



Rocky Linux

ONTAP SAN Host Utilities

NetApp
January 30, 2026

目錄

Rocky Linux	1
了解 Rocky Linux 對ONTAP 的支援與功能	1
下一步	1
配置 Rocky Linux 10.x 以支援 NVMe-oF 和ONTAP存儲	2
步驟 1：選擇性啟用 SAN 開機	2
步驟 2：安裝 Rocky Linux 和 NVMe 軟體並驗證您的配置	2
步驟 3：設定 NVMe/FC 和 NVMe/TCP	4
步驟 4：（可選）為 NVMe/FC 啟用 1MB I/O	12
步驟 5：驗證 NVMe 啟動服務	13
步驟 6：驗證多路徑配置	14
步驟 7：設定安全帶內身份驗證	18
步驟 8：檢閱已知問題	25
配置 Rocky Linux 9.x 以支援 NVMe-oF 和ONTAP存儲	25
步驟 1：選擇性啟用 SAN 開機	25
步驟 2：安裝 Rocky Linux 和 NVMe 軟體並驗證您的配置	25
步驟 3：設定 NVMe/FC 和 NVMe/TCP	27
步驟 4：（可選）為 NVMe/FC 啟用 1MB I/O	39
步驟 5：驗證 NVMe 啟動服務	40
步驟 6：驗證多路徑配置	41
步驟 7：設定安全帶內身份驗證	45
步驟 8：檢閱已知問題	53
配置 Rocky Linux 8.x 以支援 NVMe-oF 和ONTAP存儲	53
步驟 1：安裝 Rocky Linux 和 NVMe 軟體並驗證您的配置	53
步驟 2：設定 NVMe/FC 和 NVMe/TCP	55
步驟 3：可選，啟用 NVMe/FC 的 1MB I/O。	66
步驟 4：驗證多路徑配置	67
步驟 5：查看已知問題	71

Rocky Linux

了解 Rocky Linux 對ONTAP 的支援與功能

使用 NVMe over Fabrics (NVMe-oF) 進行主機配置所支援的功能因ONTAP和 Rocky Linux 的版本而異。

特徵	Rocky Linux 主機版本	ONTAP 版本
RHEL 主機與ONTAP控制器之間透過 NVMe/TCP 支援安全的帶內驗證。	9.3 或更高版本	9.12.1 或更高版本
NVMe/TCP 使用原生命名空間提供命名空間 `nvme-cli` 包裹	8.2 或更高版本	9.10.1 或更高版本
NVMe/TCP 是一項完全支援的企業級功能	9.0 或更高版本	9.10.1 或更高版本
在同一主機上支援 NVMe 和 SCSI 流量，NVMe-oF 命名空間使用 NVMe 多路徑，SCSI LUN 使用 dm 多路徑。	8.2 或更高版本	9.4 或更高版本

無論系統設定中執行的ONTAP版本為何，ONTAP支援以下 SAN 主機功能。

特徵	Rocky Linux 主機版本
預設啟用原生 NVMe 多路徑。	10.0 或更高版本
SAN 啟動已透過 NVMe/FC 協定啟用	9.4 或更高版本
這 `nvme-cli` 軟體包包含自動連接腳本，無需第三方腳本。	8.2 或更高版本
<code>nvme-cli</code> 軟體套件中的原生 udev 規則為 NVMe 多路徑提供輪詢負載平衡	8.2 或更高版本



有關受支援配置的詳細信息，請參閱[互通性對照表工具](#)。

下一步

如果您的 Rocky Linux 版本是...	了解...
10系列	"為 Rocky Linux 10.x 配置 NVMe"
9系列	"為 Rocky Linux 9.x 配置 NVMe"
8系列	"為 Rocky Linux 8.x 配置 NVMe"

相關資訊

["了解如何管理 NVMe 協議"](#)

配置 Rocky Linux 10.x 以支援 NVMe-oF 和ONTAP存儲

Rocky Linux 主機支援基於光纖通道的 NVMe (NVMe/FC) 和基於 TCP 的 NVMe (NVMe/TCP) 協議，並支援非對稱命名空間存取 (ANA)。ANA 提供與 iSCSI 和 FCP 環境中的非對稱邏輯單元存取 (ALUA) 等效的多路徑功能。

了解如何為 Rocky Linux 10.x 設定 NVMe over Fabrics (NVMe-oF) 主機。如需更多支援和功能信息，請參閱["Rocky Linux ONTAP支援與功能"](#)。

NVMe-oF 與 Rocky Linux 10.x 有以下已知限制：

- 這 `nvme disconnect-all` 此命令會斷開根檔案系統和資料檔案系統，可能會導致系統不穩定。請勿在透過 NVMe-TCP 或 NVMe-FC 命名空間從 SAN 啟動的系統上執行此操作。

步驟 1：選擇性啟用 SAN 開機

您可以設定主機以使用 SAN 啟動來簡化部署並提高可擴充性。使用["互通性對照表工具"](#)驗證您的 Linux 作業系統、主機匯流排適配器 (HBA)、HBA 韌體、HBA 啟動 BIOS 和ONTAP版本是否支援 SAN 啟動。

步驟

1. ["建立 NVMe 命名空間並將其對應到主機"](#)。
2. 在伺服器 BIOS 中為 SAN 啟動命名空間對應到的連接埠啟用 SAN 啟動。

如需如何啟用HBA BIOS的相關資訊、請參閱廠商專屬的文件。

3. 重新啟動主機並驗證作業系統是否已啟動並正在運行。

步驟 2：安裝 Rocky Linux 和 NVMe 軟體並驗證您的配置

若要為 NVMe-oF 設定主機，您需要安裝主機和 NVMe 軟體包，啟用多路徑，並驗證主機 NQN 設定。

步驟

1. 在伺服器上安裝 Rocky Linux 10.x。安裝完成後，請確認您執行的是所需的 Rocky Linux 10.x 核心：

```
uname -r
```

Rocky Linux 核心版本範例：

```
6.12.0-55.9.1.el10_0.x86_64
```

2. 安裝「NVMe-CLI (NVMe - CLI) 套件：

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

下面的例子展示了 `nvme-cli` 軟體包版本：

```
nvme-cli-2.11-5.el10.x86_64
```

3. 安裝 libnvme 套件：

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

下面的例子展示了 `libnvme` 軟體包版本：

```
libnvme-1.11.1-1.el10.x86_64
```

4. 在主機上，檢查 hostnqn 字串 /etc/nvme/hostnqn：

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

下面的例子展示了 `hostnqn` 價值：

```
nqn.2014-08.org.nvmeexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
```

