



Proxmox

ONTAP SAN Host Utilities

NetApp
January 30, 2026

This PDF was generated from <https://docs.netapp.com/us-en/ontap-sanhost/nvme-proxmox-supported-features.html> on January 30, 2026. Always check docs.netapp.com for the latest.

Table of Contents

- Proxmox 1
 - Learn about ONTAP support and features for Proxmox hosts 1
 - What's next 1
 - Configure Proxmox VE 9.x for NVMe-oF with ONTAP storage 1
 - Step 1: Install Proxmox VE and NVMe software and verify your configuration 2
 - Step 2: Configure NVMe/FC and NVMe/TCP 3
 - Step 3: Optionally, enable 1MB I/O for NVMe/FC 9
 - Step 4: Verify NVMe boot services 10
 - Step 5: Verify the multipathing configuration 12
 - Step 6: Review the known issues 16
 - Configure Proxmox VE 8.x for NVMe-oF with ONTAP storage 17
 - Step 1: Install Proxmox VE and NVMe software and verify your configuration 17
 - Step 2: Configure NVMe/FC and NVMe/TCP 18
 - Step 3: Optionally, enable 1MB I/O for NVMe/FC 27
 - Step 4: Verify NVMe boot services 28
 - Step 5: Verify the multipathing configuration 29
 - Step 6: Review the known issues 33

Proxmox

Learn about ONTAP support and features for Proxmox hosts

Features supported for host configuration with NVMe over Fabrics (NVMe-oF) varies based on your version of ONTAP and Proxmox.

ONTAP feature	Proxmox host version	ONTAP version
NVMe/TCP is a fully supported enterprise feature	9.0	9.10.1 or later
Provides ONTAP details for both NVMe/FC and NVMe/TCP namespaces using the native <code>nvme-cli</code> package.	8.0	9.10.1 or later
NVMe and SCSI traffic is supported on the same host using NVMe multipath for NVMe-oF namespaces and dm-multipath for SCSI LUNs	8.0	9.4 or later

ONTAP supports the following SAN host features regardless of the ONTAP version running on your system setup.

Feature	Proxmox host version
The <code>nvme-cli</code> package includes auto-connect scripts removing the need for third-party scripts	9.0
The native udev rule in the <code>nvme-cli</code> package provides round-robin load balancing for NVMe multipathing	9.0
Native NVMe multipathing is enabled by default	8.0



For details on supported configurations, see the [Interoperability Matrix Tool](#).

What's next

If your Proxmox VE version is ..	Learn about ..
9 series	Configuring NVMe for Proxmox VE 9.x
8 series	Configuring NVMe for Proxmox VE 8.x

Related information

[Learn about managing NVMe protocols](#)

Configure Proxmox VE 9.x for NVMe-oF with ONTAP storage

The Proxmox VE 9.x host supports the NVMe over Fibre Channel (NVMe/FC) and NVMe

over TCP (NVMe/TCP) protocols with Asymmetric Namespace Access (ANA). ANA provides multipathing functionality equivalent to asymmetric logical unit access (ALUA) in iSCSI and FCP environments.

Learn how to configure NVMe over Fabrics (NVMe-oF) hosts for Proxmox VE 9.x. For more support and feature information, see [ONTAP support and features](#).

NVMe-oF with Proxmox VE 9.x has the following known limitation:

- The SAN boot configuration for NVMe-FC is not supported.

Step 1: Install Proxmox VE and NVMe software and verify your configuration

To configure your host for NVMe-oF you need to install the host and NVMe software packages, enable multipathing, and verify your host NQN configuration.

Steps

1. Install Proxmox VE 9.x on the server. After the installation is complete, verify that you are running the required Proxmox VE 9.x kernel:

```
uname -r
```

Example Proxmox VE 9.x kernel version:

```
6.14.8-2-pve
```

2. Install the `nvme-cli` package:

```
apt list|grep nvme-cli
```

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

```
nvme-cli/stable,now 2.13-2 amd64
```

3. Install the `libnvme` package:

```
apt list|grep libnvme
```

The following example shows an `libnvme` package version:

```
libnvme-dev/stable 1.13-2 amd64
```

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

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

The following example shows a `hostnqn` value:

```
nqn.2014-08.org.nvmexpress:uuid:39333550-3333-4753-4844-32594d4a524c
```

5. On the ONTAP system, verify that the `hostnqn` string matches the `hostnqn` string for the corresponding subsystem on the ONTAP array:

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

Show example

```
Vserver Subsystem Priority Host NQN
-----
vs_proxmox_FC_NVMeFC
    sub_176
        regular nqn.2014-08.org.nvmexpress:uuid:39333550-3333-4753-4844-32594d4a4834
        regular nqn.2014-08.org.nvmexpress:uuid:39333550-3333-4753-4844-32594d4a524c
2 entries were displayed
```



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

Step 2: Configure NVMe/FC and NVMe/TCP

Configure NVMe/FC with Broadcom/Emulex or Marvell/QLogic adapters, or configure NVMe/TCP using manual discovery and connect operations.

