



# **Rocky Linux**

## ONTAP SAN Host Utilities

NetApp  
January 30, 2026

This PDF was generated from <https://docs.netapp.com/ja-jp/ontap-sanhost/nvme-rockylinux-supported-features.html> on January 30, 2026. Always check [docs.netapp.com](https://docs.netapp.com) for the latest.

# 目次

Rocky Linux . . . . .	1
Rocky Linux のONTAPサポートと機能について学ぶ . . . . .	1
次の手順 . . . . .	1
ONTAPストレージでNVMe-oF用にRocky Linux 10.xを構成する . . . . .	2
手順1：必要に応じてSANブートを有効にします。 . . . . .	2
ステップ2: Rocky LinuxとNVMeソフトウェアをインストールし、構成を確認する . . . . .	2
ステップ3: NVMe/FCとNVMe/TCPを構成する . . . . .	4
ステップ4: オプションでNVMe/FCの1MB I/Oを有効にする . . . . .	12
ステップ5: NVMeブートサービスを確認する . . . . .	13
ステップ6: マルチパス構成を確認する . . . . .	14
ステップ7: 安全なインバンド認証を設定する . . . . .	18
手順8：既知の問題を確認する . . . . .	25
ONTAPストレージでNVMe-oF用にRocky Linux 9.xを構成する . . . . .	25
手順1：必要に応じてSANブートを有効にします。 . . . . .	25
ステップ2: Rocky LinuxとNVMeソフトウェアをインストールし、構成を確認する . . . . .	25
ステップ3: NVMe/FCとNVMe/TCPを構成する . . . . .	27
ステップ4: オプションでNVMe/FCの1MB I/Oを有効にする . . . . .	39
ステップ5: NVMeブートサービスを確認する . . . . .	40
ステップ6: マルチパス構成を確認する . . . . .	41
ステップ7: 安全なインバンド認証を設定する . . . . .	45
手順8：既知の問題を確認する . . . . .	53
ONTAPストレージで NVMe-oF 用に Rocky Linux 8.x を構成する . . . . .	53
ステップ1: Rocky LinuxとNVMeソフトウェアをインストールし、構成を確認する . . . . .	53
ステップ2: NVMe/FCとNVMe/TCPを構成する . . . . .	55
ステップ3: オプションでNVMe/FCの1MB I/Oを有効にする . . . . .	66
ステップ4: マルチパス構成を確認する . . . . .	67
ステップ5: 既知の問題を確認する . . . . .	71

# Rocky Linux

## Rocky Linux のONTAPサポートと機能について学ぶ

NVMe over Fabrics (NVMe-oF) を使用したホスト構成でサポートされる機能は、ONTAPおよびRocky Linuxのバージョンによって異なります。

特徴	Rocky Linuxホストバージョン	ONTAPのバージョン
RHELホストとONTAPコントローラ間のNVMe/TCP 経由の安全なインバンド認証がサポートされています。	9.3以降	9.12.1以降
NVMe/TCPはネイティブを使用して名前空間を提供します `nvme-cli` パッケージ	8.2以降	9.10.1以降
NVMe/TCPは完全にサポートされているエンタープライズ機能です	9.0以降	9.10.1以降
NVMe と SCSI トラフィックは、NVMe-oF 名前空間の場合は NVMe マルチパス、SCSI LUN の場合は dm-multipath を使用して、同じホスト上でサポートされます。	8.2以降	9.4以降

ONTAPは、システムセットアップで実行されているONTAPのバージョンに関係なく、次のSANホスト機能をサポートします。

特徴	Rocky Linuxホストバージョン
ネイティブNVMeマルチパスはデフォルトで有効になっています	10.0以降
SANブートはNVMe/FCプロトコルを使用して有効化されます	9.4以降
その `nvme-cli` パッケージには自動接続スクリプトが含まれており、サードパーティのスクリプトは不要になります。	8.2以降
`nvme-cli` パッケージ内のネイティブudevルールは、NVMeマルチパスのラウンドロビンロードバランシングを提供します	8.2以降



サポートされている構成の詳細については、"Interoperability Matrix Tool"。

### 次の手順

Rocky Linux のバージョンが..	について学ぶ..
10シリーズ	"Rocky Linux 10.x 用の NVMe の設定"
9シリーズ	"Rocky Linux 9.x 用の NVMe の設定"
8シリーズ	"Rocky Linux 8.x 用の NVMe の設定"

## ONTAPストレージでNVMe-oF用にRocky Linux 10.xを構成する

Rocky Linux ホストは、非対称名前空間アクセス (ANA) を備えた NVMe over Fibre Channel (NVMe/FC) および NVMe over TCP (NVMe/TCP) プロトコルをサポートします。ANA は、iSCSI および FCP 環境における非対称論理ユニット アクセス (ALUA) と同等のマルチパス機能を提供します。

Rocky Linux 10.x 用に NVMe over Fabrics (NVMe-oF) ホストを構成する方法を学習します。詳細なサポートと機能情報については、["Rocky Linux ONTAPのサポートと機能"](#)。

Rocky Linux 10.x の NVMe-oF には、次の既知の制限があります。

- その `nvme disconnect-all` このコマンドはルートファイルシステムとデータファイルシステムの両方を切断し、システムが不安定になる可能性があります。NVMe-TCP または NVMe-FC 名前空間を介して SAN から起動するシステムではこれを発行しないでください。

### 手順1：必要に応じてSANブートを有効にします。

SAN ブートを使用するようにホストを構成すると、展開が簡素化され、スケーラビリティが向上します。使用["Interoperability Matrix Tool"](#)Linux OS、ホストバスアダプタ (HBA)、HBA フームウェア、HBA ブート BIOS、およびONTAPバージョンが SAN ブートをサポートしていることを確認します。

#### 手順

1. ["NVMe名前空間を作成し、ホストにマッピングする"](#)。
  2. SAN ブート名前空間がマップされているポートに対して、サーバー BIOS で SAN ブートを有効にします。
- HBA BIOS を有効にする方法については、ベンダー固有のマニュアルを参照してください。
3. ホストを再起動し、OS が起動して実行されていることを確認します。

### ステップ2: Rocky LinuxとNVMeソフトウェアをインストールし、構成を確認する

NVMe-oF 用にホストを構成するには、ホストおよび NVMe ソフトウェア パッケージをインストールし、マルチパスを有効にして、ホストの NQN 構成を確認する必要があります。

#### 手順

1. サーバーに Rocky Linux 10.x をインストールします。インストールが完了したら、必要な Rocky Linux 10.x カーネルが実行されていることを確認します。

```
uname -r
```

Rocky Linux カーネルバージョンの例:

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

2. 「nvme-cli」パッケージをインストールします。

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

次の例は、`nvme-cli`パッケージバージョン:

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

3. をインストールします libnvme パッケージ:

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

次の例は、`libnvme`パッケージバージョン:

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

4. ホスト上で、hostnqn文字列を確認します。 /etc/nvme/hostnqn :

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

次の例は、`hostnqn`値:

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

5. ONTAPシステムで、`hostnqn`文字列が一致する `hostnqn`ONTAPアレイ上の対応するサブシステムの文字列:

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

例を示します

```
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.
```



もし `hostnqn` 文字列が一致しない場合は、`vserver modify` 更新するコマンド `hostnqn` 対応するONTAPストレージシステムサブシステムの文字列を `hostnqn` 文字列から `/etc/nvme/hostnqn` ホスト上。

### ステップ3: NVMe/FCとNVMe/TCPを構成する

Broadcom/Emulex または Marvell/QLogic アダプタを使用して NVMe/FC を構成するか、手動の検出および接続操作を使用して NVMe/TCP を構成します。

## NVMe/FC - プロードコム/エミュレックス

Broadcom/Emulexアダプタ用にNVMe/FCを設定します。

### 手順

1. サポートされているアダプタモデルを使用していることを確認します。

- a. モデル名を表示します。

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

次の出力が表示されます。

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

- b. モデルの説明を表示します。

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

次の例のような出力が表示されます。

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

2. 推奨されるBroadcomを使用していることを確認します lpfcc フームウェアおよび受信トレイドライバ:

- a. フームウェアのバージョンを表示します。

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

このコマンドはフームウェアのバージョンを返します。

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

- b. 受信トレイのドライバーのバージョンを表示します。

```
cat /sys/module/lpfcc/version
```

次の例は、ドライバーのバージョンを示しています。

```
0:14.4.0.6
```

サポートされているアダプタドライバおよびファームウェアバージョンの最新リストについては、を参照してください["Interoperability Matrix Tool"](#)。

3. 確認します `lpfc_enable_fc4_type` がに設定されます 3 :

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

4. イニシエータポートを表示できることを確認します。

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

次のような出力が表示されます：

```
0x2100f4c7aa0cd7c2  
0x2100f4c7aa0cd7c3
```

5. イニシエータポートがオンラインであることを確認します。

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

次の出力が表示されます。

```
Online  
Online
```

6. NVMe/FCイニシエータポートが有効になっており、ターゲットポートが認識されることを確認します。

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

例を示します

```
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
fffffffffffffe498bd
    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 000000007af4434 Issue 000000007ae31b83 OutIO
fffffffffffffe5d74f
    abort 000014a5 noxri 00000000 nondlp 0000006a qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 000014a5 Err 0002149a
```

## NVMe/FC - マーベル/QLogic

Marvell/QLogicアダプタ用にNVMe/FCを設定します。

### 手順

1. サポートされているアダプタ ドライバーとファームウェア バージョンを使用していることを確認します。

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

次の例は、ドライバーとファームウェアのバージョンを示しています。

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

2. 確認します `ql2xnvmeenable` が設定されます。これにより、MarvellアダプタをNVMe/FCイニシエータとして機能させることができます。

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

想定される出力は1です。

## NVMe/FC

NVMe/TCP プロトコルは自動接続操作をサポートしていません。代わりに、NVMe/TCPサブシステムと名前空間をNVMe/TCPコマンドで検出することができます。`'connect'` または `'connect-all'` 手動で操作します。

### 手順

1. イニシエーター ポートが、サポートされている NVMe/TCP LIF 全体で検出口格 ページ データを得できることを確認します。

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

例を示します

```
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
adrifam: 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
adrifam: 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
adrifam: 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
adrifam: 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
adrifam: 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
adrifam: 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
adrifam: 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
adrifam: 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
```

- 他の NVMe/TCP イニシエーターとターゲット LIF の組み合わせで検出ログ ページ データを正常に取得できることを確認します。

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

例を示します

```
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
```

- を実行します nvme connect-all ノード全体でサポートされているすべてのNVMe/TCPイニシエータ/ターゲットLIFを対象としたコマンド：

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

例を示します

```
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
```

Rocky Linux 9.4以降、NVMe/TCPの設定は`ctrl\_loss\_tmo timeout`自動的に「オフ」に設定されます。その結果、次のようにになります

- 再試行回数に制限はありません（無期限再試行）。
- 特定の設定を手動で行う必要はありません`ctrl\_loss\_tmo timeout`使用時の持続時間`nvme connect`または`nvme connect-all`コマンド（オプション`-I`）。
- NVMe/TCP コントローラーは、パス障害が発生した場合でもタイムアウトが発生せず、無期限に接続されたままになります。

#### ステップ4: オプションでNVMe/FCの1MB I/Oを有効にする

ONTAPは、識別コントローラ データで最大データ転送サイズ (MDTS) が8であると報告します。つまり、最大I/O要求サイズは1MBまでになります。Broadcom NVMe/FCホストに1MBのI/Oリクエストを発行するには、`lpfc`の値`lpfc\_sg\_seg\_cnt`パラメータをデフォルト値の64から256に変更します。



この手順は、Qlogic NVMe/FCホストには適用されません。

手順

- `lpfc\_sg\_seg\_cnt`パラメータを256に設定します。

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

次の例のような出力が表示されます。

```
options lpfc lpfc_sg_seg_cnt=256
```

- コマンドを実行し`dracut -f`、ホストをリブートします。
- の値が256であることを確認し`lpfc\_sg\_seg\_cnt`ます。

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

## ステップ5: NVMeブートサービスを確認する

その `nvmefc-boot-connections.service` そして `nvmf-autoconnect.service` NVMe/FCに含まれるブートサービス `nvme-cli` パッケージはシステムの起動時に自動的に有効になります。

起動が完了したら、`nvmefc-boot-connections.service` そして `nvmf-autoconnect.service` ブート サービスが有効になっています。

手順

1. が有効であることを確認し `nvmf-autoconnect.service` ます。

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

出力例を表示します。

```
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. が有効であることを確認し `nvmefc-boot-connections.service` ます。

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

出力例を表示します。

```
nvmefc-boot-connections.service - Auto-connect to subsystems on FC-NVME devices found during boot
   Loaded: loaded (/usr/lib/systemd/system/nvmefc-boot-connections.service; enabled; preset: enabled)
     Active: inactive (dead) since Tue 2025-06-10 01:08:36 EDT; 2h 59min ago
       PID: 7090 (code=exited, status=0/SUCCESS)
         CPU: 30ms

Jun 10 01:08:36 localhost systemd[1]: Starting Auto-connect to subsystems on FC-NVME devices found during boot...
Jun 10 01:08:36 localhost systemd[1]: nvmefc-boot-connections.service: Deactivated successfully.
Jun 10 01:08:36 localhost systemd[1]: Finished Auto-connect to subsystems on FC-NVME devices found during boot.
```

## ステップ6: マルチパス構成を確認する

カーネル内のNVMeマルチパスステータス、ANAステータス、およびONTAPネームスペースがNVMe-oF構成に対して正しいことを確認します。

手順

1. カーネル内NVMeマルチパスが有効になっていることを確認します。

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

次の出力が表示されます。

```
Y
```

2. 該当するONTAPネームスペースの適切なNVMe-oF設定（modelをNetApp ONTAPコントローラに設定し、load balancing iopolicyをラウンドロビンに設定するなど）がホストに正しく反映されていることを確認します。
  - a. サブシステムを表示します。

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

次の出力が表示されます。

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

- b. ポリシーを表示します。

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

次の出力が表示されます。

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

3. ネームスペースが作成され、ホストで正しく検出されたことを確認します。

```
nvme list
```

例を示します

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

  

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

4. 各パスのコントローラの状態がliveであり、正しいANAステータスが設定されていることを確認します。

## NVMe/FC

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

例を示します

```
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/FC

```
nvme list-subsys /dev/nvme4n2
```

## 例を示します

```
nvme-subsy4 - NQN=nqn.1992-
08.com.netapp:sn.64e65e6caae711ef9668d039ea951c46:subsystem.nvme
4
hostnqn=nqn.2014-08.org.nvmeexpress:uuid:4c4c4544-
0035-5910-804b-c2c04f444d33
\
+- nvme102 tcp
traddr=192.168.21.20,trsvcid=4420,host_traddr=192.168.21.1,src_a
ddr=192.168.21.1 live non-optimized
+- nvme151 tcp
traddr=192.168.21.21,trsvcid=4420,host_traddr=192.168.21.1,src_a
ddr=192.168.21.1 live optimized
+- nvme4 tcp
traddr=192.168.20.20,trsvcid=4420,host_traddr=192.168.20.1,src_a
ddr=192.168.20.1 live non-optimized
+- nvme53 tcp
traddr=192.168.20.21,trsvcid=4420,host_traddr=192.168.20.1,src_a
ddr=192.168.20.1 live optimized
```

5. ネットアッププラグインで、ONTAP ネームスペースデバイスごとに正しい値が表示されていることを確認します。

## 列 (Column)

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

例を示します

Device	Vserver	Namespace	Path
/dev/nvme10n1	vs_tcp_rockylinux10		/vol/vol10/ns10

  

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

## JSON

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

例を示します

```
{
  "ONTAPdevices": [
    {
      "Device": "/dev/nvme10n1",
      "Vserver": "vs_tcp_rockylinux10",
      "Namespace_Path": "/vol/vol10/ns10",
      "NSID": 1,
      "UUID": "bbf51146-fc64-4197-b8cf-8a24f6f359b3",
      "Size": "21.47GB",
      "LBA_Data_Size": 4096,
      "Namespace_Size": 5242880
    }
  ]
}
```

## ステップ7: 安全なインバンド認証を設定する

Rocky Linux 10.x ホストとONTAPコントローラ間の NVMe/TCP 経由の安全なインバンド認証がサポートされます。

各ホストまたはコントローラは、`DH-HMAC-CHAP`安全な認証を設定するためのキー。`DH-HMAC-CHAP`キーは、NVMe ホストまたはコントローラの NQN と管理者によって設定された認証シークレットの組み合わせです。ピアを認証するには、NVMe ホストまたはコントローラがピアに関連付けられたキーを認識する必要があります。