5. 在 ONTAP 系統中，驗證以下資訊：`hostnqn` 字串匹配 `hostnqn` ONTAP 陣列中對應子系統的字串：

```
::> vserver nvme subsystem host show -vserver vs_nvme_194_rockylinux10
```

顯示範例

```
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.
```



如果 `hostnqn` 字串不匹配，請使用 `vserver modify` 命令來更新 `hostnqn` 對應 ONTAP 儲存系統子系統上的字串以匹配 `hostnqn` 字串來自 `/etc/nvme/hostnqn` 在主機上。

步驟 3：設定 NVMe/FC 和 NVMe/TCP

使用 Broadcom/Emulex 或 Marvell/QLogic 適配器配置 NVMe/FC，或使用手動發現和連接操作來設定 NVMe/TCP。

NVMe/FC - 博通/Emulex

為 Broadcom / Emulex 介面卡設定 NVMe / FC 。

步驟

1. 確認您使用的是支援的介面卡機型：

a. 顯示模型名稱：

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

您應該會看到下列輸出：

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

b. 顯示模型描述：

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

您應該會看到類似以下範例的輸出：

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

2. 驗證您使用的是建議的Broadcom lpfc 韌體與收件匣驅動程式：

a. 顯示韌體版本：

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

該命令返回韌體版本：

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

b. 顯示收件匣驅動程式版本：

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

以下範例顯示了驅動程式版本：

```
0:14.4.0.6
```

如需支援的介面卡驅動程式和韌體版本的最新清單，請參閱["互通性對照表工具"](#)。

3. 請確認 `lpfc_enable_fc4_type` 設為 3：

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

4. 確認您可以檢視啟動器連接埠：

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

您應該會看到類似如下的輸出：

```
0x2100f4c7aa0cd7c2  
0x2100f4c7aa0cd7c3
```

5. 驗證啟動器連接埠是否在線上：

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

您應該會看到下列輸出：

```
Online  
Online
```

6. 確認已啟用 NVMe / FC 啟動器連接埠、且目標連接埠可見：

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


顯示範例

```
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
ffffffffffffe498bd
        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
ffffffffffffe5d74f
        abort 000014a5 noxri 00000000 nondlp 0000006a qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 000014a5 Err 0002149a
```

NVMe/FC - Marvell/QLogic

為 Marvell/QLogic 介面卡設定 NVMe / FC 。

步驟

1. 驗證您使用的適配器驅動程式和韌體版本是否受支援：

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

以下範例顯示了驅動程式和韌體版本：

```
QLE2872 FW:v9.15.00 DVR:v10.02.09.300-k  
QLE2872 FW:v9.15.00 DVR:v10.02.09.300-k
```

2. 請確認 `ql2xnvmeenable` 已設定。這可讓 Marvell 介面卡作為 NVMe / FC 啟動器運作：

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

預期輸出為 1。

NVMe / TCP

NVMe/TCP 協定不支援自動連線操作。相反，您可以透過執行 NVMe/TCP 來發現 NVMe/TCP 子系統和命名空間 `connect` 或者 `connect-all` 手動操作。

步驟

1. 檢查啟動器連接埠是否可以跨支援的 NVMe/TCP LIF 取得發現日誌頁面資料：

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

顯示範例

```
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.rock
ylinux10_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.rock
ylinux10_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.rock
ylinux10_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.rock
ylinux10_tcp_subsystem
traddr: 192.168.20.20
eflags: none
sectype: none

```

2. 驗證其他 NVMe/TCP 啟動器-目標 LIF 組合是否可以成功檢索發現日誌頁面資料：

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

顯示範例

```

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. 執行 nvme connect-all 跨所有節點支援的 NVMe / TCP 啟動器目標生命體執行命令：

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

顯示範例

```
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
```

從 Rocky Linux 9.4 開始，NVMe/TCP 的設置 `ctrl\_loss\_tmo timeout` 自動設定為“關閉”。因此：

- 重試次數沒有限制（無限重試）。
- 您不需要手動配置特定的 `ctrl\_loss\_tmo timeout` 使用時長 `nvme connect` 或者 `nvme connect-all` 命令（選項 -l）。
- 如果發生路徑故障，NVMe/TCP 控制器不會逾時，並且會無限期地保持連線。

步驟 4：（可選）為 NVMe/FC 啟用 1MB I/O

ONTAP 在識別控制器資料中報告最大資料傳輸大小 (MDTS) 為 8。這意味著最大 I/O 請求大小可達 1MB。若要向 Broadcom NVMe/FC 主機發出 1MB 大小的 I/O 要求，您應該會增加 `lpfc` 的價值 `lpfc\_sg\_seg\_cnt` 參數從預設值 64 更改為 256。



這些步驟不適用於 Qlogic NVMe / FC 主機。

步驟

1. 將 `lpfc\_sg\_seg\_cnt` 參數設定為 256：

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

您應該會看到類似以下範例的輸出：

```
options lpfc lpfc_sg_seg_cnt=256
```

2. 執行 `dracut -f` 命令，然後重新啟動主機。
3. 確認的值 `lpfc\_sg\_seg\_cnt` 為 256：

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

步驟 5：驗證 NVMe 啟動服務

這 `nvmeof-boot-connections.service` 和 `nvmf-autoconnect.service` NVMe/FC 中包含的啟動服務 `nvme-cli` 系統啟動時，軟體包會自動啟用。

啟動完成後，驗證 `nvmeof-boot-connections.service` 和 `nvmf-autoconnect.service` 啟動服務已啟用。

步驟

1. 確認 `nvmf-autoconnect.service` 已啟用：

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

顯示範例輸出

```
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. 確認 `nvmeof-boot-connections.service` 已啟用：

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

顯示範例輸出

```
nvme-fc-boot-connections.service - Auto-connect to subsystems on FC-
NVME devices found during boot
   Loaded: loaded (/usr/lib/systemd/system/nvme-fc-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]: nvme-fc-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.
```