NVMe/FC - 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:

```
SN1700E2P  
SN1700E2P
```

b. Show the model descriptions:

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

You should see an output similar to the following example:

```
HPE SN1700E 64Gb 2p FC HBA  
HPE SN1700E 64Gb 2p FC HBA
```

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 command returns the firmware versions:

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

b. Display the inbox driver version:

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

The following example shows a driver version:

```
0:14.4.0.7
```

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
```

You should see an output similar to:

```
0x10005ced8c531948  
0x10005ced8c531949
```

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 x10005ced8c531948 WWNN x20005ced8c531948
DID x082400 ONLINE
NVME RPORT          WWPN x200ed039eac79573 WWNN x200dd039eac79573
DID x060902 TARGET DISCSRVC ONLINE
NVME RPORT          WWPN x2001d039eac79573 WWNN x2000d039eac79573
DID x060904 TARGET DISCSRVC ONLINE
```

```
NVME Statistics
LS: Xmt 0000000034 Cmpl 0000000034 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 0000000000142cfb Issue 0000000000142cfc OutIO
00000000000000001
          abort 00000005 noxri 00000000 nondlp 00000000 qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 00000005 Err 00000005
```

```
NVME Initiator Enabled
XRI Dist lpfc1 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc1 WWPN x10005ced8c531949 WWNN x20005ced8c531949
DID x082500 ONLINE
NVME RPORT          WWPN x2010d039eac79573 WWNN x200dd039eac79573
DID x062902 TARGET DISCSRVC ONLINE
NVME RPORT          WWPN x2007d039eac79573 WWNN x2000d039eac79573
DID x062904 TARGET DISCSRVC ONLINE
```

```
NVME Statistics
LS: Xmt 0000000034 Cmpl 0000000034 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 00000000000d39f1 Issue 00000000000d39f2 OutIO
00000000000000001
          abort 00000005 noxri 00000000 nondlp 00000000 qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 00000005 Err 00000005
```

NVMe/FC - Marvell/QLogic

Configure NVMe/FC for a Marvell/QLogic adapter.

Steps

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

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

The following example shows driver and firmware versions:

```
SN1700Q FW:v9.15.05 DVR:v10.02.09.400-k  
SN1700Q FW:v9.15.05 DVR:v10.02.09.400-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.

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. Check that the initiator port can get 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 192.168.165.72 -a 192.168.165.51
Discovery Log Number of Records 4, Generation counter 47
=====Discovery Log Entry 0=====
trtype: tcp
adrfam: ipv4
subtype: current discovery subsystem
treq: not specified
portid: 3
trsvcid: 8009
subnqn: nqn.1992-
08.com.netapp:sn.c770be5d934811f0b624d039eac809ba:discovery
traddr: 192.168.165.51
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.c770be5d934811f0b624d039eac809ba:discovery
traddr: 192.168.166.50
eflags: explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 2=====
trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 3
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.c770be5d934811f0b624d039eac809ba:subsystem.sub_
176
traddr: 192.168.165.51
eflags: none
sectype: none
=====Discovery Log Entry 3=====
trtype: tcp
adrfam: ipv4
```

```
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.c770be5d934811f0b624d039eac809ba:subsystem.sub_
176
traddr:  192.168.166.50
eflags:  none
sectype: none
```

2. 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 192.168.166.73 -a 192.168.166.50
nvme connect-all -t tcp -w 192.168.166.73 -a 192.168.166.51
nvme connect-all -t tcp -w 192.168.165.73 -a 192.168.165.50
nvme connect-all -t tcp -w 192.168.165.73 -a 192.168.165.51
```

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

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 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 `update-initramfs` 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 4: Verify NVMe boot services

The `nvme-fc-boot-connections.service` and `nvme-fc-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 `nvme-fc-boot-connections.service` and `nvme-fc-autoconnect.service` boot services are enabled.

Steps

1. Verify that `nvme-fc-autoconnect.service` is enabled:

