



# **RHEL**

## **ONTAP SAN Host Utilities**

NetApp  
January 29, 2026

# 目次

RHEL .....	1
RHELホストの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RHEL 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: RHEL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: オプションとして、udevルールのiopolicyを変更します。 .....	12
ステップ5: オプションでNVMe/FCの1MB I/Oを有効にする .....	13
ステップ6: NVMeブートサービスを確認する .....	14
ステップ7: マルチパス構成を確認する .....	15
ステップ8: 安全なインバンド認証を設定する .....	20
手順9：既知の問題を確認する .....	26
ONTAPストレージでNVMe-oF用にRHEL 9.xを構成する .....	26
手順1：必要に応じてSANブートを有効にします。 .....	26
ステップ2: RHELとNVMeソフトウェアをインストールし、構成を確認する .....	26
ステップ3: NVMe/FCとNVMe/TCPを構成する .....	28
ステップ4: オプションとして、udevルールのiopolicyを変更します。 .....	37
ステップ5: オプションでNVMe/FCの1MB I/Oを有効にする .....	38
ステップ6: NVMeブートサービスを確認する .....	39
ステップ7: マルチパス構成を確認する .....	40
ステップ8: 安全なインバンド認証を設定する .....	45
手順9：既知の問題を確認する .....	51
ONTAPストレージでNVMe-oF用にRHEL 8.xを構成する .....	52
手順1：必要に応じてSANブートを有効にします。 .....	52
ステップ2: RHELとNVMeソフトウェアをインストールし、構成を確認する .....	52
ステップ3: NVMe/FCとNVMe/TCPを構成する .....	54
ステップ4: オプションでNVMe/FCの1MB I/Oを有効にする .....	60
ステップ5: マルチパス構成を確認する .....	60
手順6：既知の問題を確認する .....	65

# RHEL

## RHELホストの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 RHEL のバージョンによって異なります。

特徴	RHEL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	RHEL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
`nvme-cli` パッケージ内のネイティブudevルールは、NVMeマルチパスのキュー深度ロード バランシングを提供します	9.6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

RHEL バージョンが ... の場合	... について学びましょう。
10シリーズ	<a href="#">"RHEL 10.x 用の NVMe の構成"</a>
9シリーズ	<a href="#">"RHEL 9.x 用の NVMe の構成"</a>
8シリーズ	<a href="#">"RHEL 8.x 用の NVMe の構成"</a>

## ONTAPストレージでNVMe-oF用にRHEL 10.xを構成する

Red Hat Enterprise Linux (RHEL) 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。

RHEL 10.x 用の 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、["RHEL ONTAPのサポートと機能"](#)。

RHEL 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 ファームウェア、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: RHEL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 RHEL 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 RHEL 10.x カーネルが実行されていることを確認します。

```
uname -r
```

RHEL カーネルバージョンの例:

```
6.12.0-124.8.1.el10_1.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-2.13-2.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-1.13-1.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`バージョン:

```
nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-c7c04f425633
```

5. ONTAPシステムで、`hostnqn`文字列が一致する `hostnqn`ONTAPストレージ システム上の対応するサブシステムの文字列:

```
::> vserver nvme subsystem host show -vserver vs_coexistence_QLE2872
```

例を示します

```
Vserver Subsystem Priority Host NQN
-----
vs_coexistence_QLE2872
    subsystem_1
        regular    nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-c7c04f425633
    subsystem_10
        regular    nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-c7c04f425633
    subsystem_11
        regular    nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-c7c04f425633
```



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 - ブロードコム/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。

```
SN1700E2P  
SN1700E2P
```

- 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。

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

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

- a. ファームウェアのバージョンを表示します。

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

このコマンドはファームウェアのバージョンを返します。

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

```
0:14.4.0.9
```

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：

```
0x10005cba2cfca7de  
0x10005cba2cfca7df
```

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 lpfc0 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc0 WWPN x10005cba2cfca7de WWNN x20005cba2cfca7de
DID x080f00 ONLINE
NVME RPORT          WWPN x2023d039eac03c33 WWNN x2021d039eac03c33
DID x082209 TARGET DISCSRVC ONLINE
NVME RPORT          WWPN x200ed039eac03c33 WWNN x200cd039eac03c33
DID x082203 TARGET DISCSRVC ONLINE
NVME RPORT          WWPN x2022d039eac03c33 WWNN x2021d039eac03c33
DID x082609 TARGET DISCSRVC ONLINE
NVME RPORT          WWPN x200dd039eac03c33 WWNN x200cd039eac03c33
DID x082604 TARGET DISCSRVC ONLINE
```

```
NVME Statistics
LS: Xmt 0000000501 Cmpl 0000000501 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 00000000000583b7 Issue 000000000005840d OutIO
0000000000000056
abort 0000010f noxri 00000000 nondlp 00000000 qdepth 00000000
wqerr 00000000 err 00000000
FCP CMPL: xb 0000010f Err 0000010f
```

```
NVME Initiator Enabled
XRI Dist lpfc1 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc1 WWPN x10005cba2cfca7df WWNN x20005cba2cfca7df
DID x080b00 ONLINE
NVME RPORT          WWPN x2024d039eac03c33 WWNN x2021d039eac03c33
DID x082309 TARGET DISCSRVC ONLINE
NVME RPORT          WWPN x200fd039eac03c33 WWNN x200cd039eac03c33
DID x082304 TARGET DISCSRVC ONLINE
NVME RPORT          WWPN x2025d039eac03c33 WWNN x2021d039eac03c33
DID x082708 TARGET DISCSRVC ONLINE
NVME RPORT          WWPN x2010d039eac03c33 WWNN x200cd039eac03c33
DID x082703 TARGET DISCSRVC ONLINE
```

```
NVME Statistics
LS: Xmt 00000006eb Cmpl 00000006eb Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000000004d600 Issue 000000000004d65f OutIO
000000000000005f
abort 000001c1 noxri 00000000 nondlp 00000000 qdepth 00000000
wqerr 00000000 err 00000000
FCP CMPL: xb 000001c1 Err 000001c2
```