步驟 6：驗證多路徑配置

驗證核心內建 NVMe 多重路徑狀態，ANA 狀態和 ONTAP 命名空間是否適用於 NVMe 組態。

步驟

1. 確認已啟用核心內建 NVMe 多重路徑：

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

您應該會看到下列輸出：

```
Y
```

2. 驗證個別 ONTAP 命名空間的適當 NVMe 設定（例如、模型設定為 NetApp ONTAP 控制器、負載平衡 iopolicing 設定為循環）是否正確反映在主機上：

- a. 顯示子系統：

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

您應該會看到下列輸出：


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

b. 顯示策略：

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

您應該會看到下列輸出：

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

3. 確認已在主機上建立並正確探索命名空間：

```
nvme list
```

顯示範例

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. 確認每個路徑的控制器狀態均為有效、且具有正確的ANA狀態：

NVMe / FC

```
nvme list-subsys /dev/nvme5n1
```

顯示範例

```
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
```

顯示範例

```
nvme-subsys4 - NQN=nqn.1992-08.com.netapp:sn.64e65e6caae711ef9668d039ea951c46:subsystem.nvme4
                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_addr=192.168.21.1 live non-optimized
+- nvme151 tcp
traddr=192.168.21.21,trsvcid=4420,host_traddr=192.168.21.1,src_addr=192.168.21.1 live optimized
+- nvme4 tcp
traddr=192.168.20.20,trsvcid=4420,host_traddr=192.168.20.1,src_addr=192.168.20.1 live non-optimized
+- nvme53 tcp
traddr=192.168.20.21,trsvcid=4420,host_traddr=192.168.20.1,src_addr=192.168.20.1 live optimized
```

5. 驗證NetApp外掛程式是否顯示每ONTAP 個版本名稱空間裝置的正確值：

欄位

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

顯示範例

Device	Vserver	Namespace	Path
/dev/nvme10n1	vs_tcp_rockylinux10		/vol/vol110/ns10

NSID	UUID	Size
1	bbf51146-fc64-4197-b8cf-8a24f6f359b3	21.47GB

JSON

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

顯示範例

```
{
  "ONTAPdevices": [
    {
      "Device": "/dev/nvme10n1",
      "Vserver": "vs_tcp_rockylinux10",
      "Namespace_Path": "/vol/vol110/ns10",
      "NSID": 1,
      "UUID": "bbf51146-fc64-4197-b8cf-8a24f6f359b3",
      "Size": "21.47GB",
      "LBA_Data_Size": 4096,
      "Namespace_Size": 5242880
    }
  ]
}
```

步驟 7：設定安全帶內身份驗證

Rocky Linux 10.x 主機和ONTAP控制器之間透過 NVMe/TCP 支援安全的帶內身份驗證。

每個主機或控制器必須與一個 `DH-HMAC-CHAP` 金鑰來設定安全身份驗證。`DH-HMAC-CHAP` 金鑰是 NVMe

主機或控制器的 NQN 與管理員配置的身份驗證金鑰的組合。若要驗證其對等端點、NVMe 主機或控制器必須識別與對等端點相關的金鑰。

步驟

使用 CLI 或設定 JSON 檔案設定安全帶內身份驗證。如果您需要為不同的子系統指定不同的 dhchap 金鑰、則必須使用組態 JSON 檔案。

CLI

使用 CLI 設定安全的頻內驗證。

1. 取得主機 NQN：

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

2. 為 Rocky Linux 10.x 主機產生 dhchap 金鑰。

以下輸出描述了 `gen-dhchap-key` 命令參數：

```
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

在下列範例中、會產生一個隨機的 dhchap 金鑰、其中 HMAC 設為 3（SHA-512）。

```
nvme gen-dhchap-key -m 3 -n nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0035-5910-804b-c2c04f444d33
DHHC-
1:03:7zf8I9gaRcDWH3tCH5vLGaoyjzPIvwNWusBfKdpJa+hialaKDKJQ2o53pX3wYM9
xdv5DtKNNhJInZ7X8wU2RQpQIngc=:
```

3. 在 ONTAP 控制器上、新增主機並指定兩個 dhchap 金鑰：

```
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. 主機支援兩種驗證方法：單向和雙向。在主機上、連線至 ONTAP 控制器、並根據所選的驗證方法指定 dhchap 金鑰：

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

5. 驗證 nvme connect authentication 命令驗證主機和控制器 dhchap 金鑰：

a. 驗證主機 dhchap 金鑰：

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

顯示單向組態的輸出範例

```
cat /sys/class/nvme-subsystem/nvme-subsys1/nvme*/dhchap_secret  
DHHC-  
1:03:fMcrJharXUOqRoIsOEaG6m2PH1yYvu5+z3jTmzEKUbcWu26I33b93b  
i12WR09XDho/1d3L45J+0FeCsStBEAfhYgkQU=:  
DHHC-  
1:03:fMcrJharXUOqRoIsOEaG6m2PH1yYvu5+z3jTmzEKUbcWu26I33b93b  
i12WR09XDho/1d3L45J+0FeCsStBEAfhYgkQU=:  
DHHC-  
1:03:fMcrJharXUOqRoIsOEaG6m2PH1yYvu5+z3jTmzEKUbcWu26I33b93b  
i12WR09XDho/1d3L45J+0FeCsStBEAfhYgkQU=:  
DHHC-  
1:03:fMcrJharXUOqRoIsOEaG6m2PH1yYvu5+z3jTmzEKUbcWu26I33b93b  
i12WR09XDho/1d3L45J+0FeCsStBEAfhYgkQU=:
```

b. 驗證控制器 dhchap 按鍵：

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

顯示雙向組態的輸出範例

```
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

當ONTAP控制器上有多個 NVMe 子系統可用時，您可以使用 `/etc/nvme/config.json` 文件與 `nvme connect-all` 命令。

使用 `-o` 選項來產生 JSON 檔案。有關更多語法選項，請參閱 NVMe connect-all 手冊頁。

1. 配置 JSON 檔案。



在以下範例中，`dhchap_key` 對應於 `dhchap_secret` 和 `dhchap_ctrl_key` 對應於 `dhchap_ctrl_secret`。

顯示範例

```
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+hialaKDKJQ2o53pX3wYM9xdv5DtKNNhJInZ7X8wU2RQpQIngc=",
    "subsystems": [
      {
        "nqn": "nqn.1992-08.com.netapp:sn.127ade26168811f0a50ed039eab69ad3:subsystem.inband_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. 使用組態 JSON 檔案連線至 ONTAP 控制器：