```
systemctl status nvme-fc-autoconnect.service
```

Show example output

```
o nvme-autoconnect.service - Connect NVMe-oF subsystems
automatically during boot
   Loaded: loaded (/usr/lib/systemd/system/nvme-
autoconnect.service; enabled; preset: enabled)
   Active: inactive (dead) since Thu 2025-10-16 18:00:39 IST; 6
days ago
  Invocation: e146e0b2c339432aad6e0555a528872c
    Process: 1787 ExecStart=/usr/sbin/nvme connect-all
--context=autoconnect (code=exited, status=0/SUCCESS)
   Main PID: 1787 (code=exited, status=0/SUCCESS)
    Mem peak: 2.4M
       CPU: 12ms
Oct 16 18:00:39 HPE-DL365-14-176 systemd[1]: Starting nvme-
autoconnect.service - Connect NVMe-oF subsystems automatically
during boot...
Oct 16 18:00:39 HPE-DL365-14-176 systemd[1]: nvme-
autoconnect.service: Deactivated successfully.
Oct 16 18:00:39 HPE-DL365-14-176 systemd[1]: Finished nvme-
autoconnect.service - Connect NVMe-oF subsystems automatically
during boot.
```

2. Verify that nvmeboot-connections.service is enabled:

```
systemctl status nvmeboot-connections.service
```

Show example output

```
nvmeofc-boot-connections.service - Auto-connect to subsystems on FC-
NVME devices found during boot
   Loaded: loaded (/usr/lib/systemd/system/nvmeofc-boot-
connections.service; enabled; preset: enabled)
   Active: inactive (dead) since Thu 2025-10-16 18:00:35 IST; 6
days ago
  Invocation: acf73ac1ef7a402198d6ecc4d075fab0
    Process: 1173 ExecStart=/bin/sh -c echo add >
/sys/class/fc/fc_udev_device/nvme_discovery (code=exited,
status=0/SUCCESS)
   Main PID: 1173 (code=exited, status=0/SUCCESS)
  Mem peak: 2.1M
    CPU: 11ms

Oct 16 18:00:35 HPE-DL365-14-176 systemd[1]: nvmeofc-boot-
connections.service: Deactivated successfully.
Oct 16 18:00:35 HPE-DL365-14-176 systemd[1]: Finished nvmeofc-boot-
connections.service - Auto-connect to subsystems on FC-NVME devices
found during boot.
```

Step 5: Verify the multipathing configuration

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 ONTAP namespaces show up correctly on the host:
 - a. Show 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:

```
queue-depth
queue-depth
```

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/nvme2n1	/dev/ng2n1	81PqYFYq2aVAAAAAAAAB	NetApp ONTAP
Controller	0x1	17.88 GB / 171.80 GB	4 KiB + 0 B
9.17.1			

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

NVMe/FC

```
nvme list-subsys /dev/nvme3n9
```

Show example

```
nvme-subsys3 - NQN=nqn.1992-08.com.netapp:sn.94929fdb84eb11f0b624d039eac809ba:subsystem.sub_176
                hostnqn=nqn.2014-08.org.nvmexpress:uuid:39333550-3333-4753-4844-32594d4a524c
\
+- nvme1 fc traddr=nn-0x200dd039eac79573:pn-0x2010d039eac79573,host_traddr=nn-0x20005ced8c531949:pn-0x10005ced8c531949 live optimized
+- nvme3 fc traddr=nn-0x200dd039eac79573:pn-0x200ed039eac79573,host_traddr=nn-0x20005ced8c531948:pn-0x10005ced8c531948 live optimized
+- nvme5 fc traddr=nn-0x200dd039eac79573:pn-0x200fd039eac79573,host_traddr=nn-0x20005ced8c531949:pn-0x10005ced8c531949 live non-optimized
+- nvme7 fc traddr=nn-0x200dd039eac79573:pn-0x2011d039eac79573,host_traddr=nn-0x20005ced8c531948:pn-0x10005ced8c531948 live non-optimized
```

NVMe/TCP

