and not a true NVMe/FC driver. The translational overhead doesn't necessarily impact performance, but it does negate the performance benefits of NVMe/FC. As a result, NVMe/FC and FCP performance is the same on Windows servers, unlike other operating systems such as Linux, where NVMe/FC performance is significantly better than that of FCP.

Enable NVMe/FC

Enable FC/NVMe on the Windows initiator host.

Steps

1. Install the Emulex HBA Manager utility on the Windows host.
2. On each of the HBA initiator ports, set the following HBA driver parameters:
   * EnableNVMe = 1
   * NVMEMode = 0
3. Reboot the host.

Configure the Broadcom FC adapter

The Broadcom initiator can serve both NVMe/FC and FCP traffic through the same 32G FC adapter ports. For FCP and FC/NVMe, you should use the Microsft device-specific module (DSM) as the Microsoft Multipath I/O (MPIO) option.

A `hostnqn` is associated with each host bus adapter (HBA) port for the Broadcom adapter with FC/NVMe in a Windows environment. The `hostnqn` is formatted as shown in the following example:

```
nqn.2017-01.com.broadcom:ecd:nvmf:fc:100000109b1b9765
nqn.2017-01.com.broadcom:ecd:nvmf:fc:100000109b1b9766
```

Enable MPIO for NVMe devices

You need to enable MPIO for NVMe devices to complete the NVMe configuration on the Windows host.

Steps

1. Install Windows Host Utility Kit 7.1 to set the driver parameters that are common to both FC and NVMe.
2. Open the MPIO properties.
3. From the **Discover Multi-Paths** tab, add the device ID listed for NVMe.

   MPIO becomes aware of the NVMe devices, which are visible under disk management.
4. Open **Disk Management** and go to **Disk Properties**.
5. From the **MPIO** tab, select **Details**.

6. Set the following Microsoft DSM settings:
   - PathVerifiedPeriod: **10**
   - PathVerifyEnabled: **Enable**
   - RetryCount: **6**
   - RetryInterval: **1**
   - PDORemovedPeriod: **130**
7. Select the MPIO Policy **Round Robin with Subset**.
8. Change the registry values:

```
HKLM\SYSTEM\CurrentControlSet\Services\mpio\Parameters\PathRecoveryInter
val DWORD -> 30

HKLM\SYSTEM\CurrentControlSet\Services\mpio \Parameters\
UseCustomPathRecoveryInterval  DWORD-> 1
```

9. Reboot the host.

Validate NVMe/FC

Verify that the NVMe subsystems have been discovered and the ONTAP namespaces are correct for the NVMe-oF configuration.

Steps

1. Verify that "Port Type" is `FC+NVMe`:

   ```
   listhba
   ```

Show example

```
Port WWN         : 10:00:00:10:9b:1b:97:65
Node WWN         : 20:00:00:10:9b:1b:97:65
Fabric Name      : 10:00:c4:f5:7c:a5:32:e0
Flags            : 8000e300
Host Name        : INTEROP-57-159
Mfg              : Emulex Corporation
Serial No.       : FC71367217
Port Number      : 0
Mode             : Initiator
PCI Bus Number   : 94
PCI Function     : 0
Port Type        : FC+NVMe
Model            : LPe32002-M2

Port WWN         : 10:00:00:10:9b:1b:97:66
Node WWN         : 20:00:00:10:9b:1b:97:66
Fabric Name      : 10:00:c4:f5:7c:a5:32:e0
Flags            : 8000e300
Host Name        : INTEROP-57-159
Mfg              : Emulex Corporation
Serial No.       : FC71367217
Port Number      : 1
Mode             : Initiator
PCI Bus Number   : 94
PCI Function     : 1
Port Type        : FC+NVMe
Model            : LPe32002-M2
```

2. Verify that the NVMe/FC subsystems have been discovered:

   - `nvme-list`

Show example

```
NVMe Qualified Name      :  nqn.1992-
08.com.netapp:sn.a3b74c32db2911eab229d039ea141105:subsystem.win_n
vme_interop-57-159
Port WWN                 :  20:09:d0:39:ea:14:11:04
Node WWN                 :  20:05:d0:39:ea:14:11:04
Controller ID            :  0x0180
Model Number             :  NetApp ONTAP Controller
Serial Number            :  81CGZBPU5T/uAAAAAAAB
Firmware Version         :  FFFFFFFF
Total Capacity           :  Not Available
Unallocated Capacity     :  Not Available

NVMe Qualified Name      :  nqn.1992-
08.com.netapp:sn.a3b74c32db2911eab229d039ea141105:subsystem.win_n
vme_interop-57-159
Port WWN                 :  20:06:d0:39:ea:14:11:04
Node WWN                 :  20:05:d0:39:ea:14:11:04
Controller ID            :  0x0181
Model Number             :  NetApp ONTAP Controller
Serial Number            :  81CGZBPU5T/uAAAAAAAB
Firmware Version         :  FFFFFFFF
Total Capacity           :  Not Available
Unallocated Capacity     :  Not Available
Note: At present Namespace Management is not supported by NetApp
Arrays.
```