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

顯示範例

```
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. 驗證每個子系統的相應控制器是否已啟用 dhchap 機密。

a. 驗證主機 dhchap 金鑰：

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

以下範例顯示了 dhchap 金鑰：

```
DHHC-1:03:7zf8I9gaRcDWH3tCH5vLGAoyjzPIvwNWusBfKdpJa+hial
aKDKJQ2o53pX3wYM9xdv5DtKNNhJInZ7X8wU2RQpQIngc=:
```

b. 驗證控制器 dhchap 按鍵：

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

您應該會看到類似以下範例的輸出：

```
DHHC-1:03:fMcrJharXUOqRoIsOEaG6m2PH1yYvu5+z3jT  
mzEKUbcWu26I33b93bil2WR09XDho/ld3L45J+0FeCsStBEAfhYgkQU=:
```

步驟 8：檢閱已知問題

沒有已知問題。

配置 Rocky Linux 9.x 以支援 NVMe-oF 和ONTAP存儲

Rocky Linux 主機支援基於光纖通道的 NVMe (NVMe/FC) 和基於 TCP 的 NVMe (NVMe/TCP) 協議，並支援非對稱命名空間存取 (ANA)。ANA 提供與 iSCSI 和 FCP 環境中的非對稱邏輯單元存取 (ALUA) 等效的多路徑功能。

了解如何為 Rocky Linux 9.x 設定 NVMe over Fabrics (NVMe-oF) 主機。如需更多支援和功能信息，請參閱 ["Rocky Linux ONTAP支援與功能"](#)。

NVMe-oF 與 Rocky Linux 9.x 有以下已知限制：

- 這 `nvme disconnect-all` 此命令會斷開根檔案系統和資料檔案系統，可能會導致系統不穩定。請勿在透過 NVMe-TCP 或 NVMe-FC 命名空間從 SAN 啟動的系統上執行此操作。

步驟 1：選擇性啟用 SAN 開機

您可以設定主機以使用 SAN 啟動來簡化部署並提高可擴充性。使用["互通性對照表工具"](#)驗證您的 Linux 作業系統、主機匯流排適配器 (HBA)、HBA 韌體、HBA 啟動 BIOS 和ONTAP版本是否支援 SAN 啟動。

步驟

1. ["建立 NVMe 命名空間並將其對應到主機"](#)。
2. 在伺服器 BIOS 中為 SAN 啟動命名空間對應到的連接埠啟用 SAN 啟動。

如需如何啟用HBA BIOS的相關資訊、請參閱廠商專屬的文件。

3. 重新啟動主機並驗證作業系統是否已啟動並正在運行。

步驟 2：安裝 Rocky Linux 和 NVMe 軟體並驗證您的配置

若要為 NVMe-oF 設定主機，您需要安裝主機和 NVMe 軟體包，啟用多路徑，並驗證主機 NQN 設定。

步驟

1. 在伺服器上安裝 Rocky Linux 9.x。安裝完成後，請確認您執行的是所需的 Rocky Linux 9.x 核心：

```
uname -r
```

Rocky Linux 核心版本範例：

```
5.14.0-570.12.1.el9_6.x86_64
```

2. 安裝「NVMe-CLI (NVMe - CLI) 套件：

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

以下範例顯示了 nvme-cli 軟體包版本：

```
nvme-cli-2.11-5.el9.x86_64
```

3. 安裝 libnvme 套件：

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

下面的例子展示了 `libnvme` 軟體包版本：

```
libnvme-1.11.1-1.el9.x86_64
```

4. 在 Rocky Linux 主機上，檢查 hostnqn 字串 /etc/nvme/hostnqn：

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

下面的例子展示了 `hostnqn` 版本：

```
nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
```

5. 在 ONTAP 系統中，驗證以下資訊：`hostnqn` 字串匹配 `hostnqn` ONTAP 陣列中對應子系統的字串：

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

```

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.

```



如果是 hostnqn 字串不相符、請使用 `vserver modify` 命令來更新 hostnqn 對應 ONTAP 陣列子系統上的字串、以符合 hostnqn 字串來源 `/etc/nvme/hostnqn` 在主機上。

步驟 3：設定 NVMe/FC 和 NVMe/TCP

使用 Broadcom/Emulex 或 Marvell/QLogic 適配器配置 NVMe/FC，或使用手動發現和連接操作來設定 NVMe/TCP。

NVMe/FC - 博通/Emulex

為 Broadcom / Emulex 介面卡設定 NVMe / FC 。

步驟

1. 確認您使用的是支援的介面卡機型：

a. 顯示模型名稱：

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

您應該會看到下列輸出：

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

b. 顯示模型描述：

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

您應該會看到類似以下範例的輸出：

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

2. 驗證您使用的是建議的Broadcom lpfc 韌體與收件匣驅動程式：

a. 顯示韌體版本：

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

該命令返回韌體版本：

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

b. 顯示收件匣驅動程式版本：

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

以下範例顯示了驅動程式版本：

```
0:14.4.0.6
```

如需支援的介面卡驅動程式和韌體版本的最新清單，請參閱["互通性對照表工具"](#)。

3. 請確認 `lpfc_enable_fc4_type` 設為 3：

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

4. 確認您可以檢視啟動器連接埠：

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

以下範例顯示連接埠標識：

```
0x2100f4c7aa0cd7c2  
0x2100f4c7aa0cd7c3
```

5. 驗證啟動器連接埠是否在線上：

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

您應該會看到下列輸出：

```
Online  
Online
```

6. 確認已啟用 NVMe / FC 啟動器連接埠、且目標連接埠可見：

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

```

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 DISCSRV ONLINE
NVME RPORT          WWPN x2155d039eaa7dfc8 WWNN x2154d039eaa7dfc8
DID x02130f TARGET DISCSRV ONLINE
NVME RPORT          WWPN x2001d039eaa7dfc8 WWNN x2000d039eaa7dfc8
DID x021310 TARGET DISCSRV ONLINE
NVME RPORT          WWPN x200dd039eaa7dfc8 WWNN x2008d039eaa7dfc8
DID x020b15 TARGET DISCSRV ONLINE
NVME RPORT          WWPN x2156d039eaa7dfc8 WWNN x2154d039eaa7dfc8
DID x020b0d TARGET DISCSRV ONLINE
NVME RPORT          WWPN x2003d039eaa7dfc8 WWNN x2000d039eaa7dfc8
DID x020b10 TARGET DISCSRV 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
ffffffffffffc994c
          abort 000036d3 noxri 00000313 nondlp 00000c8d qdepth
000000000 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
ffffffffffffc04be
          abort 000035cc noxri 0000038c nondlp 000009e3 qdepth
000000000 wqerr 00000082 err 00000000
FCP Cmpl: xb 000035cc Err 000fcfc0

```

NVMe/FC - Marvell/QLogic

為 Marvell/QLogic 介面卡設定 NVMe / FC 。

步驟

1. 驗證您使用的適配器驅動程式和韌體版本是否受支援：

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

以下範例顯示了驅動程式和韌體版本：