```
nvme list-subsys /dev/nvme2n3
```

Show example

```
nvme-subsys2 - NQN=nqn.1992-08.com.netapp:sn.c770be5d934811f0b624d039eac809ba:subsystem.sub_176
                hostnqn=nqn.2014-08.org.nvmexpress:uuid:39333550-3333-4753-4844-32594d4a524c
\
+- nvme2 tcp
traddr=192.168.166.50,trsvcid=4420,host_traddr=192.168.166.73,src_addr=192.168.166.73 live optimized
+- nvme4 tcp
traddr=192.168.165.51,trsvcid=4420,host_traddr=192.168.165.73,src_addr=192.168.165.73 live optimized
+- nvme6 tcp
traddr=192.168.166.51,trsvcid=4420,host_traddr=192.168.166.73,src_addr=192.168.166.73 live non-optimized
+- nvme8 tcp
traddr=192.168.165.50,trsvcid=4420,host_traddr=192.168.165.73,src_addr=192.168.165.73 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/nvme2n9	vs_proxmox_FC_NVMeFC	/vol/vol_180_data_nvme4/ns
NSID	UUID	Size
----	-----	-----
1	e3d3d544-de8b-4787-93af-bfec7769e909	32.21GB

JSON

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

Show example

```
{
  "Device":"/dev/nvme2n9",
  "Vserver":"vs_proxmox_FC_NVMeFC",
  "Subsystem":"sub_176",
  "Namespace_Path":"/vol/vol_180_data_nvme4/ns",
  "NSID":9,
  "UUID":"e3d3d544-de8b-4787-93af-bfec7769e909",
  "LBA_Size":4096,
  "Namespace_Size":32212254720,
  "UsedBytes":67899392,
  "Version":"9.17.1"
}
```

Step 6: Review the known issues

There are no known issues.

Configure Proxmox VE 8.x for NVMe-oF with ONTAP storage

The Proxmox VE 8.x host supports the NVMe over Fibre Channel (NVMe/FC) and NVMe over TCP (NVMe/TCP) protocols with Asymmetric Namespace Access (ANA). ANA provides multipathing functionality equivalent to asymmetric logical unit access (ALUA) in iSCSI and FCP environments.

Learn how to configure NVMe over Fabrics (NVMe-oF) hosts for Proxmox VE 8.x. For more support and feature information, see [ONTAP support and features](#).

NVMe-oF with Proxmox VE 8.x has the following known limitation:

- The SAN boot configuration for NVMe-FC is not supported.

Step 1: Install Proxmox VE and NVMe software and verify your configuration

To configure your host for NVMe-oF you need to install the host and NVMe software packages, enable multipathing, and verify your host NQN configuration.

Steps

1. Install Proxmox 8.x on the server. After the installation is complete, verify that you are running the specified Proxmox 8.x kernel:

```
uname -r
```

The following example shows a Proxmox kernel version:

```
6.8.12-10-pve
```

2. Install the `nvme-cli` package:

```
apt list|grep nvme-cli
```

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

```
nvme-cli/oldstable,now 2.4+really2.3-3 amd64
```

3. Install the `libnvme` package:

```
apt list|grep libnvme
```

The following example shows an `libnvme` package version:

```
libnvme1/oldstable,now 1.3-1+deb12u1 amd64
```

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

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

The following example shows a `hostnqn` value:

```
nqn.2014-08.org.nvmeexpress:uuid:1536c9a6-f954-11ea-b24d-0a94efb46eaf
```

5. On the ONTAP system, verify that the `hostnqn` string matches the `hostnqn` string for the corresponding subsystem on the ONTAP array:

```
::> vserver nvme subsystem host show -vserver proxmox_120_122
```

Show example

```
Vserver Subsystem Priority Host NQN
-----
proxmox_120_122
proxmox_120_122
regular nqn.2014-08.org.nvmeexpress:uuid:1536c9a6-f954-11ea-b24d-0a94efb46eaf
regular nqn.2014-08.org.nvmeexpress:uuid:991a7476-f9bf-11ea-8b73-0a94efb46c3b
proxmox_120_122_tcp
regular nqn.2014-08.org.nvmeexpress:uuid:1536c9a6-f954-11ea-b24d-0a94efb46eaf
regular nqn.2014-08.org.nvmeexpress:uuid:991a7476-f9bf-11ea-8b73-0a94efb46c3b
2 entries were displayed.
```



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

Step 2: Configure NVMe/FC and NVMe/TCP

Configure NVMe/FC with Broadcom/Emulex or Marvell/QLogic adapters, or configure NVMe/TCP using manual discovery and connect operations.