◦ `nvme-list`

Show example

```
NVMe Qualified Name      :  nqn.1992-
08.com.netapp:sn.a3b74c32db2911eab229d039ea141105:subsystem.win_n
vme_interop-57-159
Port WWN                 :  20:07:d0:39:ea:14:11:04
Node WWN                 :  20:05:d0:39:ea:14:11:04
Controller ID            :  0x0140
Model Number             :  NetApp ONTAP Controller
Serial Number            :  81CGZBPU5T/uAAAAAAAB
Firmware Version         :  FFFFFFFF
Total Capacity           :  Not Available
Unallocated Capacity     :  Not Available

NVMe Qualified Name      :  nqn.1992-
08.com.netapp:sn.a3b74c32db2911eab229d039ea141105:subsystem.win_n
vme_interop-57-159
Port WWN                 :  20:08:d0:39:ea:14:11:04
Node WWN                 :  20:05:d0:39:ea:14:11:04
Controller ID            :  0x0141
Model Number             :  NetApp ONTAP Controller
Serial Number            :  81CGZBPU5T/uAAAAAAAB
Firmware Version         :  FFFFFFFF
Total Capacity           :  Not Available
Unallocated Capacity     :  Not Available

Note: At present Namespace Management is not supported by NetApp
Arrays.
```

3. Verify that the namespaces have been created:

```
nvme-list-ns
```

Show example

```
Active Namespaces (attached to controller 0x0141):

                                              SCSI           SCSI
SCSI
   NSID           DeviceName        Bus Number    Target Number
OS LUN
-----------  --------------------  ------------  ---------------
---------
0x00000001   \\.\PHYSICALDRIVE9         0               1
0
0x00000002   \\.\PHYSICALDRIVE10        0               1
1
0x00000003   \\.\PHYSICALDRIVE11        0               1
2
0x00000004   \\.\PHYSICALDRIVE12        0               1
3
0x00000005   \\.\PHYSICALDRIVE13        0               1
4
0x00000006   \\.\PHYSICALDRIVE14        0               1
5
0x00000007   \\.\PHYSICALDRIVE15        0               1
6
0x00000008   \\.\PHYSICALDRIVE16        0               1
7
```

Troubleshoot

Before troubleshooting any NVMe-oF failures for RHEL, OL, and SUSE Linux Enterprise Server hosts, verify that you are running a configuration that is compliant to the Interoperability Matrix Tool (IMT) specifications and then proceed with the next steps to debug any host side issues.

> The troubleshooting instructions are not applicable for AIX, Windows, and ESXi hosts.

Enable verbose logging

If you have an issue with your configuration, verbose logging can provide essential information for troubleshooting.

The procedure to set verbose logging for Qlogic (Qla2xxx) is different from the procedure to set LPFC verbose logging.

LPFC

Set the lpfc driver for NVMe/FC.

Steps

1. Set the `lpfc_log_verbose` driver setting to any of the following values to log NVMe/FC events.

   ```
   #define LOG_NVME 0x00100000 /* NVME general events. */
   #define LOG_NVME_DISC 0x00200000 /* NVME Discovery/Connect events.
   */
   #define LOG_NVME_ABTS 0x00400000 /* NVME ABTS events. */
   #define LOG_NVME_IOERR 0x00800000 /* NVME IO Error events. */
   ```

2. After setting the values, run the `dracut-f` command and reboot the host.
3. Verify the settings.

   ```
   # cat /etc/modprobe.d/lpfc.conf options lpfc
   lpfc_log_verbose=0xf00083

   # cat /sys/module/lpfc/parameters/lpfc_log_verbose 15728771
   ```

Qla2xxx

There is no specific qla2xxx logging for NVMe/FC similar to that for the `lpfc` driver. Instead, set the general qla2xxx logging level.