#### 手順

CLI または設定 JSON ファイルを使用して、安全なインバンド認証を設定します。サブシステムごとに異なる DHCHAP キーを指定する必要がある場合は、config JSON ファイルを使用する必要があります。

## CLI の使用

CLIを使用してセキュアなインバンド認証を設定します。

1. ホストNQNを取得します。

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

2. Rocky Linux 10.x ホストの dhchap キーを生成します。

次の出力は、`gen-dhchap-key`コマンドパラメータ:

```
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
```

次の例では、HMACが3に設定されたランダムDHCHAPキー（SHA-512）が生成されます。

```
nvme gen-dhchap-key -m 3 -n nqn.2014-08.org.nvexpress:uuid:4c4c4544-0035-5910-804b-c2c04f444d33 DHHC-1:03:7zf8I9gaRcDWH3tCH5vLGaojzPIvwNWusBfKdpJa+hialaKDKJQ2o53pX3wYM9xdv5DtKNNhJInZ7X8wU2RQpQIngc=:
```

3. ONTAPコントローラで、ホストを追加し、両方のDHCHAPキーを指定します。

```
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. ホストは、単方向と双方向の2種類の認証方式をサポートします。ホストで、ONTAPコントローラに接続し、選択した認証方式に基づいてDHCHAPキーを指定します。

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

5. 検証する nvme connect authentication ホストとコントローラのDHCHAPキーを確認してコマンドを実行します。

- a. ホストDHCHAPキーを確認します。

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

に、単方向設定の出力例を示します。

```
cat /sys/class/nvme-subsystem/nvme-subsy1/nvme*/dhchap_secret  
DHHC-  
1:03:fMCrJharXUOqRoIsOEaG6m2PH1yYvu5+z3jTmzEKUBcWu26I33b93b  
i12WR09XDho/ld3L45J+0FeCsStBEAfhYgkQU=:  
DHHC-  
1:03:fMCrJharXUOqRoIsOEaG6m2PH1yYvu5+z3jTmzEKUBcWu26I33b93b  
i12WR09XDho/ld3L45J+0FeCsStBEAfhYgkQU=:  
DHHC-  
1:03:fMCrJharXUOqRoIsOEaG6m2PH1yYvu5+z3jTmzEKUBcWu26I33b93b  
i12WR09XDho/ld3L45J+0FeCsStBEAfhYgkQU=:  
DHHC-  
1:03:fMCrJharXUOqRoIsOEaG6m2PH1yYvu5+z3jTmzEKUBcWu26I33b93b  
i12WR09XDho/ld3L45J+0FeCsStBEAfhYgkQU=:
```

- b. コントローラのDHCHAPキーを確認します。

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

に、双方向設定の出力例を示します。

```
cat /sys/class/nvme-subsystem/nvme-
subsys6/nvme*/dhchap_ctrl_secret
DHHC- 1:03:7zf8I9gaRcDWH3tCH5vLGaoyjzPIvwNWusBfKdpJa+hia
1aKDKJQ2o53pX3wYM9xdv5DtKNNhJInZ7X8wU2RQpQIng=:

DHHC- 1:03:7zf8I9gaRcDWH3tCH5vLGaoyjzPIvwNWusBfKdpJa+hia
1aKDKJQ2o53pX3wYM9xdv5DtKNNhJInZ7X8wU2RQpQIng=:

DHHC- 1:03:7zf8I9gaRcDWH3tCH5vLGaoyjzPIvwNWusBfKdpJa+hia
1aKDKJQ2o53pX3wYM9xdv5DtKNNhJInZ7X8wU2RQpQIng=:

DHHC- 1:03:7zf8I9gaRcDWH3tCH5vLGaoyjzPIvwNWusBfKdpJa+hia
1aKDKJQ2o53pX3wYM9xdv5DtKNNhJInZ7X8wU2RQpQIng=:
```

## JSON

ONTAPコントローラ上で複数のNVMeサブシステムが利用可能な場合は、`/etc/nvme/config.json`ファイルに`nvme connect-all`指示。

使用`-o`JSON ファイルを生成するオプション。詳細な構文オプションについては、NVMe connect-all のマニュアル ページを参照してください。

1. JSON ファイルを設定します。



次の例では、`dhchap\_key`に対応する `dhchap\_secret` そして  
`dhchap\_ctrl\_key`に対応する `dhchap\_ctrl\_secret`。

## 例を示します

```
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+hia1aDKJQ2o53pX3
wYM9xdv5DtKNNhJInZ7X8wU2RQpQIngc=:",
    "subsystems": [
      {
        "nqn": "nqn.1992-
08.com.netapp:sn.127ade26168811f0a50ed039eab69ad3:subsystem.inba
nd_unidirectional",
        "ports": [
          {
            "transport": "tcp",
            "traddr": "192.168.20.17",
            "host_traddr": "192.168.20.1",
            "trsvcid": "4420"
          },
          {
            "transport": "tcp",
            "traddr": "192.168.20.18",
            "host_traddr": "192.168.20.1",
            "trsvcid": "4420"
          },
          {
            "transport": "tcp",
            "traddr": "192.168.21.18",
            "host_traddr": "192.168.21.1",
            "trsvcid": "4420"
          },
          {
            "transport": "tcp",
            "traddr": "192.168.21.17",
            "host_traddr": "192.168.21.1",
            "trsvcid": "4420"
          }
        ]
      }
    ]
  }
]
```

- config jsonファイルを使用してONTAPコントローラに接続します。

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

例を示します

```
traddr=192.168.20.20 is already connected
traddr=192.168.20.21 is already connected
```

3. 各サブシステムのそれぞれのコントローラに対して dhchap シークレットが有効になっていることを確認します。

- a. ホストDHCHAPキーを確認します。

```
cat /sys/class/nvme-subsystem/nvme-subsy0/nvme0/dhchap_secret
```

次の例は、dhchap キーを示しています。

```
DHHC-1:03:7zf8I9gaRcDWH3tCH5vLGaoyjzPIvwNWusBfKdpJa+hia1
aDKKJQ2o53pX3wYM9xdv5DtKNNhJInZ7X8wU2RQpQIngc=:
```

- b. コントローラのDHCHAPキーを確認します。

```
cat /sys/class/nvme-subsystem/nvme-
subsyst0/nvme0/dhchap_ctrl_secret
```

次の例のような出力が表示されます。

DHHC-1:03:fMCrJharXUOqRoIsOEaG6m2PH1yYvu5+z3jT  
mzEKUbcWu26I33b93bi12WR09XDho/ld3L45J+0FeCsStBEAfhYgkQU=:

## 手順8：既知の問題を確認する

既知の問題はありません。

## ONTAPストレージでNVMe-oF用にRocky Linux 9.xを構成する

Rocky Linux ホストは、非対称名前空間アクセス (ANA) を備えた NVMe over Fibre Channel (NVMe/FC) および NVMe over TCP (NVMe/TCP) プロトコルをサポートします。ANA は、iSCSI および FCP 環境における非対称論理ユニット アクセス (ALUA) と同等のマルチパス機能を提供します。

Rocky Linux 9.x 用に NVMe over Fabrics (NVMe-oF) ホストを構成する方法を学習します。詳細なサポートと機能情報については、["Rocky Linux ONTAPのサポートと機能"](#)。

Rocky Linux 9.x の NVMe-oF には、次の既知の制限があります。

- その `nvme disconnect-all` このコマンドはルートファイルシステムとデータファイルシステムの両方を切断し、システムが不安定になる可能性があります。NVMe-TCP または NVMe-FC 名前空間を介して SAN から起動するシステムではこれを発行しないでください。

## 手順1：必要に応じてSANブートを有効にします。

SAN ブートを使用するようにホストを構成すると、展開が簡素化され、スケーラビリティが向上します。使用["Interoperability Matrix Tool"](#)Linux OS、ホストバスアダプタ (HBA)、HBA フームウェア、HBA ブート BIOS、およびONTAPバージョンが SAN ブートをサポートしていることを確認します。

### 手順

- "[NVMe名前空間を作成し、ホストにマッピングする](#)"。
  - SAN ブート名前空間がマップされているポートに対して、サーバー BIOS で SAN ブートを有効にします。
- HBA BIOS を有効にする方法については、ベンダー固有のマニュアルを参照してください。
- ホストを再起動し、OS が起動して実行されていることを確認します。

## ステップ2: Rocky LinuxとNVMeソフトウェアをインストールし、構成を確認する

NVMe-oF 用にホストを構成するには、ホストおよび NVMe ソフトウェア パッケージをインストールし、マルチパスを有効にして、ホストの NQN 構成を確認する必要があります。

### 手順

- サーバーに Rocky Linux 9.x をインストールします。インストールが完了したら、必要な Rocky Linux 9.x カーネルが実行されていることを確認します。

```
uname -r
```

Rocky Linux カーネルバージョンの例:

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

2. 「nvme-cli」パッケージをインストールします。

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

次の例は、nvme-cli パッケージのバージョンを示しています。

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

3. をインストールします libnvme パッケージ：

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

次の例は、`libnvme`パッケージバージョン:

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

4. Rocky Linuxホストで、hostnqn文字列を確認します。 /etc/nvme/hostnqn :

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

次の例は、`hostnqn`バージョン:

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

5. ONTAPシステムで、`hostnqn`文字列が一致する `hostnqn`ONTAPアレイ上の対応するサブシステムの文字列:

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

例を示します

```
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.
```



状況に応じて hostnqn 文字列が一致しない場合は、を使用してください vserver modify コマンドを使用してを更新します hostnqn 対応するONTAP アレイサブシステムで、に一致する文字列を指定します hostnqn から文字列 /etc/nvme/hostnqn ホスト。

### ステップ3: NVMe/FCとNVMe/TCPを構成する

Broadcom/Emulex または Marvell/QLogic アダプタを使用して NVMe/FC を構成するか、手動の検出および接続操作を使用して NVMe/TCP を構成します。

## NVMe/FC - プロードコム/エミュレックス

Broadcom/Emulexアダプタ用にNVMe/FCを設定します。

### 手順

1. サポートされているアダプタモデルを使用していることを確認します。

- a. モデル名を表示します。

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

次の出力が表示されます。

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

- b. モデルの説明を表示します。

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

次の例のような出力が表示されます。

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

2. 推奨されるBroadcomを使用していることを確認します lpf‌c フームウェアおよび受信トレイドライバ:

- a. フームウェアのバージョンを表示します。

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

このコマンドはフームウェアのバージョンを返します。

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

- b. 受信トレイのドライバーのバージョンを表示します。

```
cat /sys/module/lpf‌c/version
```

次の例は、ドライバーのバージョンを示しています。

```
0:14.4.0.6
```

サポートされているアダプタドライバおよびファームウェアバージョンの最新リストについては、を参照してください["Interoperability Matrix Tool"](#)。

3. 確認します `lpfc_enable_fc4_type` がに設定されます 3 :

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

4. イニシエータポートを表示できることを確認します。

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

次の例はポート ID を示しています。

```
0x2100f4c7aa0cd7c2  
0x2100f4c7aa0cd7c3
```

5. イニシエータポートがオンラインであることを確認します。

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

次の出力が表示されます。

```
Online  
Online
```

6. NVMe/FCイニシエータポートが有効になっており、ターゲットポートが認識されることを確認します。

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

例を示します

```
NVME Initiator Enabled
XRI Dist lpfco Total 6144 IO 5894 ELS 250
NVME LPORT lpfco 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 lpfcl Total 6144 IO 5894 ELS 250
NVME LPORT lpfcl 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 lpfcc Total 6144 IO 5894 ELS 250
NVME LPORT lpfcc WWPN x100000109bf044b1 WWNN x200000109bf044b1
DID x022a00 ONLINE
NVME RPORT      WWPN x200bd039eaa7dfc8 WWNN x2008d039eaa7dfc8
DID x021319 TARGET DISCSRVC ONLINE
NVME RPORT      WWPN x2155d039eaa7dfc8 WWNN x2154d039eaa7dfc8
DID x02130f TARGET DISCSRVC ONLINE
NVME RPORT      WWPN x2001d039eaa7dfc8 WWNN x2000d039eaa7dfc8
DID x021310 TARGET DISCSRVC ONLINE
NVME RPORT      WWPN x200dd039eaa7dfc8 WWNN x2008d039eaa7dfc8
DID x020b15 TARGET DISCSRVC ONLINE
NVME RPORT      WWPN x2156d039eaa7dfc8 WWNN x2154d039eaa7dfc8
DID x020b0d TARGET DISCSRVC ONLINE
NVME RPORT      WWPN x2003d039eaa7dfc8 WWNN x2000d039eaa7dfc8
DID x020b10 TARGET DISCSRVC ONLINE
```

```

NVME Statistics
LS: Xmt 0000003049 Cmpl 0000003049 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 0000000018f9450b Issue 0000000018f5de57 OutIO
fffffffffffffc994c
    abort 000036d3 noxri 00000313 nondlp 00000c8d qdepth
00000000 wqerr 00000064 err 00000000
FCP CMPL: xb 000036d1 Err 000fef0f

NVME Initiator Enabled
XRI Dist lpfc3 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc3 WWPN x100000109bf044b2 WWNN x200000109bf044b2
DID x021b00 ONLINE
NVME RPORT      WWPN x2062d039eaa7dfc8 WWNN x2008d039eaa7dfc8
DID x022915 TARGET DISCSRVC ONLINE
NVME RPORT      WWPN x2157d039eaa7dfc8 WWNN x2154d039eaa7dfc8
DID x02290f TARGET DISCSRVC ONLINE
NVME RPORT      WWPN x2002d039eaa7dfc8 WWNN x2000d039eaa7dfc8
DID x022910 TARGET DISCSRVC ONLINE
NVME RPORT      WWPN x2065d039eaa7dfc8 WWNN x2008d039eaa7dfc8
DID x020119 TARGET DISCSRVC ONLINE
NVME RPORT      WWPN x2158d039eaa7dfc8 WWNN x2154d039eaa7dfc8
DID x02010d TARGET DISCSRVC ONLINE
NVME RPORT      WWPN x2004d039eaa7dfc8 WWNN x2000d039eaa7dfc8
DID x020110 TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 0000002f2c Cmpl 0000002f2c Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000001aab3eb5 Issue 000000001aab4373 OutIO
fffffffffffffc04be
    abort 000035cc noxri 0000038c nondlp 000009e3 qdepth
00000000 wqerr 00000082 err 00000000
FCP CMPL: xb 000035cc Err 000fcfc0

```

## NVMe/FC - マーベル/QLogic

Marvell/QLogicアダプタ用にNVMe/FCを設定します。

### 手順

1. サポートされているアダプタ ドライバーとファームウェア バージョンを使用していることを確認します。

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

次の例は、ドライバーとファームウェアのバージョンを示しています。

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

2. 確認します `ql2xnvmeenable` が設定されます。これにより、MarvellアダプタをNVMe/FCイニシエータとして機能させることができます。

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

想定される出力は1です。

## NVMe/FC

NVMe/TCP プロトコルは自動接続操作をサポートしていません。 NVMe/TCP サブシステムと名前空間を検出するには、NVMe/TCP 接続または `connect-all` 操作を手動で実行する必要があります。

### 手順

1. イニシエーター ポートが、サポートされている NVMe/TCP LIF 全体で検出口格 ページ データを取得できることを確認します。

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

例を示します

```
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
adrifam: ipv4
subtype: current discovery subsystem
treq: not specified
portid: 4
trsvcid: 8009
subnqn: nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr: 192.168.2.25
eflags: explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 1=====
trtype: tcp
adrifam: ipv4
subtype: current discovery subsystem
treq: not specified
portid: 2
trsvcid: 8009
subnqn: nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr: 192.168.1.25
eflags: explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 2=====
trtype: tcp
adrifam: ipv4
subtype: current discovery subsystem
treq: not specified
portid: 5
trsvcid: 8009
subnqn: nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr: 192.168.2.24
eflags: explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 3=====
```

```
trtype: tcp
adrifam: 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
adrifam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 4
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.
Unidirectional_DHCP_NONE_1_3
traddr: 192.168.2.25
eflags: none
sectype: none
=====Discovery Log Entry 5=====
trtype: tcp
adrifam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 2
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.
Unidirectional_DHCP_NONE_1_4
traddr: 192.168.1.25
eflags: none
sectype: none
=====Discovery Log Entry 6=====
trtype: tcp
adrifam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 5
trsvcid: 4420
subnqn: nqn.1992-
```

```
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.  
Unidirectional_DHCP_NONE_1_5  
traddr: 192.168.2.24  
eflags: none  
sectype: none  
=====Discovery Log Entry 7=====  
trtype: tcp  
adrfam: ipv4  
subtype: nvme subsystem  
treq: not specified  
portid: 1  
trsvcid: 4420  
subnqn: nqn.1992-  
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.  
Unidirectional_DHCP_2_2  
traddr: 192.168.1.24  
eflags: none  
sectype: none  
=====Discovery Log Entry 8=====  
trtype: tcp  
adrfam: ipv4  
subtype: nvme subsystem  
treq: not specified  
portid: 4  
trsvcid: 4420  
subnqn: nqn.1992-  
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.  
Unidirectional_DHCP_2_3  
traddr: 192.168.2.25  
eflags: none  
sectype: none  
=====Discovery Log Entry 9=====  
trtype: tcp  
adrfam: ipv4  
subtype: nvme subsystem  
treq: not specified  
portid: 2  
trsvcid: 4420  
subnqn: nqn.1992-  
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.  
Unidirectional_DHCP_2_5  
traddr: 192.168.1.25  
eflags: none  
sectype: none  
=====Discovery Log Entry 10=====  
trtype: tcp
```

```
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 5
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.
Bidirectional_DHCP_2_2
traddr: 192.168.2.24
eflags: none
sectype: none
=====Discovery Log Entry 11=====
trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 1
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.
Bidirectional_DHCP_2_3
traddr: 192.168.1.24
eflags: none
sectype: none
=====Discovery Log Entry 12=====
trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 4
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.
Bidirectional_DHCP_2_3
traddr: 192.168.2.25
eflags: none
sectype: none
=====Discovery Log Entry 13=====
trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 2
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.
```

```
Bidirectional_DHCP_NONE_2_4
traddr: 192.168.1.25
eflags: none
sectype: none
=====Discovery Log Entry 14=====
trtype: tcp
adrifam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 5
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.
Bidirectional_DHCP_NONE_2_5
traddr: 192.168.2.24
eflags: none
sectype: none
=====Discovery Log Entry 15=====
trtype: tcp
adrifam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 1
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.
Bidirectional_DHCP_NONE_2_6
traddr: 192.168.1.24
eflags: none
sectype: none
=====Discovery Log Entry 16=====
trtype: tcp
adrifam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 4
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.
Bidirectional_DHCP_NONE_2_7
traddr: 192.168.2.25
eflags: none
sectype: none
=====Discovery Log Entry 17=====
trtype: tcp
adrifam: ipv4
```

```

subtype: nvme subsystem
treq: not specified
portid: 2
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.
Bidirectional_DHCP_NONE_2_8
traddr: 192.168.1.25
eflags: none
sectype: none
=====Discovery Log Entry 18=====
trtype: tcp
adrifam: 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
adrifam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 1
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.
Bidirectional_DHCP_NONE_2_9
traddr: 192.168.1.24
eflags: none
sectype: none

```

2. 他の NVMe/TCP イニシエーターとターゲット LIF の組み合わせで検出ログ ページ データを正常に取得できることを確認します。

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

例を示します

```
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. を実行します `nvme connect-all` ノード全体でサポートされているすべてのNVMe/TCPイニシエータ/ターゲットLIFを対象としたコマンド：

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

例を示します

```
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
```

Rocky Linux 9.4以降、NVMe/TCPの設定は`ctrl\_loss\_tmo timeout`自動的に「オフ」に設定されます。その結果、次のようにになります

- 再試行回数に制限はありません（無期限再試行）。
- 特定の設定を手動で行う必要はありません`ctrl\_loss\_tmo timeout`使用時の持続時間`nvme connect`または`nvme connect-all`コマンド（オプション`-I`）。
- NVMe/TCP コントローラーは、パス障害が発生した場合でもタイムアウトが発生せず、無期限に接続されたままになります。

#### ステップ4: オプションでNVMe/FCの1MB I/Oを有効にする

ONTAPは、識別コントローラデータで最大データ転送サイズ(MDTS)が8であると報告します。つまり、最大I/O要求サイズは1MBまでになります。Broadcom NVMe/FCホストに1MBのI/Oリクエストを発行するには、`lpfc`の値`lpfc\_sg\_seg\_cnt`パラメータをデフォルト値の64から256に変更します。



この手順は、Qlogic NVMe/FCホストには適用されません。

##### 手順

- `lpfc\_sg\_seg\_cnt`パラメータを256に設定します。

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

次の例のような出力が表示されます。

```
options lpfc lpfc_sg_seg_cnt=256
```

2. コマンドを実行し dracut -f、ホストをリブートします。
3. の値が256であることを確認し `lpfc\_sg\_seg\_cnt` ます。

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

## ステップ5: NVMeブートサービスを確認する

その `nvmefc-boot-connections.service` そして `nvmf-autoconnect.service` NVMe/FCに含まれるブートサービス `nvme-cli` パッケージはシステムの起動時に自動的に有効になります。

起動が完了したら、 `nvmefc-boot-connections.service` そして `nvmf-autoconnect.service` ブート サービスが有効になっています。

手順

1. が有効であることを確認し `nvmf-autoconnect.service` ます。

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

出力例を表示します。

```
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. が有効であることを確認し `nvmefc-boot-connections.service` ます。

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

出力例を表示します。

```
nvmefc-boot-connections.service - Auto-connect to subsystems on FC-NVME devices found during boot
   Loaded: loaded (/usr/lib/systemd/system/nvmefc-boot-connections.service; enabled; preset: enabled)
     Active: inactive (dead) since Tue 2025-06-10 01:08:36 EDT; 2h 59min ago
       PID: 7090 (code=exited, status=0/SUCCESS)
         CPU: 30ms

Jun 10 01:08:36 localhost systemd[1]: Starting Auto-connect to subsystems on FC-NVME devices found during boot...
Jun 10 01:08:36 localhost systemd[1]: nvmefc-boot-connections.service: Deactivated successfully.
Jun 10 01:08:36 localhost systemd[1]: Finished Auto-connect to subsystems on FC-NVME devices found during boot.
```

## ステップ6: マルチパス構成を確認する

カーネル内のNVMeマルチパスステータス、ANAステータス、およびONTAPネームスペースがNVMe-oF構成に対して正しいことを確認します。

手順

1. カーネル内NVMeマルチパスが有効になっていることを確認します。

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

次の出力が表示されます。

```
Y
```

2. 該当するONTAPネームスペースの適切なNVMe-oF設定（modelをNetApp ONTAPコントローラに設定し、load balancing iopolicyをラウンドロビンに設定するなど）がホストに正しく反映されていることを確認します。
  - a. サブシステムを表示します。

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

次の出力が表示されます。

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

b. ポリシーを表示します。

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

次の出力が表示されます。

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

3. ネームスペースが作成され、ホストで正しく検出されたことを確認します。

```
nvme list
```

例を示します

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

  

Namespace	Usage	Format	FW	Rev
1	21.47 GB / 21.47 GB	4 KiB + 0 B	FFFFFFFFFF	

4. 各パスのコントローラの状態がliveであり、正しいANAステータスが設定されていることを確認します。

## NVMe/FC

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

例を示します

```
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/FC

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

## 例を示します

```
nvme-subsyss5 - NQN=nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme
_tcp_3
hostnqn=nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0035-5910-804b-
b5c04f444d33
iopolicy=round-robin
\
+- nvme13 tcp
traddr=192.168.2.25,trsvcid=4420,host_traddr=192.168.2.31,
src_addr=192.168.2.31 live optimized
+- nvme14 tcp
traddr=192.168.2.24,trsvcid=4420,host_traddr=192.168.2.31,
src_addr=192.168.2.31 live non-optimized
+- nvme5 tcp
traddr=192.168.1.25,trsvcid=4420,host_traddr=192.168.1.31,
src_addr=192.168.1.31 live optimized
+- nvme6 tcp
traddr=192.168.1.24,trsvcid=4420,host_traddr=192.168.1.31,
src_addr=192.168.1.31 live non-optimized
```

5. ネットアッププラグインで、ONTAP ネームスペースデバイスごとに正しい値が表示されていることを確認します。

## 列 (Column)

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

例を示します

Device	Vserver	Namespace	Path
/dev/nvme1n1	linux_tcnvme_iscsi		
/vol/tcpnvme_1_0_0		tcpnvme_ns	

  

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

## JSON

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

例を示します

```
{  
  "ONTAPdevices": [  
    {  
      "Device": "/dev/nvme1n1",  
      "Vserver": "linux_tcnvme_iscsi",  
      "Namespace_Path": "/vol/tcpnvme_1_0_0/tcpnvme_ns",  
      "NSID": 1,  
      "UUID": "5f7f630d-8ea5-407f-a490-484b95b15dd6",  
      "Size": "21.47GB",  
      "LBA_Data_Size": 4096,  
      "Namespace_Size": 5242880  
    },  
  ]  
}
```

## ステップ7: 安全なインバンド認証を設定する

Rocky Linux 9x ホストとONTAPコントローラ間の NVMe/TCP 経由の安全なインバンド認証がサポートされます。

各ホストまたはコントローラは、`DH-HMAC-CHAP`安全な認証を設定するためのキー。`DH-HMAC-CHAP`キーは、NVMe ホストまたはコントローラの NQN と管理者によって設定された認証シークレットの組み合わせです。ピアを認証するには、NVMeホストまたはコントローラがピアに関連付けられたキーを認識する必要があります。

## 手順

CLI または設定 JSON ファイルを使用して、安全なインバンド認証を設定します。サブシステムごとに異なるDHCHAPキーを指定する必要がある場合は、config JSONファイルを使用する必要があります。

## CLI の使用

CLIを使用してセキュアなインバンド認証を設定します。

1. ホストNQNを取得します。

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

2. Rocky Linux 9.x ホストの dhchap キーを生成します。

次の出力は、`gen-dhchap-key`コマンドパラメータ:

```
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
```

次の例では、HMACが3に設定されたランダムDHCHAPキー（SHA-512）が生成されます。

```
nvme gen-dhchap-key -m 3 -n nqn.2014-08.org.nvmexpress:uuid:e6dade64-216d-11ec-b7bb-7ed30a5482c3
DHHC-
1:03:wSpuuKbBHTzC0W9JZxMBsYd9JFV8Si9aDh22k2BR/4m852vH7KGlrJeMpzhmyjDWoo0PJMJ6yZsTeEpGkDHMHQ25+g=:
```

3. ONTAPコントローラで、ホストを追加し、両方のDHCHAPキーを指定します。

```
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. ホストは、単方向と双方向の2種類の認証方式をサポートします。ホストで、ONTAPコントローラに接続し、選択した認証方式に基づいてDHCHAPキーを指定します。

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

5. 検証する nvme connect authentication ホストとコントローラのDHCHAPキーを確認してコマンドを実行します。

- a. ホストDHCHAPキーを確認します。

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

に、単方向設定の出力例を示します。

```
cat /sys/class/nvme-subsystem/nvme-subsy1/nvme*/dhchap_secret  
DHHC-1:01:CNxTYq73T9vJk0JpOfDBZrhDCqpWBN4XVZI5WxwPgDUieHAI:  
DHHC-1:01:CNxTYq73T9vJk0JpOfDBZrhDCqpWBN4XVZI5WxwPgDUieHAI:  
DHHC-1:01:CNxTYq73T9vJk0JpOfDBZrhDCqpWBN4XVZI5WxwPgDUieHAI:  
DHHC-1:01:CNxTYq73T9vJk0JpOfDBZrhDCqpWBN4XVZI5WxwPgDUieHAI:
```

- b. コントローラのDHCHAPキーを確認します。

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

に、双方向設定の出力例を示します。

```
cat /sys/class/nvme-subsystem/nvme-  
subsy6/nvme*/dhchap_ctrl_secret  
DHHC-  
1:03:wSpuuKbBHTzC0W9JZxMBsYd9JFV8Si9aDh22k2BR/4m852vH7KG1rJeMp  
zhmyjDWOo0PJ JM6yZsTeEpGkDHMHQ255+g=:  
DHHC-  
1:03:wSpuuKbBHTzC0W9JZxMBsYd9JFV8Si9aDh22k2BR/4m852vH7KG1rJeMp  
zhmyjDWOo0PJ JM6yZsTeEpGkDHMHQ255+g=:  
DHHC-  
1:03:wSpuuKbBHTzC0W9JZxMBsYd9JFV8Si9aDh22k2BR/4m852vH7KG1rJeMp  
zhmyjDWOo0PJ JM6yZsTeEpGkDHMHQ255+g=:  
DHHC-  
1:03:wSpuuKbBHTzC0W9JZxMBsYd9JFV8Si9aDh22k2BR/4m852vH7KG1rJeMp  
zhmyjDWOo0PJ JM6yZsTeEpGkDHMHQ255+g=:
```

## JSON

ONTAPコントローラ上で複数のNVMeサブシステムが利用可能な場合は、`/etc/nvme/config.json`ファイルに`nvme connect-all`指示。

使用`-o`JSON ファイルを生成するオプション。詳細な構文オプションについては、NVMe connect-all のマニュアル ページを参照してください。

1. JSON ファイルを設定します。



次の例では、`dhchap\_key`対応する `dhchap\_secret` そして  
`dhchap\_ctrl\_key`対応する `dhchap\_ctrl\_secret`。

## 例を示します

```
cat /etc/nvme/config.json
[
{
    "hostnqn": "nqn.2014-08.org.nvmexpress:uuid:9796c1ec-0d34-11eb-
b6b2-3a68dd3bab57",
    "hostid": "b033cd4fd6db4724adb48655bfb55448",
    "dhchap_key": " DHHC-
1:01:CNxTYq73T9vJk0Jp0fDBZrhDCqpWBN4XVZI5WxwPgDUieHAI:",
},
{
    "hostnqn": "nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0035-5910-
804b-b5c04f444d33",
    "subsystems": [
        {
            "nqn": "nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.bidi
r_DHCP",
            "ports": [
                {
                    "transport": "tcp",
                    "traddr": " 192.168.1.24 ",
                    "host_traddr": " 192.168.1.31 ",
                    "trsvcid": "4420",
                    "dhchap_ctrl_key": "DHHC-
1:03:
wSpuuKbBHTzC0W9JZxMBsYd9JFV8Si9aDh22k2BR/4m852vH7KGlrJeMpzhmyjDW
Oo0PJ JM6yZsTeEpGkDHMHQ255+g=",
                },
                {
                    "transport": "tcp",
                    "traddr": " 192.168.1.25 ",
                    "host_traddr": " 192.168.1.31",
                    "trsvcid": "4420",
                    "dhchap_ctrl_key": "DHHC-
1:03:
wSpuuKbBHTzC0W9JZxMBsYd9JFV8Si9aDh22k2BR/4m852vH7KGlrJeMpzhmyjDW
Oo0PJ JM6yZsTeEpGkDHMHQ255+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. config jsonファイルを使用してONTAPコントローラに接続します。

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

## 例を示します

```
already connected to hostnqn=nqn.2014-
08.org.nvmeexpress: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.nvmeexpress: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.nvmeexpress: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.nvmeexpress: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. 各サブシステムの各コントローラでDHCHAPシークレットが有効になっていることを確認します。

- a. ホストDHCHAPキーを確認します。

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

次の例は、dhchap キーを示しています。

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

- b. コントローラのDHCHAPキーを確認します。

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

次の例のような出力が表示されます。

```
DHHC-
1:03:wSpuuKbBHTzC0W9JZxMBsYd9JFV8Si9aDh22k2BR/4m852vH7KG1rJeMpzhmyjD
WOo0PJMJ6yZsTeEpGkDHMHQ255+g=:
```

## 手順8：既知の問題を確認する

既知の問題はありません。

## ONTAPストレージで NVMe-oF 用に Rocky Linux 8.x を構成する

Rocky Linux ホストは、非対称名前空間アクセス (ANA) を備えた NVMe over Fibre Channel (NVMe/FC) および NVMe over TCP (NVMe/TCP) プロトコルをサポートします。ANA は、iSCSI および FCP 環境における非対称論理ユニット アクセス (ALUA) と同等のマルチパス機能を提供します。

Rocky Linux 8.x 用に NVMe over Fabrics (NVMe-oF) ホストを構成する方法を学習します。詳細なサポートと機能情報については、["Rocky Linux ONTAPのサポートと機能"](#)。

Rocky Linux 8.x の NVMe-oF には、次の既知の制限があります。

- NVMe-oF プロトコルを使用した SAN ブートは現在サポートされていません。
- Rocky Linux 8.x の NVMe-oF ホストでは、カーネル内 NVMe マルチパスはデフォルトで無効になっているため、手動で有効にする必要があります。
- 既知の問題のため、NVMe/TCP はテクノロジー プレビューとして利用できます。

## ステップ1: Rocky LinuxとNVMeソフトウェアをインストールし、構成を確認する

NVMe-oF 用にホストを構成するには、ホストおよび NVMe ソフトウェア パッケージをインストールし、マルチパスを有効にして、ホストの NQN 構成を確認する必要があります。

### 手順

1. サーバーに Rocky Linux 8.x をインストールします。インストールが完了したら、必要な Rocky Linux 8.x カーネルが実行されていることを確認します。

```
uname -r
```

Rocky Linux カーネルバージョンの例:

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

2. 「nvme-cli」パッケージをインストールします。

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

次の例は、nvme-cli パッケージのバージョンを示しています。

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

3. をインストールします libnvme パッケージ：

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

次の例は、`libnvme` パッケージバージョン：

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

4. Rocky Linuxホストで、hostnqn文字列を確認します。 /etc/nvme/hostnqn :

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

次の例は、`hostnqn` バージョン：

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

5. ONTAPシステムで、`hostnqn`文字列が一致する `hostnqn`ONTAPアレイ上の対応するサブシステムの文字列:

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

例を示します

```
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.
```



状況に応じて hostnqn 文字列が一致しない場合は、を使用してください vserver modify コマンドを使用してを更新します hostnqn 対応するONTAP アレイサブシステムで、に一致する文字列を指定します hostnqn から文字列 /etc/nvme/hostnqn ホスト。

## ステップ2: NVMe/FCとNVMe/TCPを構成する

Broadcom/Emulex または Marvell/QLogic アダプタを使用して NVMe/FC を構成するか、手動の検出および接続操作を使用して NVMe/TCP を構成します。

## NVMe/FC - プロードコム/エミュレックス

Broadcom/Emulexアダプタ用にNVMe/FCを設定します。

### 手順

1. サポートされているアダプタモデルを使用していることを確認します。

- a. モデル名を表示します。

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

次の出力が表示されます。

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

- b. モデルの説明を表示します。

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

次の例のような出力が表示されます。

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

2. 推奨されるBroadcomを使用していることを確認します lpf‌c フームウェアおよび受信トレイドライバ:

- a. フームウェアのバージョンを表示します。

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

このコマンドはフームウェアのバージョンを返します。

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

- b. 受信トレイのドライバーのバージョンを表示します。

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

次の例は、ドライバーのバージョンを示しています。

```
0:14.4.0.2
```

サポートされているアダプタドライバおよびファームウェアバージョンの最新リストについては、を参照してください["Interoperability Matrix Tool"](#)。

3. の想定される出力がに設定されている `3` ことを確認し `lpfc\_enable\_fc4\_type` ます。

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

4. イニシエータポートを表示できることを確認します。

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

次の例はポート ID を示しています。

```
0x100000109bf044b1  
0x100000109bf044b2
```

5. イニシエータポートがオンラインであることを確認します。

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

次の出力が表示されます。

```
Online  
Online
```

6. NVMe/FCイニシエータポートが有効になっており、ターゲットポートが認識されることを確認します。

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

例を示します

```
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
fffffffffffffe498bd
    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 000000007af4434 Issue 000000007ae31b83 OutIO
fffffffffffffe5d74f
    abort 000014a5 noxri 00000000 nondlp 0000006a qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 000014a5 Err 0002149a
```

## NVMe/FC - マーベル/QLogic

Marvell/QLogicアダプタ用にNVMe/FCを設定します。

### 手順

1. サポートされているアダプタドライバとファームウェアのバージョンが実行されていることを確認します。

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

次の例は、ドライバーとファームウェアのバージョンを示しています。

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

2. 確認します `ql2xnvmeenable` が設定されます。これにより、MarvellアダプタをNVMe/FCイニシエータとして機能させることができます。

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

想定される出力は1です。

## NVMe/FC

NVMe/TCP プロトコルは自動接続操作をサポートしていません。代わりに、NVMe/TCPサブシステムと名前空間をNVMe/TCPコマンドで検出することができます。`'connect'` または `'connect-all'` 手動で操作します。

### 手順

1. イニシエーター ポートが、サポートされている NVMe/TCP LIF 全体で検出口格 ページ データを得できることを確認します。

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

例を示します

```
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
adrifam: ipv4
subtype: current discovery subsystem
treq: not specified
portid: 4
trsvcid: 8009
subnqn: nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr: 192.168.2.25
eflags: explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 1=====
trtype: tcp
adrifam: ipv4
subtype: current discovery subsystem
treq: not specified
portid: 2
trsvcid: 8009
subnqn: nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr: 192.168.1.25
eflags: explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 2=====
trtype: tcp
adrifam: ipv4
subtype: current discovery subsystem
treq: not specified
portid: 5
trsvcid: 8009
subnqn: nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr: 192.168.2.24
eflags: explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 3=====
trtype: tcp
```

```
adrfam: ipv4
subtype: current discovery subsystem
treq: not specified
portid: 1
trsvcid: 8009
subnqn: nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:discovery
traddr: 192.168.1.24
eflags: explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 4=====
trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 4
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme
_tcp_1
traddr: 192.168.2.25
eflags: none
sectype: none
=====Discovery Log Entry 5=====
trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 2
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme
_tcp_1
traddr: 192.168.1.25
eflags: none
sectype: none
=====Discovery Log Entry 6=====
trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 5
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme
```

```
_tcp_1
traddr: 192.168.2.24
eflags: none
sectype: none
=====Discovery Log Entry 7=====
trtype: tcp
adrfrm: ipv4
subtype: nvme subsystem
treq: not specified
portid: 1
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme
_tcp_1
traddr: 192.168.1.24
eflags: none
sectype: none
=====Discovery Log Entry 8=====
trtype: tcp
adrfrm: ipv4
subtype: nvme subsystem
treq: not specified
portid: 4
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme
_tcp_4
traddr: 192.168.2.25
eflags: none
sectype: none
=====Discovery Log Entry 9=====
trtype: tcp
adrfrm: ipv4
subtype: nvme subsystem
treq: not specified
portid: 2
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme
_tcp_4
traddr: 192.168.1.25
eflags: none
sectype: none
=====Discovery Log Entry 10=====
trtype: tcp
adrfrm: ipv4
```

```
subtype: nvme subsystem
treq:    not specified
portid:   5
trsvcid: 4420
subnqn:   nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme
_tcp_4
traddr:  192.168.2.24
eflags:   none
sectype:  none
=====Discovery Log Entry 11=====
trtype:  tcp
adrifam: ipv4
subtype: nvme subsystem
treq:    not specified
portid:   1
trsvcid: 4420
subnqn:   nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme
_tcp_4
traddr:  192.168.1.24
eflags:   none
sectype:  none
=====Discovery Log Entry 12=====
trtype:  tcp
adrifam: ipv4
subtype: nvme subsystem
treq:    not specified
portid:   4
trsvcid: 4420
subnqn:   nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme
_tcp_3
traddr:  192.168.2.25
eflags:   none
sectype:  none
=====Discovery Log Entry 13=====
trtype:  tcp
adrifam: ipv4
subtype: nvme subsystem
treq:    not specified
portid:   2
trsvcid: 4420
subnqn:   nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme
_tcp_3
```

```
traddr: 192.168.1.25
eflags: none
sectype: none
=====Discovery Log Entry 14=====
trtype: tcp
adrifam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 5
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme
_tcp_3
traddr: 192.168.2.24
eflags: none
sectype: none
=====Discovery Log Entry 15=====
trtype: tcp
adrifam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 1
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme
_tcp_3
traddr: 192.168.1.24
eflags: none
sectype: none
=====Discovery Log Entry 16=====
trtype: tcp
adrifam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 4
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme
_tcp_2
traddr: 192.168.2.25
eflags: none
sectype: none
=====Discovery Log Entry 17=====
trtype: tcp
adrifam: 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
adrifam: 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
adrifam: 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

```

- NVMe/TCPイニシエータとターゲットLIFの他の組み合わせで検出口ログページのデータを正常に取得できることを確認します。

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

例を示します