```
QLE2872 FW:v9.15.00 DVR:v10.02.09.300-k  
QLE2872 FW:v9.15.00 DVR:v10.02.09.300-k
```

2. 請確認 `ql2xnvmeenable` 已設定。這可讓 Marvell 介面卡作為 NVMe / FC 啟動器運作：

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

預期輸出為 1。

NVMe / TCP

NVMe/TCP協定不支援自動連線操作。您必須手動執行 NVMe/TCP 連接或連接所有操作才能發現 NVMe/TCP 子系統和命名空間。

步驟

1. 檢查啟動器連接埠是否可以跨支援的 NVMe/TCP LIF 取得發現日誌頁面資料：

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

顯示範例

```
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
treql: not specified
portid: 4
trsvcid: 8009
subnqn: nqn.1992-
08.com.netapp:sn.0f4bale74eb611ef9f50d039eab6cb6d: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
treql: not specified
portid: 2
trsvcid: 8009
subnqn: nqn.1992-
08.com.netapp:sn.0f4bale74eb611ef9f50d039eab6cb6d: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
treql: not specified
portid: 5
trsvcid: 8009
subnqn: nqn.1992-
08.com.netapp:sn.0f4bale74eb611ef9f50d039eab6cb6d: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.0f4bale74eb611ef9f50d039eab6cb6d: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.0f4bale74eb611ef9f50d039eab6cb6d: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.0f4bale74eb611ef9f50d039eab6cb6d: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.0f4bale74eb611ef9f50d039eab6cb6d: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.0f4bale74eb611ef9f50d039eab6cb6d: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.0f4bale74eb611ef9f50d039eab6cb6d: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.0f4bale74eb611ef9f50d039eab6cb6d: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.0f4bale74eb611ef9f50d039eab6cb6d: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.0f4bale74eb611ef9f50d039eab6cb6d: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.0f4bale74eb611ef9f50d039eab6cb6d: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.0f4bale74eb611ef9f50d039eab6cb6d: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.0f4bale74eb611ef9f50d039eab6cb6d: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.0f4bale74eb611ef9f50d039eab6cb6d:subsystem.
Bidirectional_DHCP_NONE_2_9
traddr:  192.168.1.24
eflags:  none
sectype: none

```

2. 驗證其他 NVMe/TCP 啟動器-目標 LIF 組合是否可以成功取得發現日誌頁面資料：

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


顯示範例

```
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. 執行 `nvme connect-all` 跨所有節點支援的 NVMe / TCP 啟動器目標生命體執行命令：

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

顯示範例

```
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
```

從 Rocky Linux 9.4 開始，NVMe/TCP 的設置 `ctrl_loss_tmo timeout` 自動設定為“關閉”。因此：

- 重試次數沒有限制（無限重試）。
- 您不需要手動配置特定的 `ctrl_loss_tmo timeout` 使用時長 `nvme connect` 或者 `nvme connect-all` 命令（選項 -l）。
- 如果發生路徑故障，NVMe/TCP 控制器不會逾時，並且會無限期地保持連線。

步驟 4：（可選）為 NVMe/FC 啟用 1MB I/O

ONTAP 在識別控制器資料中報告最大資料傳輸大小 (MDTS) 為 8。這意味著最大 I/O 請求大小可達 1MB。若要向 Broadcom NVMe/FC 主機發出 1MB 大小的 I/O 要求，您應該會增加 `lpfc` 的價值 `lpfc_sg_seg_cnt` 參數從預設值 64 更改為 256。



這些步驟不適用於 Qlogic NVMe / FC 主機。

步驟

1. 將 `lpfc_sg_seg_cnt` 參數設定為 256：

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

您應該會看到類似以下範例的輸出：

```
options lpfc lpfc_sg_seg_cnt=256
```

2. 執行 `dracut -f` 命令，然後重新啟動主機。
3. 確認的值 `lpfc\_sg\_seg\_cnt` 為 256：

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

步驟 5：驗證 NVMe 啟動服務

這 `nvme-fc-boot-connections.service` 和 `nvme-fc-autoconnect.service` NVMe/FC 中包含的啟動服務 `nvme-cli` 系統啟動時，軟體包會自動啟用。

啟動完成後，驗證 `nvme-fc-boot-connections.service` 和 `nvme-fc-autoconnect.service` 啟動服務已啟用。

步驟

1. 確認 `nvme-fc-autoconnect.service` 已啟用：

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

顯示範例輸出

```
nvme-fc-autoconnect.service - Connect NVMe-oF subsystems automatically
during boot
   Loaded: loaded (/usr/lib/systemd/system/nvme-fc-
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]: nvme-fc-
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. 確認 `nvme-fc-boot-connections.service` 已啟用：

```
systemctl status nvme-fc-boot-connections.service
```

顯示範例輸出

```
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 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]: nvmeofc-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.
```

步驟 6：驗證多路徑配置

驗證核心內建 NVMe 多重路徑狀態，ANA 狀態和 ONTAP 命名空間是否適用於 NVMe 組態。

步驟

1. 確認已啟用核心內建 NVMe 多重路徑：

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

您應該會看到下列輸出：

```
Y
```

2. 驗證個別 ONTAP 命名空間的適當 NVMe 設定（例如、模型設定為 NetApp ONTAP 控制器、負載平衡 iopolicing 設定為循環）是否正確反映在主機上：

- a. 顯示子系統：

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

您應該會看到下列輸出：

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

b. 顯示策略：

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

您應該會看到下列輸出：

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

3. 確認已在主機上建立並正確探索命名空間：

```
nvme list
```

顯示範例

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. 確認每個路徑的控制器狀態均為有效、且具有正確的ANA狀態：

NVMe / FC

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

顯示範例