Steps

1. Append the `ql2xextended_error_logging=0x1e400000` value to the corresponding `modprobe qla2xxx conf` file.
2. Execute the `dracut -f` command and then reboot the host.
3. After reboot, verify that the verbose logging has been enabled:

   ```
   # cat /etc/modprobe.d/qla2xxx.conf
   ```

   Example output:

   ```
   options qla2xxx ql2xnvmeenable=1
   ql2xextended_error_logging=0x1e400000
   # cat /sys/module/qla2xxx/parameters/ql2xextended_error_logging
   507510784
   ```

Common nvme-cli errors and workarounds

The errors displayed by `nvme-cli` during `nvme discover`, `nvme connect`, or `nvme connect-all` operations and the workarounds are shown in the following table:

Error message	Probable cause	Workaround
`Failed to write to /dev/nvme-fabrics: Invalid argument`	Incorrect syntax	Verify that you are using the correct syntax for the `nvme discover`, `nvme connect`, and `nvme connect-all` commands.

<table>
<tr><th>Error message</th><th>Probable cause</th><th>Workaround</th></tr>
<tr><td><code>Failed to write to /dev/nvme-fabrics: No such file or directory</code></td><td>Multiple issues can trigger this, for example, providing wrong arguments to the NVMe commands is one of the common causes.</td><td>

Verify that you have passed the correct arguments (such as, correct WWNN string, WWPN string, and more) to the commands.
If the arguments are correct, but you still see this error, check whether the <code>/sys/class/scsi_host/host*/nvme_info</code> command output is correct, the NVMe initiator is displayed as <code>Enabled</code>, and the NVMe/FC target LIFs are correctly displayed under the remote ports sections. Example:
<pre>
cat
/sys/class/scsi_host/host*/nvme_info
NVME Initiator Enabled
NVME LPORT lpfc0 WWPN
x10000090fae0ec9d WWNN
x20000090fae0ec9d DID x012000 ONLINE
NVME RPORT WWPN x200b00a098c80f09
WWNN x200a00a098c80f09 DID x010601
TARGET DISCSRVC ONLINE
NVME Statistics
LS: Xmt 0000000000000006 Cmpl
0000000000000006
FCP: Rd 0000000000000071 Wr
0000000000000005 IO 0000000000000031
Cmpl 00000000000000a6 Outstanding
0000000000000001
NVME Initiator Enabled
NVME LPORT lpfc1 WWPN
x10000090fae0ec9e WWNN
x20000090fae0ec9e DID x012400 ONLINE
NVME RPORT WWPN x200900a098c80f09
WWNN x200800a098c80f09 DID x010301
TARGET DISCSRVC ONLINE
NVME Statistics
LS: Xmt 0000000000000006 Cmpl
0000000000000006
FCP: Rd 0000000000000073 Wr
0000000000000005 IO 0000000000000031
Cmpl 00000000000000a8 Outstanding
0000000000000001
</pre>
If the <code>nvme_info</code> command output doesn’t display the target LIFs as shown in the example above, check the <code>/var/log/messages</code> and <code>dmesg</code> command outputs for any suspicious NVMe/FC failures, and report or fix accordingly.

</td></tr>
</table>

Error message	Probable cause	Workaround
`No discovery log entries to fetch`	Generally observed when the `/etc/nvme/hostnqn` string has not been added to the corresponding subsystem on the NetApp array or an incorrect `hostnqn` string has been added to the respective subsystem.	Verify that the exact `/etc/nvme/hostnqn` string is added to the corresponding subsystem on the NetApp array (verify using the `vserver nvme subsystem host show` command).
`Failed to write to /dev/nvme-fabrics: Operation already in progress`	Observed when the controller associations or specified operation is already created or in the process of being created. This could happen as part of the auto-connect scripts installed above.	None. Try running the `nvme discover` command again after some time. For `nvme connect` and `connect-all`, run the `nvme list` command to verify that the namespace devices are already created and displayed on the host.

When to contact technical support

If you are still facing issues, collect the following files and command outputs and contact NetApp support for further triage:

```
cat /sys/class/scsi_host/host*/nvme_info
/var/log/messages
dmesg
nvme discover output as in:
nvme discover --transport=fc --traddr=nn-0x200a00a098c80f09:pn
-0x200b00a098c80f09 --host-traddr=nn-0x20000090fae0ec9d:pn
-0x10000090fae0ec9d
nvme list
nvme list-subsys /dev/nvmeXnY
```