## 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:v9.15.06 DVR:v10.02.09.400-k  
QLE2872 FW:v9.15.06 DVR:v10.02.09.400-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イニシエーター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 全体で検出ログ 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 192.168.20.21 -a 192.168.20.28
Discovery Log Number of Records 8, Generation counter 10
====Discovery Log Entry 0=====
trtype: tcp
adrfam: ipv4
subtype: current discovery subsystem
treq: not specified
portid: 8
trsvcid: 8009
subnqn: nqn.1992-
08.com.netapp:sn.17e32b6e8c7f11f09545d039eac03c33:discovery
traddr: 192.168.21.29
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: 6
trsvcid: 8009
subnqn: nqn.1992-
08.com.netapp:sn.17e32b6e8c7f11f09545d039eac03c33:discovery
traddr: 192.168.20.29
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: 7
trsvcid: 8009
subnqn: nqn.1992-
08.com.netapp:sn.17e32b6e8c7f11f09545d039eac03c33:discovery
traddr: 192.168.21.28
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:  5
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.17e32b6e8c7f11f09545d039eac03c33:discovery
traddr:  192.168.20.28
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 4=====
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  8
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.17e32b6e8c7f11f09545d039eac03c33:subsystem.Bidi
rectional_DHCP_1_0
traddr:  192.168.21.29
eflags:  none
sectype: none
=====Discovery Log Entry 5=====
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  6
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.17e32b6e8c7f11f09545d039eac03c33:subsystem.Bidi
rectional_DHCP_1_0
traddr:  192.168.20.29
eflags:  none
sectype: none
=====Discovery Log Entry 6=====
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  7
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.17e32b6e8c7f11f09545d039eac03c33:subsystem.Bidi

```

```

rectional_DHCP_1_0
traddr: 192.168.21.28
eflags: none
sectype: none
=====Discovery Log Entry 7=====
trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 5
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.17e32b6e8c7f11f09545d039eac03c33:subsystem.Bidi
rectional_DHCP_1_0
traddr: 192.168.20.28
eflags: none
sectype: non

```

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 192.168.20.21 -a 192.168.20.28
nvme discover -t tcp -w 192.168.21.21 -a 192.168.21.28
nvme discover -t tcp -w 192.168.20.21 -a 192.168.20.29
nvme discover -t tcp -w 192.168.21.21 -a 192.168.21.29

```

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イニシエーター/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  192.168.20.21    -a 192.168.20.28
nvme    connect-all -t    tcp -w  192.168.21.21    -a 192.168.21.28
nvme    connect-all -t    tcp -w  192.168.20.21    -a 192.168.20.29
nvme    connect-all -t    tcp -w  192.168.21.21    -a 192.168.21.29
```

RHEL 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 -l）。
- 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: オプションとして、**udev**ルールの*iopolicy*を変更します。

RHEL 10.0ではNVMe-oFのデフォルトの*iopolicy*を次のように設定します。 `round-robin`。RHEL 10.0を使用しており、*iopolicy*を次のように変更したい場合 `queue-depth`、`udev` ルール ファイルを次のように変更します。

手順

1. ルート権限でテキスト エディターで `udev` ルール ファイルを開きます。

```
/usr/lib/udev/rules.d/71-nvmf-netapp.rules
```

次の出力が表示されます。

```
vi /usr/lib/udev/rules.d/71-nvmf-netapp.rules
```

2. 次の例のルールに示すように、NetApp ONTAPコントローラーの *iopolicy* を設定する行を見つけます。

```
ACTION=="add", SUBSYSTEM=="nvme-subsystem", ATTR{subsysstype}=="nvm",
ATTR{model}=="NetApp ONTAP Controller", ATTR{iopolicy}="round-robin"
```

3. ルールを修正して `round-robin`` なる ``queue-depth:`

```
ACTION=="add", SUBSYSTEM=="nvme-subsystem", ATTR{subsysstype}=="nvm",  
ATTR{model}=="NetApp ONTAP Controller", ATTR{iopolicy}="queue-depth"
```

4. udev ルールを再読み込みし、変更を適用します。

```
udevadm control --reload  
udevadm trigger --subsystem-match=nvme-subsystem
```

5. サブシステムの現在の iopolicy を確認します。 <subsystem>を置き換えます。例: nvme-subsys0。

```
cat /sys/class/nvme-subsystem/<subsystem>/iopolicy
```

次の出力が表示されます。

```
queue-depth.
```



新しい iopolicy は、一致する NetApp ONTAP コントローラ デバイスに自動的に適用されます。再起動する必要はありません。

## ステップ5: 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 1 MB 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

1. `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6: NVMeブートサービスを確認する

その `nvme-fc-boot-connections.service` として `nvme-fc-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、`nvme-fc-boot-connections.service` として `nvme-fc-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 `nvme-fc-autoconnect.service` ます。

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

出力例を表示します。

```
nvme-fc-autoconnect.service - Connect NVMe-oF subsystems automatically
during boot
   Loaded: loaded (/usr/lib/systemd/system/nvme-
autoconnect.service; enabled; preset: disabled)
   Active: inactive (dead) since Sun 2025-10-12 19:41:15 IST; 1
day 1h ago
  Invocation: 7b5b99929c6b41199d493fa25b629f6c
    Main PID: 10043 (code=exited, status=0/SUCCESS)
   Mem peak: 2.9M
      CPU: 50ms

Oct 12 19:41:15 localhost.localdomain systemd[1]: Starting nvme-
autoconnect.service - Connect NVMe-oF subsystems automatically
during boot...
Oct 12 19:41:15 localhost.localdomain systemd[1]: nvme-
autoconnect.service: Deactivated successfully.
Oct 12 19:41:15 localhost.localdomain systemd[1]: Finished nvme-
autoconnect.service - Connect NVMe-oF subsystems automatically
during boot.
```

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

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



出力例を表示します。

```
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 Sun 2025-10-12 19:40:33 IST; 1
day 1h ago
    Invocation: 0ec258a9f8c342ffb82408086d409bc6
    Main PID: 4151 (code=exited, status=0/SUCCESS)
    Mem peak: 2.9M
    CPU: 17ms

Oct 12 19:40:33 localhost systemd[1]: Starting nvmeofc-boot-
connections.service - Auto-connect to subsystems on FC-NVME devices
found during boot...
Oct 12 19:40:33 localhost systemd[1]: nvmeofc-boot-
connections.service: Deactivated successfully.
Oct 12 19:40:33 localhost systemd[1]: Finished nvmeofc-boot-
connections.service - Auto-connect to subsystems on FC-NVME devices
found during boot.
```

## ステップ7: 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. それぞれのONTAP名前空間の適切な NVMe-oF 設定 (モデルをNetApp ONTAPコントローラに設定し、ロード バランシングiopolicy を queue-depth に設定するなど) がホストに正しく表示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-subsys*/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-subsys*/iopolicy
```

次の出力が表示されます。

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

2. 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          Generic          SN  
Model  
-----  
-----  
/dev/nvme11n1 /dev/ng11n1          810cqJXhgWtsAAAAAAAI  
NetApp ONTAP Controller  
  
Namespace  Usage          Format          FW Rev  
-----  
0x1        951.90 MB / 21.47 GB  4 KiB + 0 B    9.18.1
```

3. 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/nvme9n2
```