NVMe/FC - 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:

```
LPe35002-M2  
LPe35002-M2
```

b. Display the model descriptions:

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

You should see an output similar to the following example:

```
Emulex LPe35002-M2 2-Port 32Gb Fibre Channel Adapter  
Emulex LPe35002-M2 2-Port 32Gb 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 command returns the firmware versions:

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

b. Display the inbox driver version:

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

The following example shows a driver 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
```

4. Verify that you can view your initiator ports:

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

You should see an output similar to:

```
0x100000109b95467e  
0x100000109b95467f
```

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 x10005ced8c531948 WWNN x20005ced8c531948
DID x082400
ONLINE
NVME RPORT WWPN x200ed039eac79573 WWNN x200dd039eac79573 DID
x060902
TARGET DISCSRV ONLINE
NVME RPORT WWPN x2001d039eac79573 WWNN x2000d039eac79573 DID
x060904
TARGET DISCSRV ONLINE
NVME Statistics LS: Xmt 0000000034 Cmpl 0000000034 Abort
00000000 LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 00000000000142cfb Issue 00000000000142cfc OutIO
0000000000000001 abort 00000005 noxri 00000000 nondlp 00000000
qdepth 00000000 wgerr 00000000 err 00000000 FCP CMPL: xb
00000005 Err 00000005 NVME Initiator Enabled XRI Dist lpfc1
Total 6144 IO 5894 ELS 250 NVME LPORT lpfc1 WWPN
x10005ced8c531949 WWNN x20005ced8c531949 DID x082500
ONLINE
NVME RPORT WWPN x2010d039eac79573 WWNN x200dd039eac79573 DID
x062902
TARGET DISCSRV ONLINE
NVME RPORT WWPN x2007d039eac79573 WWNN x2000d039eac79573 DID
x062904
TARGET DISCSRV ONLINE
NVME Statistics LS: Xmt 0000000034 Cmpl 0000000034 Abort
00000000 LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000000000d39f1 Issue 000000000000d39f2 OutIO
0000000000000001 abort 00000005 noxri 00000000 nondlp 00000000
qdepth 00000000 wgerr 00000000 err 00000000 FCP CMPL: xb
00000005 Err 00000005
```

NVMe/FC - Marvell/QLogic

Configure NVMe/FC for a Marvell/QLogic adapter.

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

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

The following 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 output is 1.

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. Check that the initiator port can get 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 192.168.2.22 -a 192.168.2.30

Discovery Log Number of Records 12, Generation counter 13
=====Discovery Log Entry 0=====
trtype: tcp
adrfam: ipv4
subtype: current discovery subsystem
treq: not specified
portid: 10
trsvcid: 8009
subnqn: nqn.1992-
08.com.netapp:sn.ae9f2d55a7ec11ef8751d039ea9e891c:discovery
traddr: 192.168.2.30
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.ae9f2d55a7ec11ef8751d039ea9e891c:discovery
traddr: 192.168.1.30
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.ae9f2d55a7ec11ef8751d039ea9e891c:discovery
traddr: 192.168.2.25
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:  11
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.ae9f2d55a7ec11ef8751d039ea9e891c:discovery
traddr:  192.168.1.25
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
====Discovery Log Entry 4=====
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  10
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.ae9f2d55a7ec11ef8751d039ea9e891c:subsystem.prox
mox_120_122
traddr:  192.168.2.30
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.ae9f2d55a7ec11ef8751d039ea9e891c:subsystem.prox
mox_120_122
traddr:  192.168.1.30
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.ae9f2d55a7ec11ef8751d039ea9e891c:subsystem.prox

```

```

mox_120_122
traddr: 192.168.2.25
eflags: none
sectype: none
=====Discovery Log Entry 7=====
trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 11
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.ae9f2d55a7ec11ef8751d039ea9e891c:subsystem.prox
mox_120_122
traddr: 192.168.1.25
eflags: none
sectype: none
=====Discovery Log Entry 8=====
trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 10
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.ae9f2d55a7ec11ef8751d039ea9e891c:subsystem.prox
mox_120_122_tcp
traddr: 192.168.2.30
eflags: none
sectype: none
=====Discovery Log Entry 9=====
trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 9
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.ae9f2d55a7ec11ef8751d039ea9e891c:subsystem.prox
mox_120_122_tcp
traddr: 192.168.1.30
eflags: none
sectype: none
=====Discovery Log Entry 10=====
trtype: tcp
adrfam: ipv4