```
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. を実行します `nvme connect-all` ノード全体でサポートされているすべてのNVMe/TCPイニシエータ/ターゲットLIFを対象としたコマンド：

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

例を示します

```
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
```

### ステップ3: オプションで**NVMe/FCの1MB I/Oを有効にする**

Broadcomアダプタで構成されたNVMe/FCでは、1MBのI/Oリクエストを有効にすることができます。ONTAPは、コントローラ識別データで最大データ転送サイズ（MDTS）を8と報告します。1MBつまり、最大I/O要求サイズは1MBです。のI/Oリクエストを発行するには、`lpfc`の値を増やす必要があります。`lpfc\_sg\_seg\_cnt`パラメータをデフォルト値の64から256に変更します。



この手順は、Qlogic NVMe/FCホストには適用されません。

#### 手順

1. `lpfc\_sg\_seg\_cnt`パラメータを256に設定します。

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

```
options lpfc lpfc_sg_seg_cnt=256
```

2. コマンドを実行し `dracut -f`、ホストをリブートします。
3. の値が256であることを確認し `lpfc\_sg\_seg\_cnt` ます。

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

#### ステップ4: マルチパス構成を確認する

カーネル内のNVMeマルチパスステータス、ANAステータス、およびONTAPネームスペースがNVMe-oF構成に対して正しいことを確認します。

手順

1. カーネル内NVMeマルチパスが有効になっていることを確認します。

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

次の出力が表示されます。

```
Y
```

2. 該当するONTAPネームスペースの適切なNVMe-oF設定（modelをNetApp ONTAPコントローラに設定し、load balancing iopolicyをラウンドロビンに設定するなど）がホストに正しく反映されていることを確認します。

- a. サブシステムを表示します。

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

次の出力が表示されます。

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

- b. ポリシーを表示します。

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

次の出力が表示されます。

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

3. ネームスペースが作成され、ホストで正しく検出されたことを確認します。

```
nvme list
```

例を示します

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

  

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

4. 各パスのコントローラの状態がliveであり、正しいANAステータスが設定されていることを確認します。

## NVMe/FC

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

例を示します

```
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/FC

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

例を示します

```
nvme-subsyss5 - NQN=nqn.1992-
08.com.netapp:sn.0f4ba1e74eb611ef9f50d039eab6cb6d:subsystem.nvme
_tcp_3
hostnqn=nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0035-5910-804b-
b5c04f444d33
iopolicy=round-robin
\
+- nvme13 tcp
traddr=192.168.2.25,trsvcid=4420,host_traddr=192.168.2.31,
src_addr=192.168.2.31 live optimized
+- nvme14 tcp
traddr=192.168.2.24,trsvcid=4420,host_traddr=192.168.2.31,
src_addr=192.168.2.31 live non-optimized
+- nvme5 tcp
traddr=192.168.1.25,trsvcid=4420,host_traddr=192.168.1.31,
src_addr=192.168.1.31 live optimized
+- nvme6 tcp
traddr=192.168.1.24,trsvcid=4420,host_traddr=192.168.1.31,
src_addr=192.168.1.31 live non-optimized
```

5. ネットアッププラグインで、ONTAP ネームスペースデバイスごとに正しい値が表示されていることを確認します。

## 列 (Column)

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

例を示します

Device	Vserver	Namespace	Path
/dev/nvme1n1	linux_tcnvme_iscsi		
/vol/tcpnvme_1_0_0		tcpnvme_ns	

  

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

## JSON

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

例を示します

```
{  
  "ONTAPdevices": [  
    {  
      "Device": "/dev/nvme1n1",  
      "Vserver": "linux_tcnvme_iscsi",  
      "Namespace_Path": "/vol/tcpnvme_1_0_0/tcpnvme_ns",  
      "NSID": 1,  
      "UUID": "5f7f630d-8ea5-407f-a490-484b95b15dd6",  
      "Size": "21.47GB",  
      "LBA_Data_Size": 4096,  
      "Namespace_Size": 5242880  
    },  
  ]  
}
```

## ステップ5: 既知の問題を確認する

既知の問題は次のとおりです:

NetApp バグ ID	タイトル	説明
"1479047"	Rocky Linux 8.x NVMe-oF ホストは重複した永続的検出コントローラを作成します	NVMe-oF ホストでは、「nvme discover -p」コマンドを使用して永続検出コントローラ (PDC) を作成できます。ただし、NVMe-oF ホストで Rocky Linux 8.x を実行している場合は、「nvme discover -p」を実行するたびに重複した PDC が作成されます。このコマンドを使用する場合、イニシエーターとターゲットの組み合わせごとに 1 つの PDC のみを作成する必要があります。ただし、Rocky Linux 8.x をNVMe-oFホストで実行している場合は、「nvme discover -p」を実行するたびに重複したPDCが作成されます。これにより、ホストとターゲットの両方でリソースが不必要に使用されることになります。

## 著作権に関する情報

Copyright © 2026 NetApp, Inc. All Rights Reserved. Printed in the U.S.このドキュメントは著作権によって保護されています。著作権所有者の書面による事前承諾がある場合を除き、画像媒体、電子媒体、および写真複写、記録媒体、テープ媒体、電子検索システムへの組み込みを含む機械媒体など、いかなる形式および方法による複製も禁止します。

ネットアップの著作物から派生したソフトウェアは、次に示す使用許諾条項および免責条項の対象となります。

このソフトウェアは、ネットアップによって「現状のまま」提供されています。ネットアップは明示的な保証、または商品性および特定目的に対する適合性の暗示的保証を含み、かつこれに限定されないいかなる暗示的な保証も行いません。ネットアップは、代替品または代替サービスの調達、使用不能、データ損失、利益損失、業務中断を含み、かつこれに限定されない、このソフトウェアの使用により生じたすべての直接的損害、間接的損害、偶発的損害、特別損害、懲罰的損害、必然的損害の発生に対して、損失の発生の可能性が通知されていたとしても、その発生理由、根拠とする責任論、契約の有無、厳格責任、不法行為（過失またはそうでない場合を含む）にかかわらず、一切の責任を負いません。

ネットアップは、ここに記載されているすべての製品に対する変更を隨時、予告なく行う権利を保有します。ネットアップによる明示的な書面による合意がある場合を除き、ここに記載されている製品の使用により生じる責任および義務に対して、ネットアップは責任を負いません。この製品の使用または購入は、ネットアップの特許権、商標権、または他の知的所有権に基づくライセンスの供与とはみなされません。

このマニュアルに記載されている製品は、1つ以上の米国特許、その他の国の特許、および出願中の特許によって保護されている場合があります。

権利の制限について：政府による使用、複製、開示は、DFARS 252.227-7013（2014年2月）およびFAR 5225.227-19（2007年12月）のRights in Technical Data -Noncommercial Items（技術データ - 非商用品目に関する諸権利）条項の(b)(3)項、に規定された制限が適用されます。

本書に含まれるデータは商用製品および / または商用サービス（FAR 2.101の定義に基づく）に関係し、データの所有権はNetApp, Inc.にあります。本契約に基づき提供されるすべてのネットアップの技術データおよびコンピュータソフトウェアは、商用目的であり、私費のみで開発されたものです。米国政府は本データに対し、非独占的かつ移転およびサブライセンス不可で、全世界を対象とする取り消し不能の制限付き使用権を有し、本データの提供の根拠となった米国政府契約に関連し、当該契約の裏付けとする場合にのみ本データを使用できます。前述の場合を除き、NetApp, Inc.の書面による許可を事前に得ることなく、本データを使用、開示、転載、改変するほか、上演または展示することはできません。国防総省にかかる米国政府のデータ使用権については、DFARS 252.227-7015(b)項（2014年2月）で定められた権利のみが認められます。

## 商標に関する情報

NetApp、NetAppのロゴ、<http://www.netapp.com/TM>に記載されているマークは、NetApp, Inc.の商標です。その他の会社名と製品名は、それを所有する各社の商標である場合があります。