例を示します

```
nvme-subsys9 - NQN=nqn.1992-08.com.netapp:sn.7c34ab26675e11f0a6c0d039eac03c33:subsystem.subsystem_46
                hostnqn=nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-c7c04f425633
\
+- nvme105 fc traddr=nn-0x2018d039eac03c33:pn-0x201bd039eac03c33,host_traddr=nn-0x2000f4c7aa0cd7c3:pn-0x2100f4c7aa0cd7c3 live optimized
+- nvme107 fc traddr=nn-0x2018d039eac03c33:pn-0x2019d039eac03c33,host_traddr=nn-0x2000f4c7aa0cd7c2:pn-0x2100f4c7aa0cd7c2 live optimized
+- nvme42 fc traddr=nn-0x2018d039eac03c33:pn-0x201cd039eac03c33,host_traddr=nn-0x2000f4c7aa0cd7c3:pn-0x2100f4c7aa0cd7c3 live optimized
+- nvme44 fc traddr=nn-0x2018d039eac03c33:pn-0x201ad039eac03c33,host_traddr=nn-0x2000f4c7aa0cd7c2:pn-0x2100f4c7aa0cd7c2 live optimized
```

## NVMe/FC

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

例を示します

```
nvme-subsys4 - NQN=nqn.1992-08.com.netapp:sn.17e32b6e8c7f11f09545d039eac03c33:subsystem.Bidirectional_DHCP_1_0
                hostnqn=nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0054-5110-8039-c3c04f523034
\
+- nvme4 tcp
traddr=192.168.20.28,trsvcid=4420,host_traddr=192.168.20.21,src_addr=192.168.20.21 live optimized
+- nvme5 tcp
traddr=192.168.20.29,trsvcid=4420,host_traddr=192.168.20.21,src_addr=192.168.20.21 live optimized
+- nvme6 tcp
traddr=192.168.21.28,trsvcid=4420,host_traddr=192.168.21.21,src_addr=192.168.21.21 live optimized
+- nvme7 tcp
traddr=192.168.21.29,trsvcid=4420,host_traddr=192.168.21.21,src_addr=192.168.21.21 live optimized
```

4. 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	Subsystem
Namespace Path		
-----	-----	-----
/dev/nvme0n1	vs_nvme_sanboot_tcp	rhel_sanboot_tcp170
tcp_97		
NSID	UUID	Size
----	-----	-----
1	982c0f2a-6b8b-11f0-a6c0-d039eac03c33	322.12GB

## JSON

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

### 例を示します

```
{
  "ONTAPdevices":[
    {
      "Device":"/dev/nvme0n1",
      "Vserver":"vs_nvme_sanboot_tcp",
      "Subsystem":"rhel_sanboot_tcp170",
      "Namespace_Path":"tcp_97",
      "NSID":1,
      "UUID":"982c0f2a-6b8b-11f0-a6c0-d039eac03c33",
      "LBA_Size":4096,
      "Namespace_Size":322122547200,
      "UsedBytes":16285069312,
      "Version":"9.18.1"
    }
  ]
}
```

## ステップ8: 安全なインバンド認証を設定する

RHEL 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. RHEL 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.nvmexpress:uuid:4c4c4544-0054-5110-8039-c3c04f523034
DHHC-
1:03:AppJHkJygA6ZC4BxyQntJST+4k4IOv47MAJk0xBITwFOHIC2nV/uE04RoSpy1z2
SXYqNW1bhLe9hJ+MDHigGexaG2Ig=:
```

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-subsysX>/nvme*/dhchap_secret
```

に、単方向設定の出力例を示します。

```
cat /sys/class/nvme-subsystem/nvme-subsys4/nvme*/dhchap_secret
DHHC-1:01:2G71sg9PMO00h1Wf1g4QtP0XT11kREz0qVuLm2xvZdbaWR/g:
DHHC-1:01:2G71sg9PMO00h1Wf1g4QtP0XT11kREz0qVuLm2xvZdbaWR/g:
DHHC-1:01:2G71sg9PMO00h1Wf1g4QtP0XT11kREz0qVuLm2xvZdbaWR/g:
DHHC-1:01:2G71sg9PMO00h1Wf1g4QtP0XT11kREz0qVuLm2xvZdbaWR/g:
```

- b. コントローラのDHCHAPキーを確認します。

```
cat /sys/class/nvme-subsystem/<nvme-
subsysX>/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-
subsys4/nvme*/dhchap_ctrl_secret
DHHC-
1:03:5CgWULVnU5HUOWP1MNg95pkiUAwayiO+IvrALZR8HpeJIHw3xyHdGlTnv
EJ81HDjBb+fGteUgIn0fj8ASHZigkuFIx8=:
DHHC-
1:03:5CgWULVnU5HUOWP1MNg95pkiUAwayiO+IvrALZR8HpeJIHw3xyHdGlTnv
EJ81HDjBb+fGteUgIn0fj8ASHZigkuFIx8=:
DHHC-
1:03:5CgWULVnU5HUOWP1MNg95pkiUAwayiO+IvrALZR8HpeJIHw3xyHdGlTnv
EJ81HDjBb+fGteUgIn0fj8ASHZigkuFIx8=:
DHHC-
1:03:5CgWULVnU5HUOWP1MNg95pkiUAwayiO+IvrALZR8HpeJIHw3xyHdGlTnv
EJ81HDjBb+fGteUgIn0fj8ASHZigkuFIx8=:
```



## 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

```
[
  {
    "hostnqn":"nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0054-
5110-8039-c3c04f523034",
    "hostid":"44454c4c-5400-1051-8039-c3c04f523034",
    "dhchap_key":"DHHC-
1:01:2G7lsg9PM000h1Wf1g4QtP0XT11kREz0qVuLm2xvZdbaWR/g:",
    "subsystems":[
      {
        "nqn":"nqn.1992-
08.com.netapp:sn.5857c8c9b22411f08d0ed039eac03c33:subsystem.Bidi
rectional_DHCP_1_0",
        "ports":[
          {
            "transport":"tcp",
            "traddr":"192.168.20.28",
            "host_traddr":"192.168.20.21",
            "trsvcid":"4420",
            "dhchap_ctrl_key":"DHHC-
1:03:5CgWULVnU5HUOwPlMN95pkiUAwayiO+IvrALZR8HpeJIHw3xyHdGlTnvEJ
81HDjBb+fGteUgIn0fj8ASHZlgkuFIx8="
          },
          {
            "transport":"tcp",
            "traddr":"192.168.20.29",
            "host_traddr":"192.168.20.21",
            "trsvcid":"4420",
            "dhchap_ctrl_key":"DHHC-
1:03:5CgWULVnU5HUOwPlMN95pkiUAwayiO+IvrALZR8HpeJIHw3xyHdGlTnvEJ
81HDjBb+fGteUgIn0fj8ASHZlgkuFIx8="
          },
          {
            "transport":"tcp",
            "traddr":"192.168.21.28",
            "host_traddr":"192.168.21.21",
            "trsvcid":"4420",
            "dhchap_ctrl_key":"DHHC-
1:03:5CgWULVnU5HUOwPlMN95pkiUAwayiO+IvrALZR8HpeJIHw3xyHdGlTnvEJ
81HDjBb+fGteUgIn0fj8ASHZlgkuFIx8="
          },
          {
            "transport":"tcp",
            "traddr":"192.168.21.29",
```

```

        "host_traddr": "192.168.21.21",
        "trsvcid": "4420",
        "dhchap_ctrl_key": "DHHC-
1:03:5CgWULVnU5HUOWPlMN95pkiUAWayio+IvrALZR8HpeJIHw3xyHdGlTnvEJ
81HDjBb+fGteUgIn0fj8ASHZigkuFIx8="
    }
}
}
]
}
]
]

```

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

```

traddr=192.168.20.28 is already connected
traddr=192.168.20.28 is already connected
traddr=192.168.20.29 is already connected
traddr=192.168.20.29 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-subsys4/nvme4/dhchap_secret
```

次の例は、dhchap キーを示しています。

```
DHHC-1:01:2G71sg9PM00h1Wf1g4QtP0XT11kREz0qVuLm2xvZdbaWR/g:
```

- b. コントローラのDHCHAPキーを確認します。

```
cat /sys/class/nvme-subsystem/nvme-
subsys4/nvme4/dhchap_ctrl_secret
```

次の例のような出力が表示されます。

```
DHHC-  
1:03:5CgWULVnU5HUOWPlMNq95pkiUAWayio+IvrALZR8HpeJIHw3xyHdGlTnvEJ81HD  
jBb+fGteUgIn0fj8ASHZigkuFIx8=:
```

## 手順9：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RHEL 9.xを構成する

Red Hat Enterprise Linux (RHEL) 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。

RHEL 9.x 用の 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、["RHEL ONTAPのサポートと機能"](#)。

RHEL 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 ファームウェア、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: RHEL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 RHEL 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 RHEL 9.x カーネルが実行されていることを確認します。

```
uname -r
```

RHEL カーネルバージョンの例:

```
5.14.0-611.5.1.el9_7.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-2.13-1.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-1.13-1.el9.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`バージョン:

```
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ストレージ システム上の対応するサブシステムの文字列:

```
::> vserver nvme subsystem host show -vserver vs_188
```

例を示します

```
Vserver Subsystem Priority Host NQN
-----
vs_188  Nvme1
          regular  nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
          Nvme10
          regular  nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
          Nvme11
          regular  nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
          Nvme12
          regular  nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
48 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 - ブロードコム/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 ファームウェアおよび受信トレイドライバ:

- a. ファームウェアのバージョンを表示します。

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

このコマンドはファームウェアのバージョンを返します。

```
14.4.393.53, sli-4:6:d  
14.4.393.53, 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。

```
0:14.4.0.9
```

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。

```
0x1000000109bf044b1  
0x1000000109bf044b2
```

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 lpfc0 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc0 WWPN x100000109b954518 WWNN x200000109b954518
DID x020700 ONLINE
NVME RPORT          WWPN x2022d039eaa7dfc8 WWNN x201fd039eaa7dfc8
DID x020b03 TARGET DISCSRVC ONLINE
NVME RPORT          WWPN x2023d039eaa7dfc8 WWNN x201fd039eaa7dfc8
DID x020103 TARGET DISCSRVC ONLINE
```

```
NVME Statistics
LS: Xmt 0000000548 Cmpl 0000000548 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 00000000000001a68 Issue 00000000000001a68 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 RPORT          WWPN x2027d039eaa7dfc8 WWNN x2025d039eaa7dfc8
DID x020b01 TARGET DISCSRVC ONLINE
```

```
NVME Statistics
LS: Xmt 00000005ab Cmpl 00000005ab Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 00000000000086ce1 Issue 00000000000086ce2 OutIO
0000000000000001
          abort 0000009c noxri 00000000 nondlp 00000002 qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 000000b8 Err 000000b8
```

```
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 x2027d039eaa7dfc8 WWNN x2025d039eaa7dfc8
DID x020b01 TARGET DISCSRVC ONLINE
NVME RPORT          WWPN x2011d039eaa7dfc8 WWNN x200fd039eaa7dfc8
DID x020b02 TARGET DISCSRVC ONLINE
NVME RPORT          WWPN x2002d039eaa7dfc8 WWNN x2000d039eaa7dfc8
DID x020b05 TARGET DISCSRVC ONLINE
```

```
NVME RPORT          WWPN x2026d039eaa7dfc8 WWNN x2025d039eaa7dfc8
DID x021301 TARGET DISCSRV ONLINE
NVME RPORT          WWPN x2010d039eaa7dfc8 WWNN x200fd039eaa7dfc8
DID x021302 TARGET DISCSRV ONLINE
NVME RPORT          WWPN x2001d039eaa7dfc8 WWNN x2000d039eaa7dfc8
DID x021305 TARGET DISCSRV ONLINE
```

#### NVME Statistics

```
LS: Xmt 000000c186 Cmpl 000000c186 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 00000000c348ca37 Issue 00000000c3344057 OutIO
ffffffffffffeb7620
        abort 0000815b noxri 000018b5 nondlp 00000116 qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 0000915b Err 000c6091
```

#### 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 x2028d039eaa7dfc8 WWNN x2025d039eaa7dfc8
DID x020101 TARGET DISCSRV ONLINE
NVME RPORT          WWPN x2012d039eaa7dfc8 WWNN x200fd039eaa7dfc8
DID x020102 TARGET DISCSRV ONLINE
NVME RPORT          WWPN x2003d039eaa7dfc8 WWNN x2000d039eaa7dfc8
DID x020105 TARGET DISCSRV ONLINE
NVME RPORT          WWPN x2029d039eaa7dfc8 WWNN x2025d039eaa7dfc8
DID x022901 TARGET DISCSRV ONLINE
NVME RPORT          WWPN x2013d039eaa7dfc8 WWNN x200fd039eaa7dfc8
DID x022902 TARGET DISCSRV ONLINE
NVME RPORT          WWPN x2004d039eaa7dfc8 WWNN x2000d039eaa7dfc8
DID x022905 TARGET DISCSRV ONLINE
```

#### NVME Statistics

```
LS: Xmt 000000c186 Cmpl 000000c186 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 00000000b5761af5 Issue 00000000b564b55e OutIO
ffffffffffffee9a69
        abort 000083d7 noxri 000016ea nondlp 00000195 qdepth
00000000 wqerr 00000002 err 00000000
FCP CMPL: xb 000094a4 Err 000c22e7
```

### NVMe/FC - マーベル/QLLogic

Marvell/QLLogic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:v9.15.06 DVR:v10.02.09.400-k  
QLE2872 FW:v9.15.06 DVR:v10.02.09.400-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 全体で検出ログ 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 192.168.30.15 -a 192.168.30.48

Discovery Log Number of Records 8, Generation counter 18
=====Discovery Log Entry 0=====
trtype: tcp
adrfam: ipv4
subtype: current discovery subsystem
treql: not specified
portid: 8
trsvcid: 8009
subnqn: nqn.1992-
08.com.netapp:sn.51a3c9846e0c11f08f5dd039eaa7dfc9:discovery
traddr: 192.168.31.49
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: 7
trsvcid: 8009
subnqn: nqn.1992-
08.com.netapp:sn.51a3c9846e0c11f08f5dd039eaa7dfc9:discovery
traddr: 192.168.31.48
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: 6
trsvcid: 8009
subnqn: nqn.1992-
08.com.netapp:sn.51a3c9846e0c11f08f5dd039eaa7dfc9:discovery
traddr: 192.168.30.49
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: 5
trsvcid: 8009
subnqn: nqn.1992-
08.com.netapp:sn.51a3c9846e0c11f08f5dd039eaa7dfc9:discovery
traddr: 192.168.30.48
eflags: explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 4=====
trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 8
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.51a3c9846e0c11f08f5dd039eaa7dfc9:subsystem.Nvme
38
traddr: 192.168.31.49
eflags: none
sectype: none
=====Discovery Log Entry 5=====
trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 7
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.51a3c9846e0c11f08f5dd039eaa7dfc9:subsystem.Nvme
38
traddr: 192.168.31.48
eflags: none
sectype: none
=====Discovery Log Entry 6=====
trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 6
trsvcid: 4420
subnqn: nqn.1992-

```

```
08.com.netapp:sn.51a3c9846e0c11f08f5dd039eaa7dfc9:subsystem.Nvme
38
traddr: 192.168.30.49
eflags: none
sectype: none
=====Discovery Log Entry 7=====
trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 5
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.51a3c9846e0c11f08f5dd039eaa7dfc9:subsystem.Nvme
38
traddr: 192.168.30.48
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 192.168.30.15 -a 192.168.30.48
nvme discover -t tcp -w 192.168.30.15 -a 192.168.30.49
nvme discover -t tcp -w 192.168.31.15 -a 192.168.31.48
nvme discover -t tcp -w 192.168.31.15 -a 192.168.31.49
```

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イニシエーター/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 192.168.30.15 -a 192.168.30.48
nvme connect-all -t tcp -w 192.168.30.15 -a
192.168.30.49
nvme connect-all -t tcp -w 192.168.31.15 -a
192.168.31.48
nvme connect-all -t tcp -w 192.168.31.15 -a
192.168.31.49
```

RHEL 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: オプションとして、**udev**ルールの**iopolicy**を変更します。

RHEL 9.6ではNVMe-oFのデフォルトの*iopolicy*を次のように設定します。 *round-robin*。 RHEL 9.6を使用しており、*iopolicy*を次のように変更したい場合 *queue-depth*、*udev* ルール ファイルを次のように変更します。

手順

1. ルート権限でテキスト エディターで *udev* ルール ファイルを開きます。

```
/usr/lib/udev/rules.d/71-nvmf-netapp.rules
```

次の出力が表示されます。

```
vi /usr/lib/udev/rules.d/71-nvmf-netapp.rules
```

2. 次の例のルールに示すように、NetApp ONTAPコントローラーの *iopolicy* を設定する行を見つけます。

```
ACTION=="add", SUBSYSTEM=="nvme-subsystem", ATTR{subsysstype}=="nvm",
ATTR{model}=="NetApp ONTAP Controller", ATTR{iopolicy}="round-robin"
```

3. ルールを修正して round-robin`なる `queue-depth:

```
ACTION=="add", SUBSYSTEM=="nvme-subsystem", ATTR{subsysstype}=="nvm",  
ATTR{model}=="NetApp ONTAP Controller", ATTR{iopolicy}="queue-depth"
```

4. udev ルールを再読み込みし、変更を適用します。

```
udevadm control --reload  
udevadm trigger --subsystem-match=nvme-subsystem
```

5. サブシステムの現在の iopolicy を確認します。 <subsystem>を置き換えます。例: nvme-subsys0。

```
cat /sys/class/nvme-subsystem/<subsystem>/iopolicy
```

次の出力が表示されます。

```
queue-depth.
```



新しい iopolicy は、一致する NetApp ONTAP コントローラ デバイスに自動的に適用されます。再起動する必要はありません。

## ステップ5: 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 1 MB 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

1. `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6: NVMeブートサービスを確認する

その `nvme-fc-boot-connections.service` そして `nvmmf-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、`nvme-fc-boot-connections.service` そして `nvmmf-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 `nvmmf-autoconnect.service` ます。

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

出力例を表示します。

```
nvmmf-autoconnect.service - Connect NVMe-oF subsystems automatically
during boot
      Loaded: loaded (/usr/lib/systemd/system/nvmmf-
autoconnect.service; enabled; preset: disabled)

Active: inactive (dead) since Wed 2025-10-29 00:42:03 EDT; 6h ago
Main PID: 8487 (code=exited, status=0/SUCCESS) CPU: 66ms

Oct 29 00:42:03 R650-14-188 systemd[1]: Starting Connect NVMe-oF
subsystems automatically during boot...
Oct 29 00:42:03 R650-14-188 systemd[1]: nvmmf-autoconnect.service:
Deactivated successfully.
Oct 29 00:42:03 R650-14-188 systemd[1]: Finished Connect NVMe-oF
subsystems automatically during boot.
```

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

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

出力例を表示します。

```
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 Wed 2025-10-29 00:41:51 EDT; 6h
ago
    Main PID: 4652 (code=exited, status=0/SUCCESS)
    CPU: 13ms

Oct 29 00:41:51 R650-14-188 systemd[1]: Starting Auto-connect to
subsystems on FC-NVME devices found during boot... Oct 29 00:41:51
R650-14-188 systemd[1]: nvmeofc-boot-connections.service: Deactivated
successfully. Oct 29 00:41:51 R650-14-188 systemd[1]: Finished
Auto-connect to subsystems on FC-NVME devices found during boot
```

## ステップ7: 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. それぞれのONTAP名前空間の適切な NVMe-oF 設定 (モデルがNetApp ONTAPコントローラに設定され、負荷分散 iopolicy がラウンドロビンに設定されているなど) がホストに正しく表示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-subsys*/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-subsys*/iopolicy
```

次の出力が表示されます。

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

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		Generic	SN
Model			
-----			
/dev/nvme100n1	/dev/ng100n1	81LJCJYaKOHhAAAAAAaf	NetApp ONTAP
Controller			
Namespace Usage	Format	FW	Rev
-----			
0x1	1.19 GB /	5.37 GB 4 KiB + 0 B	9.18.1

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/nvme100n1
```

例を示します

```
nvme-subsys4 - NQN=nqn.1992-  
08.com.netapp:sn.3623e199617311f09257d039eaa7dfc9:subsystem.Nvme  
31  
          hostnqn=nqn.2014-08.org.nvmexpress:uuid:  
4c4c4544-0056-5410-8048-b9c04f42563  
          \  
+- nvme199 fc   traddr=nn-0x200fd039eaa7dfc8:pn-  
0x2010d039eaa7dfc8,host_traddr=nn-0x200000109bf044b1:pn-  
0x100000109bf044b1 live optimized  
+- nvme246 fc   traddr=nn-0x200fd039eaa7dfc8:pn-  
0x2011d039eaa7dfc8,host_traddr=nn-0x200000109bf044b1:pn-  
0x100000109bf044b1 live non-optimized  
+- nvme249 fc   traddr=nn-0x200fd039eaa7dfc8:pn-  
0x2013d039eaa7dfc8,host_traddr=nn-0x200000109bf044b2:pn-  
0x100000109bf044b2 live optimized  
+- nvme251 fc   traddr=nn-0x200fd039eaa7dfc8:pn-  
0x2012d039eaa7dfc8,host_traddr=nn-0x200000109bf044b2:pn-  
0x100000109bf044b2 live non-optimized
```

## NVMe/FC

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

例を示します

```
nvme-subsys0 - NQN=nqn.1992-08.com.netapp:sn.51a3c9846e0c11f08f5dd039eaa7dfc9:subsystem.Nvme1
hostnqn=nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0035-5910-804b-b5c04f444d33
\
+- nvme0 tcp
traddr=192.168.30.48,trsvcid=4420,host_traddr=192.168.30.15,
src_addr=192.168.30.15 live optimized
+- nvme1 tcp
traddr=192.168.30.49,trsvcid=4420,host_traddr=192.168.30.15,
src_addr=192.168.30.15 live non-optimized
+- nvme2 tcp
traddr=192.168.31.48,trsvcid=4420,host_traddr=192.168.31.15,
src_addr=192.168.31.15 live optimized
+- nvme3 tcp
traddr=192.168.31.49,trsvcid=4420,host_traddr=192.168.31.15,
src_addr=192.168.31.15 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 NSID	Vserver	Subsystem	Namespace Path
----- ----	-----	-----	-----
/dev/nvme0n1 1	vs_iscsi_tcp	Nvme1	/vol/Nvmevol1/ns1
UUID			Size
-----			-----
d8efef7d-4dde-447f-b50e-b2c009298c66			26.84GB

## JSON

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

### 例を示します

```
{
  "ONTAPdevices":[
    {
      "Device":"/dev/nvme0n1",
      "Vserver":"vs_iscsi_tcp",
      "Subsystem":"Nvme1",
      "Namespace_Path":"/vol/Nvmevol1/ns1",
      "NSID":1,
      "UUID":"d8efef7d-4dde-447f-b50e-b2c009298c66",
      "LBA_Size":4096,
      "Namespace_Size":26843545600,
    },
  ]
}
```

## ステップ8: 安全なインバンド認証を設定する

RHEL 9.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. RHEL 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/4m852vH7KGlrJeMpzhmyjD
W0o0PJJM6yZsTeEpGkDHMHQ255+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-subsysX>/nvme*/dhchap_secret
```

に、単方向設定の出力例を示します。

```
cat /sys/class/nvme-subsystem/nvme-subsys1/nvme*/dhchap_secret  
DHHC-1:01:hhdIYK7rGxHiNYS4d421GxHeDRUAuY0vmdqCp/NOaYND2PSc:  
DHHC-1:01:hhdIYK7rGxHiNYS4d421GxHeDRUAuY0vmdqCp/NOaYND2PSc:  
DHHC-1:01:hhdIYK7rGxHiNYS4d421GxHeDRUAuY0vmdqCp/NOaYND2PSc:  
DHHC-1:01:hhdIYK7rGxHiNYS4d421GxHeDRUAuY0vmdqCp/NOaYND2PSc:
```

- b. コントローラのDHCHAPキーを確認します。

```
cat /sys/class/nvme-subsystem/<nvme-  
subsysX>/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-  
subsys*/nvme*/dhchap_ctrl_secret  
  
DHHC-  
1:03:ZCRrP9MQOeXhFitT7Fvvf/3P6K/qY1HfSmSfM8nLjESJdOjbjK/J6m00y  
gJgjm0VrRlgrnHzjtWJmsnoVBO3rPDGEk=:  
DHHC-  
1:03:ZCRrP9MQOeXhFitT7Fvvf/3P6K/qY1HfSmSfM8nLjESJdOjbjK/J6m00y  
gJgjm0VrRlgrnHzjtWJmsnoVBO3rPDGEk=:  
DHHC-  
1:03:ZCRrP9MQOeXhFitT7Fvvf/3P6K/qY1HfSmSfM8nLjESJdOjbjK/J6m00y  
gJgjm0VrRlgrnHzjtWJmsnoVBO3rPDGEk=:  
DHHC-  
1:03:ZCRrP9MQOeXhFitT7Fvvf/3P6K/qY1HfSmSfM8nLjESJdOjbjK/J6m00y  
gJgjm0VrRlgrnHzjtWJmsnoVBO3rPDGEk=:
```

## 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:4c4c4544-0035-5910-
804b-b5c04f444d33",
  "hostid":"4c4c4544-0035-5910-804b-b5c04f444d33",
  "dhchap_key":"DHHC-
1:01:GhgaLS+0h0W/IxKhSa0iaMHg17SOHRTzBduPzoJ6LKEJs3/f:",
  "subsystems":[
    {
      "nqn":"nqn.1992-
08.com.netapp:sn.2c0c80d9873a11f0bc60d039eab6cb6d:subsystem.istp
MNTC_subsys",
      "ports":[
        {
          "transport":"tcp",
          "traddr":"192.168.30.44",
          "host_traddr":"192.168.30.15",
          "trsvcid":"4420",
          "dhchap_ctrl_key":"DHHC-
1:03:GaraCO84o/uM0jF4rKJlgTy22bVoV0dRn1M+9QDfQRNVwJDHfPu2LrK5Y+/
XG8iGcRtBCdm3
fYm3ZmO6NiepCORoY5Q=:"
        },
        {
          "transport":"tcp",
          "traddr":"192.168.30.45"
          "host_traddr":"192.168.30.15",
          "trsvcid":"4420",
          "dhchap_ctrl_key":"DHHC-
1:03:GaraCO84o/uM0jF4rKJlgTy22bVoV0dRn1M+9QDfQRNVwJDHfPu2LrK5Y+/
XG8iGcRtBCdm3
fYm3ZmO6NiepCORoY5Q=:"
        },
        {
          "transport":"tcp",
          "traddr":"192.168.31.44",
          "host_traddr":"192.168.31.15",
          "trsvcid":"4420",
          "dhchap_ctrl_key":"DHHC-
1:03:
GaraCO84o/uM0jF4rKJlgTy22bVoV0dRn1M+9QDfQRNVwJDHfPu2LrK5Y+/XG8iG
c
```

```

RtBCdm3fYm3ZmO6NiepCORoY5Q=: "
    {
        "transport": "tcp",
        "traddr": "192.168.31.45",
        "host_traddr": "192.168.31.15",
        "trsvcid": "4420",
        "dhchap_ctrl_key": "DHHC-
1:03:
GaraCO84o/uM0jF4rKJlgTy22bVoV0dRn1M+9QDfQRNVwJDHfPu2LrK5Y+/XG8iG
cRtBCdm3fYm3ZmO6NiepCORoY5Q=: "
    }
]
}
]

```

## 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.nvmexpress:uuid:4c4c4544-0035-5910-804b-
b5c04f444d33,nqn=nqn.1992-
08.com.netapp:sn.2c0c80d9873a11f0bc60d039eab6cb6d:subsystem.istp
MNTC_subsys,transport=tcp,traddr=192.168.30.44,trsvcid=4420
already connected to hostnqn=nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0035-5910-804b-
b5c04f444d33,nqn=nqn.1992-
08.com.netapp:sn.2c0c80d9873a11f0bc60d039eab6cb6d:subsystem.istp
MNTC_subsys,transport=tcp,traddr=192.168.31.44,trsvcid=4420
already connected to hostnqn=nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0035-5910-804b-
b5c04f444d33,nqn=nqn.1992-
08.com.netapp:sn.2c0c80d9873a11f0bc60d039eab6cb6d:subsystem.istp
MNTC_subsys,transport=tcp,traddr=192.168.30.45,trsvcid=4420
already connected to hostnqn=nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0035-5910-804b-
b5c04f444d33,nqn=nqn.1992-
08.com.netapp:sn.2c0c80d9873a11f0bc60d039eab6cb6d:subsystem.istp
MNTC_subsys,transport=tcp,traddr=192.168.31.45,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-subsys96/nvme96/dhchap_secret
```

次の例は、dhchap キーを示しています。

```
DHHC-1:01:hhdIYK7rGxHiNYS4d421GxHeDRUAuY0vmdqCp/NOaYND2PSc:
```

b. コントローラのDHCHAPキーを確認します。

```
cat /sys/class/nvme-subsystem/nvme-subsys96/nvme96/dhchap_ctrl_secret
```

次の例のような出力が表示されます。

```
DHHC-
1:03:ZCRrP9MQOeXhFitT7Fvvf/3P6K/qY1HfSmSfM8nLjESJdOjbjK/J6m00ygJgjm0
VrRlrgnrnHzjtWJmsnoVBO3rPDGEk=:
```

## 手順9：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
1503468	RHEL 9.1では、`nvme list-subsys`コマンドは、指定されたサブシステムの繰り返しNVMEコントローラリストを返します。	その`nvme list-subsys`コマンドは、指定されたサブシステムの NVMe コントローラーのリストを返します。 RHEL 9.1 では、このコマンドは、サブシステム内のすべての名前空間のコントローラーとその ANA 状態を表示します。 ANA の状態は名前空間ごとの属性であるため、コマンドは各名前空間のパスの状態を持つ一意のコントローラー エントリを表示する必要があります。

NetApp バグ ID	タイトル	説明
"1479047"	RHEL 9.0 NVMe-oF ホストは重複した永続検出コントローラ (PDC) を作成します。	NVMe-oF ホストでは、 <code>nvme discover -p</code> コマンドを使用して PDC を作成できます。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、NVMe-oF ホストで ONTAP 9.10.1 および RHEL 9.0 を実行している場合は、 <code>`nvme discover -p`</code> 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。

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

Red Hat Enterprise Linux (RHEL) 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。

RHEL 8.x 用の 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、["RHEL ONTAP のサポートと機能"](#)。

RHEL 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。
- RHEL 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。

**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 ファームウェア、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: RHEL 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

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 RHEL 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 RHEL 8.x カーネルが実行されていることを確認します。

```
uname -r
```

RHEL カーネルバージョンの例:

```
4.18.0-553.el8_10.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-1.16-9.el8.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 パッケージのバージョンを示しています。

```
libnvme-1.4-3.el8.x86_64
```

4. カーネル内の NVMe マルチパスを有効にします。

```
grubby --args=nvme_core.multipath=Y --update-kernel /boot/vmlinuz-  
4.18.0-553.el8_10.x86_64
```

5. RHEL 8.xホストでは、hostnqn`文字列` /etc/nvme/hostnqn:

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

次の例は、`hostnqn`バージョン:

```
nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0032-3410-8035-b8c04f4c5132
```

6. ONTAPシステムで、`hostnqn`文字列が一致する `hostnqn`ONTAPストレージ システム上の対応するサブシステムの文字列:

```
::> vserver nvme subsystem host show -vserver vs_fcnvme_141
```

例を示します

Vserver	Subsystem	Host NQN
vs_25_2742	rhel_101_QLe2772	nqn.2014-08.org.nvmexpress:uuid:546399fc-160f-11e5-89aa-98be942440ca



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。

7. ホストをリポートします。

同じホスト上で NVMe と SCSI トラフィックの両方を実行するには、NetApp、ONTAP名前空間にはカーネル内 NVMe マルチパスを使用し、ONTAP LUN には dm-multipath を使用することを推奨しています。dm-multipathがONTAPネームスペースデバイスを要求しないようにするには、`enable\_foreign`設定する `/etc/multipath.conf` ファイル:



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

### ステップ3: NVMe/FCとNVMe/TCPを構成する

Broadcom/Emulex または Marvell/QLLogic 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 - ブロードコム/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。

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

- 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 LPe32002-M2 2-Port 32Gb Fibre Channel Adapter  
Emulex LightPulse LPe32002-M2 2-Port 32Gb 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 ファームウェアおよび受信トレイドライバ:

- a. ファームウェアのバージョンを表示します。

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

このコマンドはファームウェアのバージョンを返します。

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

- 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。