```
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
```

顯示範例

```
nvme-subsys5 - NQN=nqn.1992-08.com.netapp:sn.0f4bale74eb611ef9f50d039eab6cb6d: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. 驗證NetApp外掛程式是否顯示每ONTAP 個版本名稱空間裝置的正確值：

欄位

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

顯示範例

Device	Vserver	Namespace	Path
/dev/nvme1n1	linux_tcnvme_iscsi		
/vol/tcpcnvme_1_0_0/tcpcnvme_ns			
NSID	UUID		Size
1	5f7f630d-8ea5-407f-a490-484b95b15dd6		21.47GB

JSON

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

顯示範例

```
{
  "ONTAPdevices": [
    {
      "Device": "/dev/nvme1n1",
      "Vserver": "linux_tcnvme_iscsi",
      "Namespace_Path": "/vol/tcpcnvme_1_0_0/tcpcnvme_ns",
      "NSID": 1,
      "UUID": "5f7f630d-8ea5-407f-a490-484b95b15dd6",
      "Size": "21.47GB",
      "LBA_Data_Size": 4096,
      "Namespace_Size": 5242880
    },
  ]
}
```

步驟 7：設定安全帶內身份驗證

Rocky Linux 9x 主機和ONTAP控制器之間透過 NVMe/TCP 支援安全的帶內身份驗證。

每個主機或控制器必須與一個 `DH-HMAC-CHAP` 金鑰來設定安全身份驗證。`DH-HMAC-CHAP` 金鑰是 NVMe 主機或控制器的 NQN 與管理員配置的身份驗證金鑰的組合。若要驗證其對等端點、NVMe 主機或控制器必須識別與對等端點相關的金鑰。

步驟

使用 CLI 或設定 JSON 檔案設定安全帶內身份驗證。如果您需要為不同的子系統指定不同的 dhchap 金鑰、則必須使用組態 JSON 檔案。

CLI

使用 CLI 設定安全的頻內驗證。

1. 取得主機 NQN：

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

2. 為 Rocky Linux 9.x 主機產生 dhchap 金鑰。

以下輸出描述了 `gen-dhchap-key` 命令參數：

```
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
```

在下列範例中、會產生一個隨機的 dhchap 金鑰、其中 HMAC 設為 3（SHA-512）。

```
nvme gen-dhchap-key -m 3 -n nqn.2014-
08.org.nvmexpress:uuid:e6dade64-216d-11ec-b7bb-7ed30a5482c3
DHHC-
1:03:wSpuuKbBHTzC0W9JZxMBSYd9JFV8Si9aDh22k2BR/4m852vH7KGlrJeMpzhmyjD
W0o0PJJM6yZsTeEpGkDHMHQ255+g=:
```

3. 在 ONTAP 控制器上、新增主機並指定兩個 dhchap 金鑰：

```
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. 主機支援兩種驗證方法：單向和雙向。在主機上、連線至 ONTAP 控制器、並根據所選的驗證方法指定 dhchap 金鑰：

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

5. 驗證 nvme connect authentication 命令驗證主機和控制器 dhchap 金鑰：

a. 驗證主機 dhchap 金鑰：

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

顯示單向組態的輸出範例

```
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. 驗證控制器 dhchap 按鍵：

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

顯示雙向組態的輸出範例

```
cat /sys/class/nvme-subsystem/nvme-
subsys6/nvme*/dhchap_ctrl_secret
DHHC-
1:03:wSpuuKbBHTzC0W9JZxMBsYd9JFV8Si9aDh22k2BR/4m852vH7KGlrJeMp
zhmyjDWOo0PJJM6yZsTEpGkDHMHQ255+g=:
DHHC-
1:03:wSpuuKbBHTzC0W9JZxMBsYd9JFV8Si9aDh22k2BR/4m852vH7KGlrJeMp
zhmyjDWOo0PJJM6yZsTEpGkDHMHQ255+g=:
DHHC-
1:03:wSpuuKbBHTzC0W9JZxMBsYd9JFV8Si9aDh22k2BR/4m852vH7KGlrJeMp
zhmyjDWOo0PJJM6yZsTEpGkDHMHQ255+g=:
DHHC-
1:03:wSpuuKbBHTzC0W9JZxMBsYd9JFV8Si9aDh22k2BR/4m852vH7KGlrJeMp
zhmyjDWOo0PJJM6yZsTEpGkDHMHQ255+g=:
```

JSON

當ONTAP控制器上有多個 NVMe 子系統可用時，您可以使用 `/etc/nvme/config.json` 文件與 `nvme connect-all` 命令。

使用 `-o` 選項來產生 JSON 檔案。有關更多語法選項，請參閱 NVMe connect-all 手冊頁。

1. 配置 JSON 檔案。



在以下範例中，`dhchap_key` 對應於 `dhchap_secret` 和 `dhchap_ctrl_key` 對應於 `dhchap_ctrl_secret`。

顯示範例

```
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.0f4bale74eb611ef9f50d039eab6cb6d: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/4m852vH7KG1rJeMpzhmyjDW
Oo0PJJM6yZsTeEpGkDHMHQ255+g="
        },
        {
          "transport":"tcp",
          "traddr":" 192.168.1.25 ",
          "host_traddr":" 192.168.1.31",
          "trsvcid":"4420",
          "dhchap_ctrl_key":"DHHC-
1:03:
wSpuuKbBHTzC0W9JZxMBsYd9JFV8Si9aDh22k2BR/4m852vH7KG1rJeMpzhmyjDW
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. 使用組態 JSON 檔案連線至 ONTAP 控制器：

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

顯示範例

```
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. 確認已為每個子系統的個別控制器啟用 dhchap 機密：

a. 驗證主機 dhchap 金鑰：

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

以下範例顯示了 dhchap 金鑰：

```
DHHC-1:01:CNxTYq73T9vJk0JpOfDBZrhDCqpWBN4XVZI5WxwPgDUieHAI:
```

b. 驗證控制器 dhchap 按鍵：

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

您應該會看到類似以下範例的輸出：

```
DHHC-  
1:03:wSpuuKbBHTzC0W9JZxMBSYd9JFV8Si9aDh22k2BR/4m852vH7KGlrJeMpzhmyjD  
W0o0PJJM6yZsTeEpGkDHMHQ255+g=:
```