```

```

subtype: nvme subsystem
treq:    not specified
portid:  12
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.ae9f2d55a7ec11ef8751d039ea9e891c:subsystem.prox
mox_120_122_tcp
traddr:  192.168.2.25
eflags:  none
sectype: none
=====Discovery Log Entry 11=====
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  11
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.ae9f2d55a7ec11ef8751d039ea9e891c:subsystem.prox
mox_120_122_tcp
traddr:  192.168.1.25
eflags:  none
sectype: none

```

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

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

Show example

```

nvme discover -t tcp -w 192.168.1.22 -a 192.168.1.30
nvme discover -t tcp -w 192.168.2.22 -a 192.168.2.30
nvme discover -t tcp -w 192.168.1.22 -a 192.168.1.25
nvme discover -t tcp -w 192.168.2.22 -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 192.168.1.22 -a 192.168.1.30
nvme connect-all -t tcp -w 192.168.2.22 -a 192.168.2.30
nvme connect-all -t tcp -w 192.168.1.22 -a 192.168.1.25
nvme connect-all -t tcp -w 192.168.2.22 -a 192.168.2.25
```

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

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 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 `update-initramfs` 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 4: Verify NVMe boot services

With Proxmox 8.x, the `nvmeof-boot-connections.service` and `nvmmf-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 `nvmeof-boot-connections.service` and `nvmmf-autoconnect.service` boot services are enabled.

Steps

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

```
systemctl status nvmmf-autoconnect.service
```

Show example output

```
o nvmmf-autoconnect.service - Connect NVMe-oF subsystems
automatically during boot
   Loaded: loaded (/lib/systemd/system/nvmmf-autoconnect.service;
enabled; preset: enabled)
   Active: inactive (dead) since Fri 2025-11-21 19:59:10 IST; 8s
ago
     Process: 256613 ExecStartPre=/sbin/modprobe nvme-fabrics
(code=exited, status=0/SUCCESS)
     Process: 256614 ExecStart=/usr/sbin/nvme connect-all
(code=exited, status=0/SUCCESS)
    Main PID: 256614 (code=exited, status=0/SUCCESS)
      CPU: 18ms
Nov 21 19:59:07 SR665-14-122.lab.eng.btc.netapp.in systemd[1]:
Starting nvmmf-autoconnect.service - Connect NVMe-oF subsystems
automatically during boot...
Nov 21 19:59:10 SR665-14-122.lab.eng.btc.netapp.in nvme[256614]:
Failed to write to /dev/nvme-fabrics: Invalid argument
Nov 21 19:59:10 SR665-14-122.lab.eng.btc.netapp.in nvme[256614]:
Failed to write to /dev/nvme-fabrics: Invalid argument
Nov 21 19:59:10 SR665-14-122.lab.eng.btc.netapp.in systemd[1]: nvmmf-
autoconnect.service: Deactivated successfully.
Nov 21 19:59:10 SR665-14-122.lab.eng.btc.netapp.in systemd[1]:
Finished nvmmf-autoconnect.service - Connect NVMe-oF subsystems
automatically during boot.
```

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

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

Show example output

```
o nvme-fc-boot-connections.service - Auto-connect to subsystems on
FC-NVME devices found during boot
   Loaded: loaded (/lib/systemd/system/nvme-fc-boot-
connections.service; enabled; preset: enabled)
   Active: inactive (dead) since Thu 2025-11-20 17:48:29 IST; 1
day 2h ago
   Process: 1381 ExecStart=/bin/sh -c echo add >
/sys/class/fc/fc_udev_device/nvme_discovery (code=exited,
status=0/SUCCESS)
   Main PID: 1381 (code=exited, status=0/SUCCESS)
   CPU: 3ms

Nov 20 17:48:29 SR665-14-122.lab.eng.btc.netapp.in systemd[1]:
Starting nvme-fc-boot-connections.service - Auto-connect to
subsystems on FC-NVME devices found during boot..
Nov 20 17:48:29 SR665-14-122.lab.eng.btc.netapp.in systemd[1]:
nvme-fc-boot-connections.service: Deactivated successfully.
Nov 20 17:48:29 SR665-14-122.lab.eng.btc.netapp.in systemd[1]:
Finished nvme-fc-boot-connections.service - Auto-connect to
subsystems on FC-NVME devices found during boot...
```