```
0:14.0.0.21
```

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：

```
0x10000090fae0ec88  
0x10000090fae0ec89
```

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 lpfc0 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc0 WWPN x100000109bf044b1 WWNN x200000109bf044b1
DID x022a00 ONLINE
NVME RPORT          WWPN x211ad039eaa7dfc8 WWNN x2119d039eaa7dfc8
DID x021302 TARGET DISCSRVC ONLINE
NVME RPORT          WWPN x211cd039eaa7dfc8 WWNN x2119d039eaa7dfc8
DID x020b02 TARGET DISCSRVC ONLINE
```

```
NVME Statistics
LS: Xmt 00000001ff Cmpl 00000001ff Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 0000000001330ec7 Issue 0000000001330ec9 OutIO
00000000000000002
          abort 00000330 noxri 00000000 nondlp 0000000b qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 00000354 Err 00000361
```

```
NVME Initiator Enabled
XRI Dist lpfc1 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc1 WWPN x100000109bf044b2 WWNN x200000109bf044b2
DID x021b00 ONLINE
NVME RPORT          WWPN x211bd039eaa7dfc8 WWNN x2119d039eaa7dfc8
DID x022902 TARGET DISCSRVC ONLINE
NVME RPORT          WWPN x211dd039eaa7dfc8 WWNN x2119d039eaa7dfc8
DID x020102 TARGET DISCSRVC ONLINE
```

```
NVME Statistics
LS: Xmt 00000001ff Cmpl 00000001ff Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 00000000012ec220 Issue 00000000012ec222 OutIO
00000000000000002
          abort 0000033b noxri 00000000 nondlp 00000085 qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 00000368 Err 00000382
```

## 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。

```
QLE2742 FW: v9.10.11 DVR: v10.02.08.200-k  
QLE2742 FW: v9.10.11 DVR: v10.02.08.200-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イニシエーター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 全体で検出ログ 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 192.168.2.31 -a 192.168.2.25

Discovery Log Number of Records 8, Generation counter 18
=====Discovery Log Entry 0=====
trtype: tcp
adrfam: ipv4
subtype: unrecognized
treql: not specified.
portid: 0
trsvcid: 8009
subnqn: nqn.1992-
08.com.netapp:sn.alb2b785b9de11ee8e7fd039ea9e8ae9:discovery
traddr: 192.168.1.25
sectype: none
=====Discovery Log Entry 1=====
trtype: tcp
adrfam: ipv4
subtype: unrecognized
treql: not specified.
portid: 1
trsvcid: 8009
subnqn: nqn.1992-
08.com.netapp:sn.alb2b785b9de11ee8e7fd039ea9e8ae9:discovery
traddr: 192.168.2.26
sectype: none
```

2. 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 192.168.2.31 -a 192.168.2.25
nvme discover -t tcp -w 192.168.1.31 -a 192.168.1.24
nvme discover -t tcp -w 192.168.2.31 -a 192.168.2.26
nvme discover -t tcp -w 192.168.1.31 -a 192.168.1.25
```

3. 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 192.168.2.31 -a 192.168.2.25 -l 1800
nvme connect-all -t tcp -w 192.168.1.31 -a 192.168.1.24 -l 1800
nvme connect-all -t tcp -w 192.168.2.31 -a 192.168.2.26 -l 1800
nvme connect-all -t tcp -w 192.168.1.31 -a 192.168.1.25 -l 1800
```

## 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 1 MB 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

1. `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: 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-subsys*/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-subsys*/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	81Ix2BVuekWcAAAAAAB	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 list-subsys /dev/nvme0n1
```

NVMe/FCの例を表示

```
nvme-subsys0 - NQN=nqn.1992-08.com.netapp:sn.0cd9ee0dc0ec11ee8e7fd039ea9e8ae9:subsystem.nvme
\
+- nvme1 fc traddr=nn-0x2005d039eaa7dfc8:pn-0x2086d039eaa7dfc8
host_traddr=nn-0x20000024ff752e6d:pn-0x21000024ff752e6d live non-optimized
+- nvme2 fc traddr=nn-0x2005d039eaa7dfc8:pn-0x2016d039eaa7dfc8
host_traddr=nn-0x20000024ff752e6c:pn-0x21000024ff752e6c live optimized
+- nvme3 fc traddr=nn-0x2005d039eaa7dfc8:pn-0x2081d039eaa7dfc8
host_traddr=nn-0x20000024ff752e6c:pn-0x21000024ff752e6c live non-optimized
+- nvme4 fc traddr=nn-0x2005d039eaa7dfc8:pn-0x2087d039eaa7dfc8
host_traddr=nn-0x20000024ff752e6d:pn-0x21000024ff752e6d live optimized
```



## NVMe/TCPの例を示す

```
nvme-subsys0 - NQN=nqn.1992-  
08.com.netapp:sn.a1b2b785b9de11ee8e7fd039ea9e8ae9:subsystem.nvme_tcp  
_1  
\  
+- nvme0 tcp traddr=192.168.2.26 trsvcid=4420  
host_traddr=192.168.2.31 live non-optimized  
+- nvme1 tcp traddr=192.168.2.25 trsvcid=4420  
host_traddr=192.168.2.31 live optimized  
+- nvme2 tcp traddr=192.168.1.25 trsvcid=4420  
host_traddr=192.168.1.31 live non-optimized  
+- nvme3 tcp traddr=192.168.1.24 trsvcid=4420  
host_traddr=192.168.1.31 live optimized
```

5. 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	81Ix2BVuekWcAAAAAAB	NetApp ONTAP Controller		
Namespace	Usage	Format	FW	Rev
-----				
1		21.47 GB / 21.47 GB	4 KiB + 0 B	FFFFFFFF

6. 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/nvme0n1	tcpiscsi_129	/vol/tcpnvme_1_0_0/tcpnvme_ns

  

NSID	UUID	Size
1	05c2c351-5d7f-41d7-9bd8-1a56c	21.47GB

## JSON

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

### 例を示します

```
{
  "ONTAPdevices": [
    {
      "Device": "/dev/nvme0n1",
      "Vserver": "tcpiscsi_129",
      "Namespace Path": "/vol/tcpnvme_1_0_0/tcpnvme_ns ",
      "NSID": 1,
      "UUID": " 05c2c351-5d7f-41d7-9bd8-1a56c160c80b ",
      "Size2": "21.47GB",
      "LBA_Data_Size": 4096,
      "Namespace Size" : 5242880
    },
  ]
}
```

## 手順6：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
<a href="#">"1479047"</a>	RHEL 8.x NVMe-oF ホストは重複した永続検出コントローラ (PDC) 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 PDC を作成できます。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、NVMe-oF ホストで RHEL 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随時、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 5252.227-19（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权を有し、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权については、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。