步驟 8：檢閱已知問題

沒有已知問題。

配置 Rocky Linux 8.x 以支援 NVMe-oF 和ONTAP存儲

Rocky Linux 主機支援基於光纖通道的 NVMe (NVMe/FC) 和基於 TCP 的 NVMe (NVMe/TCP) 協議，並支援非對稱命名空間存取 (ANA)。ANA 提供與 iSCSI 和 FCP 環境中的非對稱邏輯單元存取 (ALUA) 等效的多路徑功能。

了解如何為 Rocky Linux 8.x 設定 NVMe over Fabrics (NVMe-oF) 主機。如需更多支援和功能信息，請參閱 ["Rocky Linux ONTAP支援與功能"](#)。

NVMe-oF 與 Rocky Linux 8.x 有以下已知限制：

- 目前不支援使用 NVMe-oF 協定的 SAN 啟動。
- 在 Rocky Linux 8.x 中，NVMe-oF 主機上的核心內 NVMe 多路徑預設為停用；您必須手動啟用它。
- 由於已知問題，NVMe/TCP 可作為技術預覽。

步驟 1：安裝 Rocky Linux 和 NVMe 軟體並驗證您的配置

若要為 NVMe-oF 設定主機，您需要安裝主機和 NVMe 軟體包，啟用多路徑，並驗證主機 NQN 設定。

步驟

1. 在伺服器上安裝 Rocky Linux 8.x。安裝完成後，請確認您執行的是所需的 Rocky Linux 8.x 核心：

```
uname -r
```

Rocky Linux 核心版本範例：

```
5.14.0-570.12.1.el9_6.x86_64
```

2. 安裝「NVMe-CLI (NVMe - CLI) 套件：

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

以下範例顯示了 nvme-cli 軟體包版本：

```
nvme-cli-2.11-5.el9.x86_64
```

3. 安裝 libnvme 套件：

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

下面的例子展示了 `libnvme` 軟體包版本：

```
libnvme-1.11.1-1.el9.x86_64
```

4. 在 Rocky Linux 主機上，檢查 hostnqn 字串 /etc/nvme/hostnqn：

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

下面的例子展示了 `hostnqn` 版本：

```
nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
```

5. 在 ONTAP 系統中，驗證以下資訊：`hostnqn` 字串匹配 `hostnqn` ONTAP 陣列中對應子系統的字串：

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



```

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.

```



如果是 hostnqn 字串不相符、請使用 `vserver modify` 命令來更新 hostnqn 對應 ONTAP 陣列子系統上的字串、以符合 hostnqn 字串來源 `/etc/nvme/hostnqn` 在主機上。

步驟 2：設定 NVMe/FC 和 NVMe/TCP

使用 Broadcom/Emulex 或 Marvell/QLogic 適配器配置 NVMe/FC，或使用手動發現和連接操作來設定 NVMe/TCP。

NVMe/FC - 博通/Emulex

為 Broadcom / Emulex 介面卡設定 NVMe / FC 。

步驟

1. 確認您使用的是支援的介面卡機型：

a. 顯示模型名稱：

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

您應該會看到下列輸出：

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

b. 顯示模型描述：

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

您應該會看到類似以下範例的輸出：

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

2. 驗證您使用的是建議的Broadcom lpfc 韌體與收件匣驅動程式：

a. 顯示韌體版本：

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

該命令返回韌體版本：

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

b. 顯示收件匣驅動程式版本：

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

以下範例顯示了驅動程式版本：

```
0:14.4.0.2
```

如需支援的介面卡驅動程式和韌體版本的最新清單，請參閱["互通性對照表工具"](#)。

3. 驗證的預期輸出是否 `lpfc_enable_fc4_type` 設置為 `3`：

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

4. 確認您可以檢視啟動器連接埠：

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

以下範例顯示連接埠標識：

```
0x100000109bf044b1  
0x100000109bf044b2
```

5. 驗證啟動器連接埠是否在線上：

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

您應該會看到下列輸出：

```
Online  
Online
```

6. 確認已啟用 NVMe / FC 啟動器連接埠、且目標連接埠可見：

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

顯示範例

```
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
ffffffffffffe498bd
          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
ffffffffffffe5d74f
          abort 000014a5 noxri 00000000 nondlp 0000006a qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 000014a5 Err 0002149a
```

NVMe/FC - Marvell/QLogic

為 Marvell/QLogic 介面卡設定 NVMe / FC 。

步驟

1. 確認您執行的是支援的介面卡驅動程式和韌體版本：

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

以下範例顯示了驅動程式和韌體版本：

```
QLE2742 FW:v9.14.00 DVR:v10.02.09.200-k  
QLE2742 FW:v9.14.00 DVR:v10.02.09.200-k
```

2. 請確認 `ql2xnvmeenable` 已設定。這可讓 Marvell 介面卡作為 NVMe / FC 啟動器運作：

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

預期輸出為 1。

NVMe / TCP

NVMe/TCP 協定不支援自動連線操作。相反，您可以透過執行 NVMe/TCP 來發現 NVMe/TCP 子系統和命名空間 `connect` 或者 `connect-all` 手動操作。

步驟

1. 檢查啟動器連接埠是否可以跨支援的 NVMe/TCP LIF 取得發現日誌頁面資料：

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

顯示範例

```
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.0f4bale74eb611ef9f50d039eab6cb6d: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.0f4bale74eb611ef9f50d039eab6cb6d: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.0f4bale74eb611ef9f50d039eab6cb6d: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.0f4bale74eb611ef9f50d039eab6cb6d: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.0f4bale74eb611ef9f50d039eab6cb6d: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.0f4bale74eb611ef9f50d039eab6cb6d: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.0f4bale74eb611ef9f50d039eab6cb6d: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.0f4bale74eb611ef9f50d039eab6cb6d: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.0f4bale74eb611ef9f50d039eab6cb6d: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.0f4bale74eb611ef9f50d039eab6cb6d: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.0f4bale74eb611ef9f50d039eab6cb6d: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.0f4bale74eb611ef9f50d039eab6cb6d: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.0f4bale74eb611ef9f50d039eab6cb6d: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. 確認其他的 NVMe / TCP 啟動器目標 LIF 組合能夠成功擷取探索記錄頁面資料：

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

顯示範例

```
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. 執行 `nvme connect-all` 跨所有節點支援的 NVMe / TCP 啟動器目標生命體執行命令：

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

顯示範例

```
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
```

步驟 3：可選，啟用 NVMe/FC 的 1MB I/O。

您可以為配置了 Broadcom 适配器的 NVMe/FC 啟用 1MB 大小的 I/O 請求。ONTAP 在識別控制器資料中報告的最大資料傳輸大小 (MDTS) 為 8。這表示最大 I/O 要求大小最多可達 1MB。要發出 1MB 大小的 I/O 請求，您需要增加 ``lpfc_sg_seg_cnt`` 參數從預設值 64 更改為 256。



這些步驟不適用於 Qlogic NVMe / FC 主機。

步驟

1. 將 ``lpfc_sg_seg_cnt`` 參數設定為 256：

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

```
options lpfc lpfc_sg_seg_cnt=256
```

2. 執行 ``dracut -f`` 命令，然後重新啟動主機。
3. 確認的值 ``lpfc_sg_seg_cnt`` 為 256：

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

步驟 4：驗證多路徑配置

驗證核心內建 NVMe 多重路徑狀態，ANA 狀態和 ONTAP 命名空間是否適用於 NVMe 組態。

步驟

1. 確認已啟用核心內建 NVMe 多重路徑：

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

您應該會看到下列輸出：

```
Y
```

2. 驗證個別 ONTAP 命名空間的適當 NVMe 設定（例如、模型設定為 NetApp ONTAP 控制器、負載平衡 iopolicing 設定為循環）是否正確反映在主機上：

- a. 顯示子系統：

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

您應該會看到下列輸出：

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

- b. 顯示策略：

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

您應該會看到下列輸出：

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

3. 確認已在主機上建立並正確探索命名空間：

```
nvme list
```

顯示範例

```
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. 確認每個路徑的控制器狀態均為有效、且具有正確的ANA狀態：

NVMe / FC

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

顯示範例

```
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
```

顯示範例

```
nvme-subsys5 - NQN=nqn.1992-08.com.netapp:sn.0f4bale74eb611ef9f50d039eab6cb6d: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. 驗證NetApp外掛程式是否顯示每ONTAP 個版本名稱空間裝置的正確值：

欄位

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

顯示範例

Device	Vserver	Namespace	Path
/dev/nvme1n1	linux_tcnvme_iscsi		

/vol/tcpcnvme_1_0_0/tcpcnvme_ns			
NSID	UUID		Size

1	5f7f630d-8ea5-407f-a490-484b95b15dd6		21.47GB

JSON

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

顯示範例

```
{
  "ONTAPdevices": [
    {
      "Device": "/dev/nvme1n1",
      "Vserver": "linux_tcnvme_iscsi",
      "Namespace_Path": "/vol/tcpcnvme_1_0_0/tcpcnvme_ns",
      "NSID": 1,
      "UUID": "5f7f630d-8ea5-407f-a490-484b95b15dd6",
      "Size": "21.47GB",
      "LBA_Data_Size": 4096,
      "Namespace_Size": 5242880
    },
  ],
}
```

步驟 5：查看已知問題

這些是已知問題：

NetApp錯誤ID	標題	說明
"1479047"	Rocky Linux 8.x NVMe-oF 主機建立重複的持久發現控制器	在 NVMe-oF 主機上，您可以使用「nvme discover -p」指令建立持久發現控制器 (PDC)。但是，如果您在 NVMe-oF 主機上執行 Rocky Linux 8.x，則每次執行「nvme discover -p」時都會建立重複的 PDC。使用此指令時，每個發起方-目標組合只能建立一個 PDC。但是，如果您在 NVMe-oF 主機上執行 Rocky Linux 8.x，每次執行 "nvme discover -p" 時都會建立重複的 PDC。這會導致主機和目標裝置上的資源被不必要地消耗掉。

版權資訊

Copyright © 2026 NetApp, Inc. 版權所有。台灣印製。非經版權所有人事先書面同意，不得將本受版權保護文件的任何部分以任何形式或任何方法（圖形、電子或機械）重製，包括影印、錄影、錄音或儲存至電子檢索系統中。

由 NetApp 版權資料衍伸之軟體必須遵守下列授權和免責聲明：

此軟體以 NETAPP「原樣」提供，不含任何明示或暗示的擔保，包括但不限於有關適售性或特定目的適用性之擔保，特此聲明。於任何情況下，就任何已造成或基於任何理論上責任之直接性、間接性、附隨性、特殊性、懲罰性或衍生性損害（包括但不限於替代商品或服務之採購；使用、資料或利潤上的損失；或企業營運中斷），無論是在使用此軟體時以任何方式所產生的契約、嚴格責任或侵權行為（包括疏忽或其他）等方面，NetApp 概不負責，即使已被告知有前述損害存在之可能性亦然。

NetApp 保留隨時變本文所述之任何產品的權利，恕不另行通知。NetApp 不承擔因使用本文所述之產品而產生的責任或義務，除非明確經過 NetApp 書面同意。使用或購買此產品並不會在依據任何專利權、商標權或任何其他 NetApp 智慧財產權的情況下轉讓授權。

本手冊所述之產品受到一項（含）以上的美國專利、國外專利或申請中專利所保障。

有限權利說明：政府機關的使用、複製或公開揭露須受 DFARS 252.227-7013（2014 年 2 月）和 FAR 52.227-19（2007 年 12 月）中的「技術資料權利 - 非商業項目」條款 (b)(3) 小段所述之限制。

此處所含屬於商業產品和 / 或商業服務（如 FAR 2.101 所定義）的資料均為 NetApp, Inc. 所有。根據本協議提供的所有 NetApp 技術資料和電腦軟體皆屬於商業性質，並且完全由私人出資開發。美國政府對於該資料具有非專屬、非轉讓、非轉授權、全球性、有限且不可撤銷的使用權限，僅限於美國政府為傳輸此資料所訂合約所允許之範圍，並基於履行該合約之目的方可使用。除非本文另有規定，否則未經 NetApp Inc. 事前書面許可，不得逕行使用、揭露、重製、修改、履行或展示該資料。美國政府授予國防部之許可權利，僅適用於 DFARS 條款 252.227-7015(b)（2014 年 2 月）所述權利。

商標資訊

NETAPP、NETAPP 標誌及 <http://www.netapp.com/TM> 所列之標章均為 NetApp, Inc. 的商標。文中所涉及的所有其他公司或產品名稱，均為其各自所有者的商標，不得侵犯。