Step 5: Verify the multipathing configuration

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 ipolicy set to round-robin) for the ONTAP namespaces show up correctly on the host:
 - a. Show 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	Usage
Model		Namespace	
Format	FW Rev		

/dev/nvme2n20	/dev/ng2n20	81K13BUDdygsAAAAAAG	
NetApp ONTAP Controller		10	5.56 GB /
91.27 GB	4 KiB + 0 B	9.18.1	

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

NVMe/FC

```
nvme list-subsys /dev/nvme2n20
```

Show example

```
nvme-subsys2 - NQN= nqn.1992-08.com.netapp:sn.ae9f2d55a7ec11ef8751d039ea9e891c:subsystem.proxmox_120_122_tcp
\
+- nvme1 fc traddr=nn-0x200dd039eac79573:pn-0x2010d039eac79573,host_traddr=nn-0x20005ced8c531949:pn-0x10005ced8c531949 live optimized
+- nvme3 fc traddr=nn-0x200dd039eac79573:pn-0x200ed039eac79573,host_traddr=nn-0x20005ced8c531948:pn-0x10005ced8c531948 live optimized
+- nvme5 fc traddr=nn-0x200dd039eac79573:pn-0x200fd039eac79573,host_traddr=nn-0x20005ced8c531949:pn-0x10005ced8c531949 live non-optimized
+- nvme7 fc traddr=nn-0x200dd039eac79573:pn-0x2011d039eac79573,host_traddr=nn-0x20005ced8c531948:pn-0x10005ced8c531948 live non-optimized
```

NVMe/TCP

```
nvme list-subsys /dev/nvme2n3
```

Show example

```
nvme-subsys2 - NQN= qn.1992-  
08.com.netapp:sn.ae9f2d55a7ec11ef8751d039ea9e891c:subsystem.prox  
mox_120_122_tcp  
\n  
+- nvme2 tcp  
traddr=192.168.1.30,trsvcid=4420,host_traddr=192.168.1.22,src_ad  
dr=192.168.1.22 live optimized  
+- nvme4 tcp  
traddr=192.168.2.30,trsvcid=4420,host_traddr=192.168.2.22,src_ad  
dr=192.168.2.22 live optimized  
+- nvme6 tcp  
traddr=192.168.1.25,trsvcid=4420,host_traddr=192.168.1.22,src_ad  
dr=192.168.1.22 live non-optimized  
+- nvme8 tcp  
traddr=192.168.2.25,trsvcid=4420,host_traddr=192.168.2.22,src_ad  
dr=192.168.2.22 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/nvme2n11	proxmox_120_122	/vol/vm120_tcp1/ns
NSID	UUID	Size
----	-----	-----
1	5aefea74-f0cf-4794-a7e9-e113c4659aca	37.58GB

JSON

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

Show example

```
{
  "Device":"/dev/nvme2n11",
  "Vserver":"proxmox_120_122",
  "Namespace_Path":"/vol/vm120_tcp1/ns",
  "NSID":1,
  "UUID":"5aefea74-f0cf-4794-a7e9-e113c4659aca",
  "Size":"37.58GB",
  "LBA_Data_Size":4096,
  "Namespace_Size":32212254720
}
```

Step 6: Review the known issues

There are no known issues.

Copyright information

Copyright © 2026 NetApp, Inc. All Rights Reserved. Printed in the U.S. No part of this document covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—without prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP “AS IS” AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

LIMITED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (b)(3) of the Rights in Technical Data -Noncommercial Items at DFARS 252.227-7013 (FEB 2014) and FAR 52.227-19 (DEC 2007).

Data contained herein pertains to a commercial product and/or commercial service (as defined in FAR 2.101) and is proprietary to NetApp, Inc. All NetApp technical data and computer software provided under this Agreement is commercial in nature and developed solely at private expense. The U.S. Government has a non-exclusive, non-transferrable, nonsublicensable, worldwide, limited irrevocable license to use the Data only in connection with and in support of the U.S. Government contract under which the Data was delivered. Except as provided herein, the Data may not be used, disclosed, reproduced, modified, performed, or displayed without the prior written approval of NetApp, Inc. United States Government license rights for the Department of Defense are limited to those rights identified in DFARS clause 252.227-7015(b) (FEB 2014).

Trademark information

NETAPP, the NETAPP logo, and the marks listed at <http://www.netapp.com/TM> are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners.