



SUSE Linux Enterprise Server 15

ONTAP SAN Host Utilities

NetApp
December 19, 2025

目次

SUSE Linux Enterprise Server 15	1
SUSE Linux Enterprise Server 15 SP7 with ONTAP向けのNVMe-oFホスト構成	1
手順1：必要に応じてSANブートを有効にします。	1
ステップ2: ソフトウェアのバージョンを検証する	2
ステップ3: NVMe/FCを構成する	4
ステップ4: オプションで1MBのI/Oを有効にする	8
ステップ5: NVMeブートサービスを確認する	9
ステップ6: NVMe/TCPを構成する	11
ステップ7: NVMe-oFを検証する	15
ステップ8: 永続的な検出コントローラを作成する	19
ステップ9: 安全なインバンド認証を設定する	24
ステップ10: トランスポート層セキュリティを構成する	30
手順11：既知の問題を確認する	35
SUSE Linux Enterprise Server 15 SP6 with ONTAP向けのNVMe-oFホスト構成	35
の機能	36
既知の制限	36
NVMe/FC を設定	36
NVMe/FC を設定	42
NVMe-oF を検証します	46
永続的検出コントローラの作成	50
セキュアなインバンド認証のセットアップ	55
Transport Layer Securityの設定	61
既知の問題	66
SUSE Linux Enterprise Server 15 SP5 with ONTAP向けのNVMe-oFホスト構成	67
の機能	67
既知の制限	67
NVMe/FC を設定	67
NVMe/FC を設定	73
NVMe-oF を検証します	76
永続的検出コントローラの作成	80
セキュアなインバンド認証のセットアップ	87
既知の問題	93
SUSE Linux Enterprise Server 15 SP4 with ONTAP向けのNVMe-oFホスト構成	93
の機能	93
既知の制限	93
NVMe/FC を設定	93
NVMe/FC を設定	99
NVMe-oF を検証します	102
永続的検出コントローラの作成	105

セキュアなインバンド認証のセットアップ	112
既知の問題	118
ONTAP を搭載した SUSE Linux Enterprise Server 15 SP3 用 NVMe-oF ホスト構成	118
の機能	118
既知の制限	118
カーネル内の NVMe マルチパスを有効にします	118
NVMe-oF イニシエータパッケージ	118
NVMe/FC を設定	120
NVMe/FC を設定	123
NVMe-oF を検証します	125
既知の問題	129
ONTAP を搭載した SUSE Linux Enterprise Server 15 SP2 用 NVMe/FC ホスト構成	129
既知の制限	130
SUSE Linux Enterprise Server 15 SP2でNVMe/FCを有効にする	130
Broadcom FC アダプタを NVMe/FC 用に設定します	131
NVMe/FC を検証	132
既知の問題	134
Broadcom NVMe/FC の 1MB I/O サイズを有効にします	134
lpfc 詳細ログ	135
SUSE Linux Enterprise Server 15 SP1 と ONTAP の NVMe/FC ホスト構成	135
既知の制限	136
SUSE Linux Enterprise Server 15 SP1でのNVMe/FCの有効化	136
Broadcom FC アダプタを NVMe/FC 用に設定します	136
NVMe/FC を検証	138
既知の問題	139
Broadcom NVMe/FC の 1MB I/O サイズを有効にします	140
lpfc 詳細ログ	140

SUSE Linux Enterprise Server 15

SUSE Linux Enterprise Server 15 SP7 with ONTAP向けのNVMe-oFホスト構成

SUSE Linux Enterprise Server 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ています。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。

このタスクについて

SUSE Linux Enterprise Server 15 SP7 の NVMe-oF ホスト構成では、次のサポートと機能を使用できます。構成プロセスを開始する前に、既知の制限事項も確認する必要があります。

- 利用可能なサポート：
 - NVMe over Fibre Channel (NVMe/FC) に加えて、NVMe over TCP (NVMe/TCP) もサポートされます。標準パッケージのNetAppプラグイン `nvme-cli` には、NVMe/FCとNVMe/TCP両方のネームスペースのONTAPの詳細が表示されます。
 - 同じホストでNVMeトライフィックとSCSIトライフィックの両方を実行している。たとえば、SCSI LUN のSCSI mpathデバイスにdm-multipathを設定し、NVMeマルチパスを使用してホストにNVMe-oFネームスペースデバイスを設定できます。
 - NVMe/FCプロトコルを使用したSANブートがサポートされます。
 - ONTAP 9.12.1 以降では、NVMe/TCP および NVMe/FC に対する安全なインバンド認証のサポートが導入されています。SUSE Linux Enterprise Server 15 SP7 では、NVMe/TCP および NVMe/FC に安全なインバンド認証を使用できます。
 - 一意の検出 NQN を使用した永続検出コントローラ (PDC) のサポート。
 - NVMe/TCP の TLS 1.3 暗号化のサポート。
 - NetApp `sanlun` SUSE Linux Enterprise Server15 SP7 ホスト上の NVMe-oF ではホスト ユーティリティ サポートは利用できません。代わりに、ネイティブに含まれるNetAppプラグインを利用できます。`nvme-cli` すべての NVMe-oF トランスポート用のパッケージ。

サポートされる構成の詳細については、を参照してください "[Interoperability Matrix Tool](#)"。

- 利用可能な機能：
 - 利用できる新しい機能はありません。
- 既知の制限
 - 発行を避ける nvme disconnect-all 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るため、使用しないでください。

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設定することで、導入を簡易化し、拡張性を向上させることができます。

開始する前に

を使用 "[Interoperability Matrix Tool](#)" して、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. SAN ブート名前空間を作成し、それをホストにマップします。

を参照してください "[NVMeストレージのプロビジョニング](#)"

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. 構成が正常に完了したことを確認するために、ホストをリブートし、OSが稼働していることを確認します。

ステップ2: ソフトウェアのバージョンを検証する

サポートされている最小の SUSE Linux Enterprise Server 15 SP7 ソフトウェア バージョンを検証するには、次の手順に従います。

手順

1. サーバーに SUSE Linux Enterprise Server15 SP7 をインストールします。インストールが完了したら、指定された SUSE Linux Enterprise Server 15 SP7 カーネルが実行されていることを確認します。

```
uname -r
```

次の例は、SUSE Linux Enterprise Server カーネルのバージョンを示しています。

```
6.4.0-150700.53.3-default
```

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+22.gd31b1a01-150700.3.3.2.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:

```
libnvme1-1.11+4.ge68a91ae-150700.4.3.2.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.nvmeexpress:uuid:f6517cae-3133-11e8-bbff-7ed30aef123f
```

5. を確認します hostnqn 文字列はに一致します 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.nvmeexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
    nvme_1
        regular    nqn.2014-
08.org.nvmeexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
    nvme_2
        regular    nqn.2014-
08.org.nvmeexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
    nvme_3
        regular    nqn.2014-
08.org.nvmeexpress: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を構成する

NVMe/FCは、Broadcom/Emulex FCアダプタまたはMarvell/Qlogic FCアダプタで構成できます。NVMe/TCPサブシステムと名前空間を手動で検出する必要があります。

Broadcom / Emulex

Broadcom/Emulex FCアダプタ用にNVMe/FCを設定

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
```

次の例はファームウェアのバージョンを示しています。

```
14.4.393.25, sli-4:2:c  
14.4.393.25, 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/lpfcc/version
```

次の例は、ドライバーのバージョンを示しています。

```
0:14.4.0.8
```

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。

```
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
```

出力例を表示します。

```
NVME Initiator Enabled
XRI Dist lpfco Total 6144 IO 5894 ELS 250
NVME LPORT lpfco WWPN x10000090fae0ec88 WWNN x20000090fae0ec88
DID x0a1300 ONLINE
NVME RPORT      WWPN x23b1d039ea359e4a WWNN x23aed039ea359e4a
DID x0a1c01 TARGET DISCSRVC ONLINE
NVME RPORT      WWPN x22bbd039ea359e4a WWNN x22b8d039ea359e4a
DID x0a1c0b TARGET DISCSRVC ONLINE
NVME RPORT      WWPN x2362d039ea359e4a WWNN x234ed039ea359e4a
DID x0a1c10 TARGET DISCSRVC ONLINE
NVME RPORT      WWPN x23afd039ea359e4a WWNN x23aed039ea359e4a
DID x0a1a02 TARGET DISCSRVC ONLINE
NVME RPORT      WWPN x22b9d039ea359e4a WWNN x22b8d039ea359e4a
DID x0a1a0b TARGET DISCSRVC ONLINE
NVME RPORT      WWPN x2360d039ea359e4a WWNN x234ed039ea359e4a
DID x0a1a11 TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 0000004ea0 Cmpl 0000004ea0 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 0000000000102c35 Issue 0000000000102c2d OutIO
fffffffffffffff8
          abort 00000175 noxri 00000000 nondlp 0000021d qdepth
00000000 wqerr 00000007 err 00000000
FCP CMPL: xb 00000175 Err 0000058b

NVME Initiator Enabled
XRI Dist lpfcl Total 6144 IO 5894 ELS 250
NVME LPORT lpfcl WWPN x10000090fae0ec89 WWNN x20000090fae0ec89
DID x0a1200 ONLINE
NVME RPORT      WWPN x23b2d039ea359e4a WWNN x23aed039ea359e4a
DID x0a1d01 TARGET DISCSRVC ONLINE
NVME RPORT      WWPN x22bcd039ea359e4a WWNN x22b8d039ea359e4a
DID x0a1d0b TARGET DISCSRVC ONLINE
NVME RPORT      WWPN x2363d039ea359e4a WWNN x234ed039ea359e4a
DID x0a1d10 TARGET DISCSRVC ONLINE
NVME RPORT      WWPN x23b0d039ea359e4a WWNN x23aed039ea359e4a
DID x0a1b02 TARGET DISCSRVC ONLINE
NVME RPORT      WWPN x22bad039ea359e4a WWNN x22b8d039ea359e4a
DID x0a1b0b TARGET DISCSRVC ONLINE
NVME RPORT      WWPN x2361d039ea359e4a WWNN x234ed039ea359e4a
DID x0a1b11 TARGET DISCSRVC ONLINE

NVME Statistics
```

```
LS: Xmt 0000004e31 Cmpl 0000004e31 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 00000000001017f2 Issue 00000000001017ef OutIO
fffffffffffffd
    abort 0000018a noxri 00000000 nondlp 0000012e qdepth
00000000 wqerr 00000004 err 00000000
FCP CMPL: xb 0000018a Err 000005ca
```

Marvell/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:v10.02.09.400-k-debug
QLE2742 FW:v9.14.00 DVR:v10.02.09.400-k-debug
```

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。

ステップ4: オプションで1MBの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

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: NVMeブートサービスを確認する

SUSE Linux Enterprise Server 15 SP7では、`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起動するように有効化されます。システムの起動が完了したら、ブート サービスが有効になっていることを確認します。

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: enabled)
     Active: inactive (dead) since Fri 2025-07-04 23:56:38 IST; 4 days ago
       Main PID: 12208 (code=exited, status=0/SUCCESS)
         CPU: 62ms

Jul 04 23:56:26 localhost systemd[1]: Starting Connect NVMe-oF subsystems automatically during boot...
Jul 04 23:56:38 localhost systemd[1]: nvmf-autoconnect.service: Deactivated successfully.
Jul 04 23:56:38 localhost systemd[1]: Finished Connect NVMe-oF subsystems automatically during boot.
```

2. 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 Mon 2025-07-07 19:52:30 IST; 1 day 4h ago
       Main PID: 2945 (code=exited, status=0/SUCCESS)
         CPU: 14ms

Jul 07 19:52:30 HP-DL360-14-168 systemd[1]: Starting Auto-connect to subsystems on FC-NVME devices found during boot...
Jul 07 19:52:30 HP-DL360-14-168 systemd[1]: nvmefc-boot-connections.service: Deactivated successfully.
Jul 07 19:52:30 HP-DL360-14-168 systemd[1]: Finished Auto-connect to subsystems on FC-NVME devices found during boot.
```

ステップ6: NVMe/TCPを構成する

NVMe/TCPプロトコルでの処理がサポートされてい `auto-connect` ません。代わりに、NVMe/TCPまたは `connect-all` の処理を手動で実行することで、NVMe/TCPサブシステムとネームスペースを検出できます `connect`。

手順

1. イニシエータポートがサポートされているNVMe/TCP LIFの検出ログページの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>
```

出力例を表示します。

```
nvme discover -t tcp -w 192.168.111.80 -a 192.168.111.70
Discovery Log Number of Records 8, Generation counter 42
=====Discovery Log Entry 0=====
trtype: tcp
adrifam: ipv4
subtype: current discovery subsystem
treq: not specified
portid: 4
trsvcid: 8009
subnqn: nqn.1992-
08.com.netapp:sn.f8e2af201b7211f0ac2bd039eab67a95:discovery
traddr: 192.168.211.71
eflags: explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 1=====
trtype: tcp
adrifam: ipv4
subtype: current discovery subsystem
treq: not specified
portid: 3
trsvcid: 8009
subnqn: nqn.1992-
08.com.netapp:sn.f8e2af201b7211f0ac2bd039eab67a95:discovery
traddr: 192.168.111.71
eflags: explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 2=====
trtype: tcp
adrifam: ipv4
subtype: current discovery subsystem
treq: not specified
portid: 2
trsvcid: 8009
subnqn: nqn.1992-
08.com.netapp:sn.f8e2af201b7211f0ac2bd039eab67a95:discovery
traddr: 192.168.211.70
eflags: explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 3=====
trtype: tcp
```

```
adrfam: ipv4
subtype: current discovery subsystem
treq: not specified
portid: 1
trsvcid: 8009
subnqn: nqn.1992-
08.com.netapp:sn.f8e2af201b7211f0ac2bd039eab67a95:discovery
traddr: 192.168.111.70
eflags: explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 4=====
trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 4
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.f8e2af201b7211f0ac2bd039eab67a95:subsystem.sample_t
cp_sub
traddr: 192.168.211.71
eflags: none
sectype: none
=====Discovery Log Entry 5=====
trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 3
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.f8e2af201b7211f0ac2bd039eab67a95:subsystem.sample_t
cp_sub
traddr: 192.168.111.71
eflags: none
sectype: none
=====Discovery Log Entry 6=====
trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 2
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.f8e2af201b7211f0ac2bd039eab67a95:subsystem.sample_t
```

```
cp_sub
traddr: 192.168.211.70
eflags: none
sectype: none
=====Discovery Log Entry 7=====
trtype: tcp
adrfrm: ipv4
subtype: nvme subsystem
treq: not specified
portid: 1
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.f8e2af201b7211f0ac2bd039eab67a95:subsystem.sample_t
cp_sub
traddr: 192.168.111.70
eflags: none
sectype: none
localhost:~ #
```

2. NVMe/TCPイニシエータとターゲットLIFの他のすべての組み合わせで、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 192.168.111.80 -a 192.168.111.66
nvme discover -t tcp -w 192.168.111.80 -a 192.168.111.67
nvme discover -t tcp -w 192.168.211.80 -a 192.168.211.66
nvme discover -t tcp -w 192.168.211.80 -a 192.168.211.67
```

3. 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 192.168.111.80 -a 192.168.111.66
nvme connect-all -t tcp -w 192.168.111.80 -a 192.168.111.67
nvme connect-all -t tcp -w 192.168.211.80 -a 192.168.211.66
nvme connect-all -t tcp -w 192.168.211.80 -a 192.168.211.67
```

SUSE Linux Enterprise Server 15 SP6以降、NVMe/TCPのデフォルト設定は `ctrl-loss-tmo`、タイムアウトはオフになります。これは再試行回数に制限がない（無期限再試行）ことを意味し、特定の再試行回数を手動で設定する必要はありません。`ctrl-loss-tmo` 使用時のタイムアウト時間 `nvme connect` または `nvme connect-all` コマンド（オプション `--1`）。さらに、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7: NVMe-oFを検証する

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
```

出力例を表示します。

```
nvme-subsys114 - NQN=nqn.1992-
08.com.netapp:sn.9e30b9760a4911f08c87d039eab67a95:subsystem.sles
_161_27
    hostnqn=nqn.2014-
08.org.nvmeexpress:uuid:f6517cae-3133-11e8-bbff-7ed30aef123f
iopolicy=round-robin\
+- nvme114 fc traddr=nn-0x234ed039ea359e4a:pn-
0x2360d039ea359e4a,host_traddr=nn-0x20000090fae0ec88:pn-
0x10000090fae0ec88 live optimized
+- nvme115 fc traddr=nn-0x234ed039ea359e4a:pn-
0x2362d039ea359e4a,host_traddr=nn-0x20000090fae0ec88:pn-
0x10000090fae0ec88 live non-optimized
+- nvme116 fc traddr=nn-0x234ed039ea359e4a:pn-
0x2361d039ea359e4a,host_traddr=nn-0x20000090fae0ec89:pn-
0x10000090fae0ec89 live optimized
+- nvme117 fc traddr=nn-0x234ed039ea359e4a:pn-
0x2363d039ea359e4a,host_traddr=nn-0x20000090fae0ec89:pn-
0x10000090fae0ec89 live non-optimized
```

NVMe/FC

```
nvme list-subsys /dev/nvme9n1
```

出力例を表示します。

```
nvme-subsy9 - NQN=nqn.1992-
08.com.netapp:sn.f8e2af201b7211f0ac2bd039eab67a95:subsystem.with
_inband_with_json hostnqn=nqn.2014-
08.org.nvmeexpress:uuid:4c4c4544-0035-5910-804b-b2c04f444d33
iopolicy=round-robin
\
+- nvme10 tcp
traddr=192.168.111.71,trsvcid=4420,src_addr=192.168.111.80 live
non-optimized
+- nvme11 tcp
traddr=192.168.211.70,trsvcid=4420,src_addr=192.168.211.80 live
optimized
+- nvme12 tcp
traddr=192.168.111.70,trsvcid=4420,src_addr=192.168.111.80 live
optimized
+- nvme9 tcp
traddr=192.168.211.71,trsvcid=4420,src_addr=192.168.211.80 live
non-optimized
```

5. 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	Size
NSID UUID			
/dev/nvme0n1	vs_161		
/vol/fc_nvme_vol1/fc_nvme_ns1 0797-428e-a577-fdb3f14d0dc3		1 32fd92c7-	5.37GB

JSON

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

例を示します

```
{  
    "Device": "/dev/nvme98n2",  
    "Vserver": "vs_161",  
    "Namespace_Path": "/vol/fc_nvme_vol71/fc_nvme_ns71",  
    "NSID": 2,  
    "UUID": "39d634c4-a75e-4fbe-ab00-3f9355a26e43",  
    "LBA_Size": 4096,  
    "Namespace_Size": 5368709120,  
    "UsedBytes": 430649344,  
}  
]  
}
```

ステップ8: 永続的な検出コントローラを作成する

ONTAP 9.11.1 以降では、SUSE Linux Enterprise Server 15 SP7 ホスト用の永続検出コントローラ (PDC) を作成できます。NVMe サブシステムの追加または削除操作と検出口ログ ページ データの変更を自動的に検出するには、PDC が必要です。

手順

- 検出ログページのデータが使用可能で、イニシエータポートとターゲットLIFの組み合わせから取得できることを確認します。

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

出力例を表示します。

```
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.4f7af2bd221811f0afadd039eab0dadd:discovery
traddr: 192.168.111.66
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.4f7af2bd221811f0afadd039eab0dadd:discovery
traddr: 192.168.211.66
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.4f7af2bd221811f0afadd039eab0dadd:discovery
traddr: 192.168.111.67
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.4f7af2bd221811f0afadd039eab0dadd:discovery
traddr:   192.168.211.67
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.4f7af2bd221811f0afadd039eab0dadd:subsystem.pdc
traddr:   192.168.111.66
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.4f7af2bd221811f0afadd039eab0dadd:subsystem.pdc
traddr:   192.168.211.66
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.4f7af2bd221811f0afadd039eab0dadd:subsystem.pdc
traddr:   192.168.111.67
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.4f7af2bd221811f0afadd039eab0dadd:subsystem.pdc  
traddr: 192.168.211.67  
eflags: none  
sectype: none
```

2. 検出サブシステムのPDCを作成します。

```
nvme discover -t <trtype> -w <host-traddr> -a <traddr> -p
```

次の出力が表示されます。

```
nvme discover -t tcp -w 192.168.111.80 -a 192.168.111.66 -p
```

3. ONTAPコントローラから、PDCが作成されたことを確認します。

```
vserver nvme show-discovery-controller -instance -vserver <vserver_name>
```

出力例を表示します。

```
vserver nvme show-discovery-controller -instance -vserver vs_pdc

    Vserver Name: vs_pdc
        Controller ID: 0101h
        Discovery Subsystem NQN: nqn.1992-
08.com.netapp:sn.4f7af2bd221811f0afadd039eab0dadd:discovery
        Logical Interface: lif2
            Node: A400-12-181
            Host NQN: nqn.2014-
08.org.nvmexpress:uuid:9796c1ec-0d34-11eb-b6b2-3a68dd3bab57
            Transport Protocol: nvme-tcp
        Initiator Transport Address: 192.168.111.80
        Transport Service Identifier: 8009
            Host Identifier: 9796c1ec0d3411ebb6b23a68dd3bab57
            Admin Queue Depth: 32
            Header Digest Enabled: false
            Data Digest Enabled: false
        Keep-Alive Timeout (msec): 30000
```

ステップ9: 安全なインバンド認証を設定する

ONTAP 9.12.1 以降では、ホストと ONTAP コントローラ間の NVMe/TCP および NVMe/FC 経由の安全なインバンド認証がサポートされます。

セキュアな認証を設定するには、各ホストまたはコントローラを DH-HMAC-CHAP キー。NVMeホストまたはコントローラのNQNと管理者が設定した認証シークレットを組み合わせたものです。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。

CLIまたは設定JSONファイルを使用して、セキュアなインバンド認証を設定できます。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。

手順

1. ホストNQNを取得します。

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

2. ホストの dhchap キーを生成します。

コマンドパラメータの出力を次に示し `gen-dhchap-key` ます。

```
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.nvmeexpress:uuid:e6dade64-216d-11ec-b7bb-7ed30a5482c3
DHHC-
1:03:1CFivw9ccz58gAcOUJrM7Vs98hd2ZHSr+iw+Amg6xZP15D2Yk+HDTZiUAg1iGgx
TYqnxukqvYeda55Bw3wtz6sJNpR4=:
```

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-  
subsys1/nvme*/dhchap_secret  
DHHC-1:01:iM63E6cX7G5SOKKOju8gmzM53qywsy+C/YwtzxhIt9ZRz+ky:  
DHHC-1:01:iM63E6cX7G5SOKKOju8gmzM53qywsy+C/YwtzxhIt9ZRz+ky:  
DHHC-1:01:iM63E6cX7G5SOKKOju8gmzM53qywsy+C/YwtzxhIt9ZRz+ky:  
DHHC-1:01:iM63E6cX7G5SOKKOju8gmzM53qywsy+C/YwtzxhIt9ZRz+ky:
```

- b. 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-
subsys6/nvme*/dhchap_ctrl_secret
DHHC-
1:03:1CFivw9ccz58gAcOUJrM7Vs98hd2ZHSr+iw+Amg6xZP15D2Yk+HDTziUA
g1iGgxTYqnxukqvYedA55Bw3wtz6sJNpR4=:
DHHC-
1:03:1CFivw9ccz58gAcOUJrM7Vs98hd2ZHSr+iw+Amg6xZP15D2Yk+HDTziUA
g1iGgxTYqnxukqvYedA55Bw3wtz6sJNpR4=:
DHHC-
1:03:1CFivw9ccz58gAcOUJrM7Vs98hd2ZHSr+iw+Amg6xZP15D2Yk+HDTziUA
g1iGgxTYqnxukqvYedA55Bw3wtz6sJNpR4=:
DHHC-
1:03:1CFivw9ccz58gAcOUJrM7Vs98hd2ZHSr+iw+Amg6xZP15D2Yk+HDTziUA
g1iGgxTYqnxukqvYedA55Bw3wtz6sJNpR4=:
```

JSON ファイル

ONTAPコントローラ構成で複数のNVMeサブシステムを使用できる場合は、コマンドでファイルを `nvme connect-all` 使用できます ` `/etc/nvme/config.json`。

JSONファイルを生成するには、オプションを使用し` -o` ます。その他の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。

手順

1. JSON ファイルを設定します。

出力例を表示します。

```
# cat /etc/nvme/config.json
[
{
    "hostnqn": "nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0035-
5910-804b-b2c04f444d33",
    "hostid": "4c4c4544-0035-5910-804b-b2c04f444d33",
    "dhchap_key": "DHHC-
1:01:i4i789R11sMuHLCY27RVI8XloC\GzjRwyhxip5hmIELsHrBq:",
    "subsystems": [
        {
            "nqn": "nqn.1992-
08.com.netapp:sn.f8e2af201b7211f0ac2bd039eab67a95:subsystem.sample_tcp_sub",
            "ports": [
                {
                    "transport": "tcp",
                    "traddr": "192.168.111.70",
                    "host_traddr": "192.168.111.80",
                    "trsvcid": "4420"
                    "dhchap_ctrl_key": "DHHC-
1:03:jqgYcJSKp73+XqAf2X6tvr9ngBpr2n0MGWbmZIZq4PieKZCoilKGef8lAvh
YS0PNK7T+04YD5CRPjh+m3qjJU++yR8s=:"
                },
                {
                    "transport": "tcp",
                    "traddr": "192.168.111.71",
                    "host_traddr": "192.168.111.80",
                    "trsvcid": "4420",
                    "dhchap_ctrl_key": "DHHC-
1:03:jqgYcJSKp73+XqAf2X6tvr9ngBpr2n0MGWbmZIZq4PieKZCoilKGef8lAvh
YS0PNK7T+04YD5CRPjh+m3qjJU++yR8s=:"
                },
                {
                    "transport": "tcp",
                    "traddr": "192.168.211.70",
                    "host_traddr": "192.168.211.80",
                    "trsvcid": "4420",
                    "dhchap_ctrl_key": "DHHC-
1:03:jqgYcJSKp73+XqAf2X6tvr9ngBpr2n0MGWbmZIZq4PieKZCoilKGef8lAvh
YS0PNK7T+04YD5CRPjh+m3qjJU++yR8s=:"
                },
                {
                    "transport": "tcp",

```

```
        "traddr": "192.168.211.71",
        "host_traddr": "192.168.211.80",
        "trsvcid": "4420",
        "dhchap_ctrl_key": "DHHC-
1:03:jqgYcJSKp73+XqAf2X6tWr9ngBpr2n0MGWbmZIZq4PieKZCoilKGef8lAvh
YS0PNK7T+04YD5CRPjh+m3qjJU++yR8s=:"}
    }
]
}
]
```



上記の例では、`host_traddr`に対応し、`dhchap_ctrl_key`に対応する`dhchap_secret`とし、`dhchap_ctrl_key`と`dhchap_ctrl_secret`を設定します。

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
```

出力例を表示します。

```
traddr=192.168.211.70 is already connected
traddr=192.168.111.71 is already connected
traddr=192.168.211.71 is already connected
traddr=192.168.111.70 is already connected
traddr=192.168.211.70 is already connected
traddr=192.168.111.70 is already connected
traddr=192.168.211.71 is already connected
traddr=192.168.111.71 is already connected
traddr=192.168.211.70 is already connected
traddr=192.168.111.71 is already connected
traddr=192.168.211.71 is already connected
traddr=192.168.111.70 is already connected
```

3. 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-subsy0/nvme0/dhchap_secret
```

次の出力が表示されます。

```
DHHC-1:01:i4i789R11sMuHLCY27RVI8X1oC/GzjRwyhxip5hmIELsHrBq:
```

- 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
```

次の出力が表示されます。

```
DHHC-
1:03:jqgYcJSKp73+XqAf2X6twr9ngBpr2n0MGWbmZIZq4PieKZCoilKGef8lAvhYS0P
NK7T+04YD5CRPjh+m3qqJU++yR8s=:
```

ステップ10: トランSPORT層セキュリティを構成する

Transport Layer Security (TLS) は、NVMe-oFホストとONTAPアレイの間のNVMe接続をエンドツーエンドでセキュアに暗号化します。.16.1以降では、CLIと設定済みの事前共有キー (PSK) を使用してONTAP 9.1.3を設定できます。

このタスクについて

この手順は、ONTAP コントローラで手順を実行するように指定されている場合を除き、SUSE Linux Enterprise Server ホストで実行します。

手順

1. 以下のものをお持ちかご確認ください ktls-utils、openssl、そして`libopenssl`ホストにインストールされているパッケージ:

- a. 確認する ktls-utils :

```
rpm -qa | grep ktls
```

次の出力が表示されます。

```
ktls-utils-0.10+33.g311d943-150700.1.5.x86_64
```

- a. SSL パッケージを確認します。

```
rpm -qa | grep ssl
```

出力例を表示します。

```
libopenssl3-3.2.3-150700.3.20.x86_64  
openssl-3-3.2.3-150700.3.20.x86_64  
libopenssl11_1-1.1.1w-150700.9.37.x86_64
```

2. 次の設定が正しいことを確認し `/etc/tlshd.conf` ます。

```
cat /etc/tlshd.conf
```

出力例を表示します。

```
[debug]  
loglevel=0  
tls=0  
nl=0  
[authenticate]  
keyrings=.nvme  
[authenticate.client]  
#x509.truststore= <pathname>  
#x509.certificate= <pathname>  
#x509.private_key= <pathname>  
[authenticate.server]  
#x509.truststore= <pathname>  
#x509.certificate= <pathname>  
#x509.private_key= <pathname>
```

3. システム起動時に起動するように有効にし `tlshd` ます。

```
systemctl enable tlshd
```

4. デーモンが実行されていることを確認し `tlshd` ます。

```
systemctl status tlshd
```

出力例を表示します。

```
tlshd.service - Handshake service for kernel TLS consumers
   Loaded: loaded (/usr/lib/systemd/system/tlshd.service; enabled;
preset: disabled)
     Active: active (running) since Wed 2024-08-21 15:46:53 IST; 4h
      57min ago
       Docs: man:tlshd(8)
    Main PID: 961 (tlshd)
      Tasks: 1
        CPU: 46ms
      CGroup: /system.slice/tlshd.service
              └─961 /usr/sbin/tlshd
Aug 21 15:46:54 RX2530-M4-17-153 tlshd[961]: Built from ktls-utils
0.11-dev on Mar 21 2024 12:00:00
```

5. を使用してTLS PSKを生成し `nvme gen-tls-key` ます。

a. ホストを確認します:

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

次の出力が表示されます。

```
nqn.2014-08.org.nvmeexpress:uuid:4c4c4544-0035-5910-804b-b2c04f444d33
```

b. キーを検証します:

```
nvme gen-tls-key --hmac=1 --identity=1 --subsysnqn= nqn.1992-
08.com.netapp:sn.a2d41235b78211efb57dd039eab67a95:subsystem.nvme1
```

次の出力が表示されます。

```
NVMeTLSkey-1:01:C50EsaGtuOp8n5fGE9EuWjbBCtshmf0Hx4XTqTJUmydf0gIj:
```

6. ONTAPコントローラで、ONTAPサブシステムにTLS PSKを追加します。

出力例を表示します。

```
nvme subsystem host add -vserver vs_iscsi_tcp -subsystem nvme1 -host -nqn nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0035-5910-804b-b2c04f444d33 -tls-configured-psk NVMeTLSkey-1:01:C50EsaGtuOp8n5fGE9EuWjbBCtshmfoHx4XTqTJUmydf0gIj:
```

7. TLS PSKをホストカーネルキーリングに挿入します。

```
nvme check-tls-key --identity=1 --subsysnqn=nqn.1992-08.com.netapp:sn.a2d41235b78211efb57dd039eab67a95:subsystem.nvme1 --keydata=NVMeTLSkey-1:01:C50EsaGtuOp8n5fGE9EuWjbBCtshmfoHx4XTqTJUmydf0gIj: --insert
```

次の TLS キーが表示されます。

```
Inserted TLS key 22152a7e
```



PSKは次のように表示されます NVMe1R01`使用するため `identity v1 TLS ハンドシェイク アルゴリズムから。Identity v1は、ONTAPがサポートする唯一のバージョンです。

8. TLS PSKが正しく挿入されていることを確認します。

```
cat /proc/keys | grep NVMe
```

出力例を表示します。

```
069f56bb I--Q---      5 perm 3b010000      0      0 psk          NVMe1R01
nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0035-5910-804b-b2c04f444d33
nqn.1992-
08.com.netapp:sn.a2d41235b78211efb57dd039eab67a95:subsystem.nvme1
oYVLelmiOwnvDjXKBmrnIgGVpFIBDJtc4hmQXE/36Sw=: 32
```

9. 挿入したTLS PSKを使用してONTAPサブシステムに接続します。

a. TLS PSK を検証します。

```
nvme connect -t tcp -w 192.168.111.80 -a 192.168.111.66 -n nqn.1992-08.com.netapp:sn.a2d41235b78211efb57dd039eab67a95:subsystem.nvme1 --tls_key=0x069f56bb -tls
```

次の出力が表示されます。

```
connecting to device: nvme0
```

a. list-subsyss を検証します。

```
nvme list-subsyss
```

出力例を表示します。

```
nvme-subsyss0 - NQN=nqn.1992-08.com.netapp:sn.a2d41235b78211efb57dd039eab67a95:subsystem.nvme1 hostnqn=nqn.2014-08.org.nvexpress:uuid:4c4c4544-0035-5910-804b-b2c04f444d33 \
+- nvme0 tcp
traddr=192.168.111.66,trsvcid=4420,host_traddr=192.168.111.80,src_addr=192.168.111.80 live
```

10. ターゲットを追加し、指定したONTAPサブシステムへのTLS接続を確認します。

```
nvme subsystem controller show -vserver sles15_tls -subsystem sles15 -instance
```

出力例を表示します。

```
(vserver nvme subsystem controller show)
    Vserver Name: vs_iscsi_tcp
    Subsystem: nvme1
    Controller ID: 0040h
    Logical Interface: tcpnvme_lif1_1
        Node: A400-12-181
        Host NQN: nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0035-5910-804b-b2c04f444d33
    Transport Protocol: nvme-tcp
    Initiator Transport Address: 192.168.111.80
        Host Identifier:
4c4c454400355910804bb2c04f444d33
    Number of I/O Queues: 2
    I/O Queue Depths: 128, 128
    Admin Queue Depth: 32
    Max I/O Size in Bytes: 1048576
    Keep-Alive Timeout (msec): 5000
    Subsystem UUID: 8bbfb403-1602-11f0-ac2b-
d039eab67a95
        Header Digest Enabled: false
        Data Digest Enabled: false
        Authentication Hash Function: sha-256
Authentication Diffie-Hellman Group: 3072-bit
        Authentication Mode: unidirectional
Transport Service Identifier: 4420
        TLS Key Type: configured
        TLS PSK Identity: NVMe1R01 nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0035-5910-804b-b2c04f444d33
nqn.1992-
08.com.netapp:sn.a2d41235b78211efb57dd039eab67a95:subsystem.nvme1
oYVLeLmiOwnvDjXKBmrnIgGVpFIBDJtc4hmQXE/36Sw=
        TLS Cipher: TLS-AES-128-GCM-SHA256
```

手順11：既知の問題を確認する

既知の問題はありません。

SUSE Linux Enterprise Server 15 SP6 with ONTAP向け のNVMe-oFホスト構成

NVMe over Fibre Channel (NVMe/FC) やその他のトランスポートを含むNVMe over

Fabrics (NVMe-oF) は、非対称ネームスペースアクセス (ANA) を備えたSUSE Linux Enterprise Server 15 SP6でサポートされます。NVMe-oF環境では、ANAはiSCSI環境およびFCP環境のALUAマルチパスに相当し、カーネル内NVMeマルチパスで実装されます。

SUSE Linux Enterprise Server 15 SP6 with ONTAPのNVMe-oFホスト構成では、次のサポートが利用できます。

- NVMeトライフィックとSCSIトライフィックを同じホストで実行している。たとえば、SCSI LUN用のSCSIデバイスにdm-multipathを設定し、NVMeマルチパスを使用してホストでNVMe-oFネームスペースデバイスを設定できます `mpath`。
- NVMe over TCP (NVMe/TCP) とNVMe/FCがサポートされます。これにより、標準パッケージのNetApp プラグイン `nvme-cli` で、NVMe/FCとNVMe/TCPの両方のネームスペースのONTAPの詳細を表示することができます。

サポートされる構成の詳細については、を参照してください "[Interoperability Matrix Tool](#)"。

の機能

- NVMeセキュア、インバンド認証のサポート
- 一意の検出NQNを使用した永続的検出コントローラ (PDC) のサポート
- NVMe/TCPのTLS 1.3暗号化のサポート

既知の制限

- 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。
- SUSE Linux Enterprise Server 15 SP6ホストのNVMe-oFでは、NetApp `sanlun` ホストユーティリティはサポートされません。代わりに、すべてのNVMe-oFトランSPORT用の標準パッケージに含まれているNetAppプラグインを利用できます ``nvme-cli``。

NVMe/FC を設定

NVMe / FCは、Broadcom/Emulex FCアダプタまたはMarvell/Qlogic FCアダプタを使用して、SUSE Linux Enterprise Server 15 SP6 with ONTAP構成用に設定できます。

Broadcom / Emulex

Broadcom/Emulex FCアダプタ用にNVMe/FCを設定

手順

- 推奨されるアダプタモデルを使用していることを確認します。

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

出力例

```
LPe32002 M2  
LPe32002-M2
```

- アダプタモデル概要を確認します。

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

出力例

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

- 推奨されるバージョンのEmulex Host Bus Adapter (HBA；ホストバスアダプタ) ファームウェアを使用していることを確認します。

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

出力例

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

- 推奨バージョンのlpfcドライバを使用していることを確認します。

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

出力例

```
0:14.4.0.1
```

5. 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
```

出力例

```
0x10000090fae0ec88  
0x10000090fae0ec89
```

6. 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
```

出力例

```
Online  
Online
```

7. 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
```

次の例では、1つのイニシエータポートが有効になっており、2つのターゲットLIFで接続されています。

出力例を表示します。

```
NVME Initiator Enabled
XRI Dist lpfc0 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc0 WWPN x10000090fae0ec88 WWNN x20000090fae0ec88
DID x0a1300 ONLINE
NVME RPORT WWPN x2070d039ea359e4a WWNN x206bd039ea359e4a DID
x0a0a05 TARGET DISCSRVC
ONLINE
NVME Statistics
LS: Xmt 00000003ba Cmpl 00000003ba Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 0000000014e3dfb8 Issue 0000000014e308db OutIO
fffffffffffff2923
abort 00000845 noxri 00000000 nondlp 00000063 qdepth 00000000
wqerr 00000003 err 00000000
FCP CMPL: xb 00000847 Err 00027f33
NVME Initiator Enabled
XRI Dist lpfc1 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc1 WWPN x10000090fae0ec89 WWNN x20000090fae0ec89
DID x0a1200 ONLINE
NVME RPORT WWPN x2071d039ea359e4a WWNN x206bd039ea359e4a DID
x0a0305 TARGET DISCSRVC
ONLINE
NVME Statistics
LS: Xmt 00000003ba Cmpl 00000003ba Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 0000000014e39f78 Issue 0000000014e2b832 OutIO
fffffffffffff18ba
abort 0000082d noxri 00000000 nondlp 00000028 qdepth 00000000
wqerr 00000007 err 00000000
FCP CMPL: xb 0000082d Err 000283bb
```

Marvell/QLogic

SUSE Linux Enterprise Server 15 SP6カーネルに含まれているネイティブの受信トレイqla2xxxドライバには、最新の修正が含まれています。これらの修正は、ONTAPのサポートに不可欠です。

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
```

出力例

```
QLE2742 FW:v9.14.01 DVR: v10.02.09.200-k  
QLE2742 FW:v9.14.01 DVR: v10.02.09.200-k
```

2. を確認します `ql2xnvmeenable` パラメータは1に設定されています。

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

想定される値は1です。

1MB の I/O サイズを有効にする（オプション）

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
```

NVMeサービスの確認

SUSE Linux Enterprise Server 15 SP6以降では、`nvmefc-boot-connections.service` NVMe/FC パッケージに含まれるブートサービスと `nvmf-autoconnect.service` ブートサービスが `nvme-cli` システムブート時に自動的に

有効になります。システムブートが完了したら、ブートサービスが有効になっていることを確認する必要があります。

手順

- 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; vendor preset: disabled)
     Active: inactive (dead) since Thu 2024-05-25 14:55:00 IST; 11min
             ago
   Process: 2108 ExecStartPre=/sbin/modprobe nvme-fabrics (code=exited,
             status=0/SUCCESS)
   Process: 2114 ExecStart=/usr/sbin/nvme connect-all (code=exited,
             status=0/SUCCESS)
     Main PID: 2114 (code=exited, status=0/SUCCESS)

systemd[1]: Starting Connect NVMe-oF subsystems automatically during
boot...
nvme[2114]: traddr=nn-0x201700a098fd4ca6:pn-0x201800a098fd4ca6 is
already connected
systemd[1]: nvmf-autoconnect.service: Deactivated successfully.
systemd[1]: Finished Connect NVMe-oF subsystems automatically during
boot.
```

- 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; vendor preset: enabled)
   Active: inactive (dead) since Thu 2024-05-25 14:55:00 IST; 11min ago
     Main PID: 1647 (code=exited, status=0/SUCCESS)

systemd[1]: Starting Auto-connect to subsystems on FC-NVME devices found during boot...
systemd[1]: nvmefc-boot-connections.service: Succeeded.
systemd[1]: Finished Auto-connect to subsystems on FC-NVME devices found during boot.
```

NVMe/FC を設定

NVMe/TCPには自動接続機能はありません。代わりに、NVMe/TCPまたは`connect-all`の処理を手動で実行することで、NVMe/TCPサブシステムとネームスペースを検出できます`connect`。

手順

1. イニシエータポートがサポートされているNVMe/TCP LIFの検出ログページの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>
```

出力例を表示します。

```
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.8b5ee9199ff411eea468d039ea36a106:discovery
traddr: 192.168.211.67
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.8b5ee9199ff411eea468d039ea36a106:discovery
traddr: 192.168.111.67
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.8b5ee9199ff411eea468d039ea36a106:discovery
traddr: 192.168.211.66
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.8b5ee9199ff411eea468d039ea36a106:discovery
traddr: 192.168.111.66
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.8b5ee9199ff411eea468d039ea36a106:subsystem.nvme_tcp
_1
traddr: 192.168.211.67
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.8b5ee9199ff411eea468d039ea36a106:subsystem.nvme_tcp
_1
traddr: 192.168.111.67
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.8b5ee9199ff411eea468d039ea36a106:subsystem.nvme_tcp
_1
```

```
traddr: 192.168.211.66
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.8b5ee9199ff411eea468d039ea36a106:subsystem.nvme_tcp
_1
traddr: 192.168.111.66
eflags: none
sectype: none
```

2. NVMe/TCPイニシエータとターゲットLIFの他のすべての組み合わせで、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>
```

出力例

```
nvme discover -t tcp -w 192.168.111.79 -a 192.168.111.66
nvme discover -t tcp -w 192.168.111.79 -a 192.168.111.67
nvme discover -t tcp -w 192.168.211.79 -a 192.168.211.66
nvme discover -t tcp -w 192.168.211.79 -a 192.168.211.67
```

3. 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>
```

出力例

```
nvme connect-all -t tcp -w 192.168.111.79 -a 192.168.111.66
nvme connect-all -t tcp -w 192.168.111.79 -a 192.168.111.67
nvme connect-all -t tcp -w 192.168.211.79 -a 192.168.211.66
nvme connect-all -t tcp -w 192.168.211.79 -a 192.168.211.67
```



SUSE Linux Enterprise Server 15 SP6以降では、NVMe/TCPタイムアウトのデフォルト設定が `ctrl-loss-tmo` オフになっています。つまり、再試行回数に制限はなく（無期限の再試行）、コマンドまたは `nvme connect-all` コマンド（オプション `--1`）を使用する場合は、特定のタイムアウト期間 `nvme connect` を手動で設定する必要はありません `ctrl-loss-tmo`。また、NVMe / TCPコントローラでは、パス障害が発生して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。

NVMe-oF を検証します

SUSE Linux Enterprise Server 15 SP6 with ONTAP構成でNVMe-oFを検証するには、次の手順を実行します。

手順

1. カーネル内の 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
```

想定される値は「Y」です。

2. ホストのコントローラモデルがONTAP NVMeネームスペースに対応していることを確認します。

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

出力例

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

3. それぞれのONTAP NVMe I/OコントローラのNVMe I/Oポリシーを確認します。

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

出力例

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

4. ONTAPネームスペースがホストから認識されることを確認します。

```
nvme list -v
```

出力例を表示します。

```
Subsystem      Subsystem=NQN
Controllers
-----
-----
nvme-subsy0    nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.unidir_d
hcha p nvme0, nvme1, nvme2, nvme3

Device   SN           MN
FR       TxPort Asdress   Subsystem   Namespaces
-----
-----
nvme0    81LGgBUqsI3EAAAAAAAE NetApp ONTAP Controller  FFFFFFFF tcp
traddr=192.168.111.66,trsvcid=4420,host_traddr=192.168.111.79 nvme-
subsyst0 nvme0n1
nvme1    81LGgBUqsI3EAAAAAAAE NetApp ONTAP Controller  FFFFFFFF tcp
traddr=192.168.111.67,trsvcid=4420,host_traddr=192.168.111.79 nvme-
subsyst0 nvme0n1
nvme2    81LGgBUqsI3EAAAAAAAE NetApp ONTAP Controller  FFFFFFFF tcp
traddr=192.168.211.66,trsvcid=4420,host_traddr=192.168.211.79 nvme-
subsyst0 nvme0n1
nvme3    81LGgBUqsI3EAAAAAAAE NetApp ONTAP Controller  FFFFFFFF tcp
traddr=192.168.211.67,trsvcid=4420,host_traddr=192.168.211.79 nvme-
subsyst0 nvme0n1
Device      Generic     NSID      Usage          Format
Controllers
-----
-----
/dev/nvme0n1 /dev/ng0n1  0x1      1.07  GB /  1.07  GB    4 KiB +
0 B      nvme0, nvme1, nvme2, nvme3
```

5. 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-subsy /dev/<subsystem_name>
```

NVMe/FC

```
nvme list-subsys /dev/nvme2n1
```

出力例を表示します。

```
nvme-subsys2 - NQN=nqn.1992-
08.com.netapp:sn.06303c519d8411eea468d039ea36a106:sub-
system.nvme
  hostnqn=nqn.2014-08.org.nvmeexpress:uuid:4c4c4544-
  0056-5410-8048-c6c04f425633
    iopolicy=round-robin
  \
  +- nvme4 fc traddr=nn-0x208fd039ea359e4a:pn-
  0x210dd039ea359e4a,host_traddr=nn-0x2000f4c7aa0cd7ab:pn-
  0x2100f4c7aa0cd7ab live optimized
  +- nvme6 fc traddr=nn-0x208fd039ea359e4a:pn-
  0x210ad039ea359e4a,host_traddr=nn-0x2000f4c7aa0cd7aa:pn-
  0x2100f4c7aa0cd7aa live optimized
```

NVMe/FC

```
nvme list-subsys
```

出力例を表示します。

```
nvme-subsy1 - NQN=nqn.1992-
08.com.netapp:sn.8b5ee9199ff411eea468d039ea36a106:subsystem.nvme
_tcp_1
hostnqn=nqn.2014-08.org.nvmeexpress:uuid:4c4c4544-0035-5910-
804b-b2c04f444d33
iopolicy=round-robin
\
+- nvme4 tcp
traddr=192.168.111.66,trsvcid=4420,host_traddr=192.168.111.79,src_addr=192.168.111.79 live
+- nvme3 tcp
traddr=192.168.211.66,trsvcid=4420,host_traddr=192.168.211.79,src_addr=192.168.111.79 live
+- nvme2 tcp
traddr=192.168.111.67,trsvcid=4420,host_traddr=192.168.111.79,src_addr=192.168.111.79 live
+- nvme1 tcp
traddr=192.168.211.67,trsvcid=4420,host_traddr=192.168.211.79,src_addr=192.168.111.79 live
```

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
```

出力例

Device NSID	Vserver UUID	Namespace Path	Size
/dev/nvme0n1 c6586535-da8a-40fa-8c20-759ea0d69d33	vs_192	/vol/fcnvme_vol_1_1_0/fcnvme_ns	1 20GB

JSON

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

出力例を表示します。

```
{
  "ONTAPdevices": [
    {
      "Device": "/dev/nvme0n1",
      "Vserver": "vs_192",
      "Namespace_Path": "/vol/fcnvme_vol_1_1_0/fcnvme_ns",
      "NSID": 1,
      "UUID": "c6586535-da8a-40fa-8c20-759ea0d69d33",
      "Size": "20GB",
      "LBA_Data_Size": 4096,
      "Namespace_Size": 262144
    }
  ]
}
```

永続的検出コントローラの作成

ONTAP 9.11.1以降では、SUSE Linux Enterprise Server 15 SP6ホスト用の永続的検出コントローラ（PDC）を作成できます。NVMeサブシステムの追加または削除処理や検出口ログページのデータに対する変更を自動的に検出するには、PDCが必要です。

手順

1. 検出ログページのデータが使用可能で、イニシエータポートとターゲットLIFの組み合わせから取得できることを確認します。

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

出力例を表示します。

```
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.8b5ee9199ff411eea468d039ea36a106:discovery
traddr: 192.168.211.67
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.8b5ee9199ff411eea468d039ea36a106:discovery
traddr: 192.168.111.67
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.8b5ee9199ff411eea468d039ea36a106:discovery
traddr: 192.168.211.66
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.8b5ee9199ff411eea468d039ea36a106:discovery
traddr: 192.168.111.66
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.8b5ee9199ff411eea468d039ea36a106:subsystem.nvme_tcp
_1
traddr: 192.168.211.67
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.8b5ee9199ff411eea468d039ea36a106:subsystem.nvme_tcp
_1
traddr: 192.168.111.67
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.8b5ee9199ff411eea468d039ea36a106:subsystem.nvme_tcp
_1
```

```
traddr: 192.168.211.66
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.8b5ee9199ff411eea468d039ea36a106:subsystem.nvme_tcp
_1
traddr: 192.168.111.66
eflags: none
sectype: none
```

2. 検出サブシステムのPDCを作成します。

```
nvme discover -t <trtype> -w <host-traddr> -a <traddr> -p
```

出力例

```
nvme discover -t tcp -w 192.168.111.79 -a 192.168.111.666 -p
```

3. ONTAPコントローラから、PDCが作成されたことを確認します。

```
vserver nvme show-discovery-controller -instance -vserver <vserver_name>
```

出力例を表示します。

```
vserver nvme show-discovery-controller -instance -vserver vs_nvme79
Vserver Name: vs_CLIENT116 Controller ID: 00C0h
Discovery Subsystem NQN: nqn.1992-
08.com.netapp:sn.48391d66c0a611ecaaa5d039ea165514:discovery Logical
Interface UUID: d23cbb0a-c0a6-11ec-9731-d039ea165abc Logical
Interface:
CLIENT116_lif_4a_1
Node: A400-14-124
Host NQN: nqn.2014-08.org.nvmexpress:uuid:12372496-59c4-4d1b-be09-
74362c0c1afc
Transport Protocol: nvme-tcp
Initiator Transport Address: 192.168.1.16
Host Identifier: 59de25be738348f08a79df4bce9573f3 Admin Queue Depth:
32
Header Digest Enabled: false Data Digest Enabled: false
Vserver UUID: 48391d66-c0a6-11ec-aaa5-d039ea165514
```

セキュアなインバンド認証のセットアップ

SuSE Linux Enterprise Server 15 SP6ホストとコントローラ間で、NVMe/TCPおよびNVMe/FC経由でのセキュアなインバンド認証がサポートされます。12.1以降ではONTAP 9、SUSE Linux Enterprise Server 15 SP6 ホストとONTAPコントローラ間でセキュアなインバンド認証がサポートされます。

セキュアな認証を設定するには、各ホストまたはコントローラを DH-HMAC-CHAP キー。NVMeホストまたはコントローラのNQNと管理者が設定した認証シークレットを組み合わせたものです。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。

CLIまたは設定JSONファイルを使用して、セキュアなインバンド認証を設定できます。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。

手順

1. ホストNQNを取得します。

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

2. SUSE Linux Enterprise Server 15 SP6ホストのDHCHAPキーを生成します。

コマンドパラメータの出力を次に示し `gen-dhchap-key` ます。

```
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.nvmeexpress:uuid:d3ca725a-ac8d-4d88-b46a-174ac235139b
DHHC-
1:03:J2UJQfj9f0pLnpF/ASDJRTyILKJRr5CougGpGdQSysPrLu6RW1fG15VSjbeDF1n
1DEh3nVBe19nQ/LxreSBeH/bx/pU=:
```

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-  
1:03:je1nQCmjJLUKD62mpYbzlpwu0OIws86NB96uNO/t3jbvhp7fjyR9bIRjO  
Hg8wQteyelJCFSMkBQH3pTKGdYR1OV9gx00=:  
DHHC-  
1:03:je1nQCmjJLUKD62mpYbzlpwu0OIws86NB96uNO/t3jbvhp7fjyR9bIRjO  
Hg8wQteyelJCFSMkBQH3pTKGdYR1OV9gx00=:  
DHHC-  
1:03:je1nQCmjJLUKD62mpYbzlpwu0OIws86NB96uNO/t3jbvhp7fjyR9bIRjO  
Hg8wQteyelJCFSMkBQH3pTKGdYR1OV9gx00=:  
DHHC-  
1:03:je1nQCmjJLUKD62mpYbzlpwu0OIws86NB96uNO/t3jbvhp7fjyR9bIRjO  
Hg8wQteyelJCFSMkBQH3pTKGdYR1OV9gx00=:
```

- 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-
subsys6/nvme*/dhchap_ctrl_secret
DHHC-
1:03:WorVEV83eYO53kV4Ie150pphbX5LApH03F8fgH3913t1rkSGDBJTt3crX
eTUB8fCwGbPsEyz6CXxdQJi6kbn4IzmkFU=:
DHHC-
1:03:WorVEV83eYO53kV4Ie150pphbX5LApH03F8fgH3913t1rkSGDBJTt3crX
eTUB8fCwGbPsEyz6CXxdQJi6kbn4IzmkFU=:
DHHC-
1:03:WorVEV83eYO53kV4Ie150pphbX5LApH03F8fgH3913t1rkSGDBJTt3crX
eTUB8fCwGbPsEyz6CXxdQJi6kbn4IzmkFU=:
DHHC-
1:03:WorVEV83eYO53kV4Ie150pphbX5LApH03F8fgH3913t1rkSGDBJTt3crX
eTUB8fCwGbPsEyz6CXxdQJi6kbn4IzmkFU=:
```

JSON ファイル

ONTAPコントローラ構成で複数のNVMeサブシステムを使用できる場合は、コマンドでファイルを `nvme connect-all` 使用できます ` `/etc/nvme/config.json`。

JSONファイルを生成するには、オプションを使用し` -o` ます。その他の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。

手順

1. JSON ファイルを設定します。

出力例を表示します。

```
cat /etc/nvme/config.json
[
  {
    "hostnqn": "nqn.2014-08.org.nvmexpress:uuid:12372496-59c4-
4d1b-be09-74362c0c1afc",
    "hostid": "3ae10b42-21af-48ce-a40b-cfb5bad81839",
    "dhchap_key": "DHHC-
1:03:Cu3ZZfIz1WMlqZFnCMqpAgn/T6EVOcIFHez215U+Pow8jTgBF2UbNk3DK4w
fk2EptWpna1rpwG5CndpOgxpRvh9m41w=:"
  },
  {
    "hostnqn": "nqn.2014-08.org.nvmexpress:uuid:12372496-59c4-
4d1b-be09-74362c0c1afc",
    "subsystems": [
      {
        "nqn": "nqn.1992-
08.com.netapp:sn.48391d66c0a611ecaaa5d039ea165514:subsystem.subs
ys_CLIENT116",
        "ports": [
          {
            "transport": "tcp",
            "traddr": " 192.168.111.66 ",
            "host_traddr": " 192.168.111.79",
            "trsvid": "4420",
            "dhchap_ctrl_key": "DHHC-
1:01:0h58bcT/uu0rCpGsDYU6ZHvRuVqsYKuBRS0Nu0VPx5HEwaZ:"
          },
          {
            "transport": "tcp",
            "traddr": " 192.168.111.66 ",
            "host_traddr": " 192.168.111.79",
            "trsvid": "4420",
            "dhchap_ctrl_key": "DHHC-
1:01:0h58bcT/uu0rCpGsDYU6ZHvRuVqsYKuBRS0Nu0VPx5HEwaZ:"
          },
          {
            "transport": "tcp",
            "traddr": " 192.168.111.66 ",
            "host_traddr": " 192.168.111.79",
            "trsvid": "4420",
            "dhchap_ctrl_key": "DHHC-
1:01:0h58bcT/uu0rCpGsDYU6ZHvRuVqsYKuBRS0Nu0VPx5HEwaZ:"
          }
        ]
      }
    ]
  }
]
```

```
{  
    "transport": "tcp",  
    "traddr": "192.168.111.66",  
    "host_traddr": "192.168.111.79",  
    "trsvcid": "4420",  
    "dhchap_ctrl_key": "DHHC-  
1:01:0h58bcT/uu0rCpGsDYU6ZHvRuVqsYKuBRS0Nu0VPx5HEwaZ:"  
}  
]  
}  
]  
}  
]
```

+



上記の例では、`は`に対応し、`は dhchap_key`に対応 `dhchap_secret` し
`dhchap_ctrl_key` `dhchap_ctrl_secret` ます。

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
```

出力例を表示します。

```
traddr=192.168.111.66 is already connected  
traddr=192.168.211.66 is already connected  
traddr=192.168.111.66 is already connected  
traddr=192.168.211.66 is already connected  
traddr=192.168.111.66 is already connected  
traddr=192.168.211.66 is already connected  
traddr=192.168.111.67 is already connected  
traddr=192.168.211.67 is already connected  
traddr=192.168.111.67 is already connected  
traddr=192.168.211.67 is already connected  
traddr=192.168.111.67 is already connected  
traddr=192.168.111.67 is already connected
```

3. 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-subsy0/nvme0/dhchap_secret
```

出力例

```
DHHC-1:01:NunEWY7AZ1XqxITGheByarwZdQvU4ebZg9HOjIr6nOHEkxJg:
```

- b. コントローラのDHCHAPキーを確認します。

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

出力例

```
DHHC-  
1:03:2YJinsxa2v3+m8qqCiTnmgBZoH6mIT6G/6f0aGO8viVZB4VLNLH4z8CvK7pV  
YxN6S5fOAtaU3DNi12rieRMfdbg3704=:
```

Transport Layer Securityの設定

Transport Layer Security (TLS) は、NVMe-oFホストとONTAPアレイの間のNVMe接続をエンドツーエンドでセキュアに暗号化します。.16.1以降では、CLIと設定済みの事前共有キー (PSK) を使用してONTAP 9.1.3を設定できます。

このタスクについて

この手順は、ONTAP コントローラで手順を実行するように指定されている場合を除き、SUSE Linux Enterprise Server ホストで実行します。

手順

1. 次のktls-utils、openssl、libopensslパッケージがホストにインストールされていることを確認します。

```
a. rpm -qa | grep ktls
```

出力例

```
ktls-utils-0.10+12.gc3923f7-150600.1.2.x86_64
```

```
b. rpm -qa | grep ssl
```

出力例

```
openssl-3-3.1.4-150600.5.7.1.x86_64  
libopenssl1_1-1.1.1w-150600.5.3.1.x86_64  
libopenssl3-3.1.4-150600.5.7.1.x86_64
```

2. 次の設定が正しいことを確認し `/etc/tlshd.conf` ます。

```
cat /etc/tlshd.conf
```

出力例を表示します。

```
[debug]  
loglevel=0  
tls=0  
nl=0  
[authenticate]  
keyrings=.nvme  
[authenticate.client]  
#x509.truststore= <pathname>  
#x509.certificate= <pathname>  
#x509.private_key= <pathname>  
[authenticate.server]  
#x509.truststore= <pathname>  
#x509.certificate= <pathname>  
#x509.private_key= <pathname>
```

3. システム起動時に起動するように有効にし `tlshd` ます。

```
systemctl enable tlshd
```

4. デーモンが実行されていることを確認し `tlshd` ます。

```
systemctl status tlshd
```

出力例を表示します。

```
tlshd.service - Handshake service for kernel TLS consumers
   Loaded: loaded (/usr/lib/systemd/system/tlshd.service; enabled;
preset: disabled)
     Active: active (running) since Wed 2024-08-21 15:46:53 IST; 4h
      57min ago
       Docs: man:tlshd(8)
    Main PID: 961 (tlshd)
      Tasks: 1
        CPU: 46ms
      CGroup: /system.slice/tlshd.service
              └─961 /usr/sbin/tlshd
Aug 21 15:46:54 RX2530-M4-17-153 tlshd[961]: Built from ktls-utils
0.11-dev on Mar 21 2024 12:00:00
```

5. を使用してTLS PSKを生成し `nvme gen-tls-key` ます。

a. cat /etc/nvme/hostnqn

出力例

```
nqn.2014-08.org.nvmeexpress:uuid:e58eca24-faff-11ea-8fee-3a68dd3b5c5f
```

b. nvme gen-tls-key --hmac=1 --identity=1 --subsysnqn=nqn.1992
-08.com.netapp:sn.1d59a6b2416b11ef9ed5d039ea50acb3:subsystem.sles15

出力例

```
NVMeTLSkey-1:01:dNcby017axByCko8GivzO09zGlgHDXJCN6KLzvYoA+NpT1uD:
```

6. ONTAP アレイで、TLS PSK を ONTAP サブシステムに追加します。

```
vserver nvme subsystem host add -vserver sles15_tls -subsystem sles15
-host-nqn nqn.2014-08.org.nvmeexpress:uuid:e58eca24-faff-11ea-8fee-
3a68dd3b5c5f -tls-configured-psk NVMeTLSkey-
1:01:dNcby017axByCko8GivzO09zGlgHDXJCN6KLzvYoA+NpT1uD:
```

7. SUSE Linux Enterprise Server ホストで、TLS PSK をホスト カーネル キーリングに挿入します。

```
nvme check-tls-key --identity=1 --subsysnqn =nqn.1992-  
08.com.netapp:sn.1d59a6b2416b11ef9ed5d039ea50acb3:subsystem.sles15  
--keydata=NVMeTLSkey  
-1:01:dNcby017axByCko8GivzO09zGlgHDXJCN6KLzvYoA+NpTluD: --insert
```

出力例

```
Inserted TLS key 22152a7e
```



PSKはTLSハンドシェイクアルゴリズムの「identity v1」を使用するため、「NVMe1R01」と表示されます。Identity v1は、ONTAPがサポートする唯一のバージョンです。

8. TLS PSKが正しく挿入されていることを確認します。

```
cat /proc/keys | grep NVMe
```

出力例

```
22152a7e I--Q--- 1 perm 3b010000 0 0 psk NVMe1R01  
nqn.2014-08.org.nvmeexpress:uuid:ffa0c815-e28b-4bb1-8d4c-7c6d5e610bfc  
nqn.1992-  
08.com.netapp:sn.1d59a6b2416b11ef9ed5d039ea50acb3:subsystem.sles15  
UoP9dEfvuCUzzps0DYxnshKDapZYmvA0/RJJ8JAqmAo=: 32
```

9. SUSE Linux Enterprise Server ホストで、挿入された TLS PSK を使用して ONTAP サブシステムに接続します。

a. `nvme connect -t tcp -w 20.20.10.80 -a 20.20.10.14 -n nqn.1992-
08.com.netapp:sn.1d59a6b2416b11ef9ed5d039ea50acb3:subsystem.sles15
--tls_key=0x22152a7e --tls`

出力例

```
connecting to device: nvme0
```

b. `nvme list-subsy`

出力例

```
nvme-subsys0 - NQN=nqn.1992-  
08.com.netapp:sn.1d59a6b2416b11ef9ed5d039ea50acb3:subsystem.sles15  
hostnqn=nqn.2014-08.org.nvmeexpress:uuid:ffa0c815-e28b-  
4bb1-8d4c-7c6d5e610bfc  
iopolicy=round-robin  
\  
+- nvme0 tcp  
traddr=20.20.10.14,trsvcid=4420,host_traddr=20.20.10.80,src_addr=20.2  
0.10.80 live
```

10. ターゲットを追加し、指定したONTAPサブシステムへのTLS接続を確認します。

```
nvme subsystem controller show -vserver sles15_tls -subsystem sles15 -instance
```

出力例を表示します。

```
(vserver nvme subsystem controller show)
    Vserver Name: sles15_tls
        Subsystem: sles15
            Controller ID: 0040h
                Logical Interface: sles15t_e1a_1
                    Node: A900-17-174
                        Host NQN: nqn.2014-
08.org.nvmexpress:uuid:ffa0c815-e28b-4bb1-8d4c-7c6d5e610bfc
                        Transport Protocol: nvme-tcp
                        Initiator Transport Address: 20.20.10.80
                        Host Identifier:
ffa0c815e28b4bb18d4c7c6d5e610bfc
                        Number of I/O Queues: 4
                        I/O Queue Depths: 128, 128, 128, 128
                        Admin Queue Depth: 32
                        Max I/O Size in Bytes: 1048576
                        Keep-Alive Timeout (msec): 5000
                        Vserver UUID: 1d59a6b2-416b-11ef-9ed5-
d039ea50acb3
                        Subsystem UUID: 9b81e3c5-5037-11ef-8a90-
d039ea50ac83
                        Logical Interface UUID: 8185dcac-5035-11ef-8abb-
d039ea50acb3
                        Header Digest Enabled: false
                        Data Digest Enabled: false
                        Authentication Hash Function: -
                        Authentication Diffie-Hellman Group: -
                            Authentication Mode: none
                        Transport Service Identifier: 4420
                            TLS Key Type: configured
                            TLS PSK Identity: NVMe1R01 nqn.2014-
08.org.nvmexpress:uuid:ffa0c815-e28b-4bb1-8d4c-7c6d5e610bfc
nqn.1992-
08.com.netapp:sn.1d59a6b2416b11ef9ed5d039ea50acb3:subsystem.sles15
UoP9dEfvuCUzzpS0DYxnshKDapZYmvA0/RJJ8JAqmAo=
                            TLS Cipher: TLS-AES-128-GCM-SHA256
```

既知の問題

既知の問題はありません。

SUSE Linux Enterprise Server 15 SP5 with ONTAP向けのNVMe-oFホスト構成

NVMe over Fibre Channel (NVMe/FC) やその他のトランスポートを含むNVMe over Fabrics (NVMe-oF) は、非対称ネームスペースアクセス (ANA) を備えたSUSE Linux Enterprise Server 15 SP5でサポートされます。NVMe-oF環境では、ANAはiSCSI環境およびFCP環境のALUAマルチパスに相当し、カーネル内NVMeマルチパスで実装されます。

SUSE Linux Enterprise Server 15 SP5 with ONTAPのNVMe-oFホスト構成では、次のサポートが利用できます。

- NVMeとSCSIの両方のトラフィックを同じホストで実行できます。そのため、SCSI LUNの場合はSCSI mpathデバイスにdm-multipathを設定できますが、NVMeマルチパスを使用してホスト上のNVMe-oFネームスペースデバイスを設定することができます。
- NVMe/FCに加えて、NVMe over TCP (NVMe/TCP) もサポートされます。NetAppプラグインをネイティブにインストールします nvme-cli [パッケージ]には、NVMe/FCとNVMe/TCPの両方のネームスペースのONTAPの詳細が表示されます。

サポートされる構成の詳細については、を参照してください "[Interoperability Matrix Tool](#)"。

の機能

- NVMeセキュア、インバンド認証のサポート
- 一意の検出NQNを使用した永続的検出コントローラ (PDC) のサポート

既知の制限

- NVMe-oFプロトコルを使用したSANポートは現在サポートされていません。
- `sanlun`NVMe-oFはサポートされていません。そのため、SUSE Linux Enterprise Server 15 SP5ホストのNVMe-oFではホストユーティリティのサポートを利用できません。すべてのNVMe-oF転送で、標準のNVMe-CLIパッケージに含まれているNetAppプラグインを使用できます。

NVMe/FC を設定

NVMe/FCは、Broadcom/Emulex FCアダプタまたはMarvell/Qlogic FCアダプタに設定できます。

Broadcom / Emulex

手順

- 推奨されるアダプタモデルを使用していることを確認します。

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

出力例：

```
LPe32002 M2  
LPe32002-M2
```

- アダプタモデル概要を確認します。

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

出力例：

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

- 推奨されるバージョンのEmulex Host Bus Adapter (HBA；ホストバスアダプタ) ファームウェアを使用していることを確認します。

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

出力例：

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

- 推奨バージョンのlpfcドライバを使用していることを確認します。

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

出力例：

```
0:14.2.0.13
```

5. 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
```

出力例：

```
0x100000109b579d5e  
0x100000109b579d5f
```

6. 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
```

出力例：

```
Online  
Online
```

7. 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
```

出力例：

次の例では、1つのイニシエータポートが有効になっており、2つのターゲットLIFで接続されています。

```

NVME Initiator Enabled
XRI Dist lpfc0 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc0 WWPN x100000109b579d5e WWNN x200000109b579d5e DID
x011c00 ONLINE
NVME RPORT WWPN x208400a098dfdd91 WWNN x208100a098dfdd91 DID x011503
TARGET DISCSRVC ONLINE
NVME RPORT WWPN x208500a098dfdd91 WWNN x208100a098dfdd91 DID x010003
TARGET DISCSRVC *ONLINE

NVME Statistics
LS: Xmt 0000000e49 Cmpl 0000000e49 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 00000003ceb594f Issue 000000003ce65dbe OutIO
ffffffffffffb046f
abort 00000bd2 noxri 00000000 nondlp 00000000 qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 000014f4 Err 00012abd

NVME Initiator Enabled
XRI Dist lpfc1 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc1 WWPN x100000109b579d5f WWNN x200000109b579d5f DID
x011b00 ONLINE
NVME RPORT WWPN x208300a098dfdd91 WWNN x208100a098dfdd91 DID x010c03
TARGET DISCSRVC ONLINE
NVME RPORT WWPN x208200a098dfdd91 WWNN x208100a098dfdd91 DID x012a03
TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 0000000e50 Cmpl 0000000e50 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 00000003c9859ca Issue 000000003c93515e OutIO
fffffffffffffaf794
abort 00000b73 noxri 00000000 nondlp 00000000 qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 0000159d Err 000135c3

```

8. ホストをリブートします。

Marvell/QLogic

SUSE Linux Enterprise Server 15 SP5カーネルに含まれているネイティブの受信トレイqla2xxxドライバには、最新の修正が含まれています。これらの修正は、ONTAPのサポートに不可欠です。

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
```

出力例：

```
QLE2742 FW:v9.12.01 DVR: v10.02.08.300-k  
QLE2742 FW:v9.12.01 DVR: v10.02.08.300-k
```

2. を確認します `ql2xnvmeenable` パラメータは1に設定されています。

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

1MB の I/O サイズを有効にする（オプション）

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
```

NVMeサービスを有効にする

には2つのNVMe/FCブートサービスが含まれています。`nvme-cli` パッケージ (*only*) `nvmefc-boot-connections.service` システム起動中に起動できるようになっています。`nvmf-`

`autoconnect.service` が有効になつていません。そのため、手動で有効にする必要があります。`nvmf-autoconnect.service` システム起動中に起動します。

手順

1. 有効にします `nvmf-autoconnect.service` :

```
# systemctl enable nvmf-autoconnect.service
Created symlink /etc/systemd/system/default.target.wants/nvmf-
autoconnect.service → /usr/lib/systemd/system/nvmf-autoconnect.service.
```

2. ホストをリブートします。
3. システムのブート後に `nvmefc-boot-connections.service` が実行されていることを確認し `nvmf-autoconnect.service` ます。

出力例：

```

# systemctl status nvmf-autoconnect.service
nvmf-autoconnect.service - Connect NVMe-oF subsystems automatically
during boot
Loaded: loaded (/usr/lib/systemd/system/nvmf-autoconnect.service;
enabled; vendor preset: disabled)
Active: inactive (dead) since Thu 2023-05-25 14:55:00 IST; 11min
ago
Process: 2108 ExecStartPre=/sbin/modprobe nvme-fabrics (code=exited,
status=0/SUCCESS)
Process: 2114 ExecStart=/usr/sbin/nvme connect-all (code=exited,
status=0/SUCCESS)
Main PID: 2114 (code=exited, status=0/SUCCESS)

systemd[1]: Starting Connect NVMe-oF subsystems automatically during
boot...
nvme[2114]: traddr=nn-0x201700a098fd4ca6:pn-0x201800a098fd4ca6 is
already connected
systemd[1]: nvmf-autoconnect.service: Deactivated successfully.
systemd[1]: Finished Connect NVMe-oF subsystems automatically during
boot.

# systemctl status nvmefc-boot-connections.service
nvmefc-boot-connections.service - Auto-connect to subsystems on FC-NVME
devices found during boot
Loaded: loaded (/usr/lib/systemd/system/nvmefc-boot-
connections.service; enabled; vendor preset: enabled)
Active: inactive (dead) since Thu 2023-05-25 14:55:00 IST; 11min ago
Main PID: 1647 (code=exited, status=0/SUCCESS)

systemd[1]: Starting Auto-connect to subsystems on FC-NVME devices found
during boot...
systemd[1]: nvmefc-boot-connections.service: Succeeded.
systemd[1]: Finished Auto-connect to subsystems on FC-NVME devices found
during boot.

```

NVMe/FC を設定

NVMe/TCPの設定には、次の手順を使用できます。

手順

1. イニシエータポートがサポートされているNVMe/TCP LIFの検出ログページの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>
```

出力例：

```
# nvme discover -t tcp -w 192.168.1.4 -a 192.168.1.31

Discovery Log Number of Records 8, Generation counter 18
=====Discovery Log Entry 0===== trtype: tcp
adrfam: ipv4
subtype: current discovery subsystem treq: not specified
portid: 0
trsvcid: 8009 subnqn: nqn.1992-
08.com.netapp:sn.48391d66c0a611ecaaa5d039ea165514:discovery traddr:
192.168.2.117
eflags: explicit discovery connections, duplicate discovery information
sectype: none
=====Discovery Log Entry 1===== trtype: tcp
adrfam: ipv4
subtype: current discovery subsystem treq: not specified
portid: 1
trsvcid: 8009 subnqn: nqn.1992-
08.com.netapp:sn.48391d66c0a611ecaaa5d039ea165514:discovery traddr:
192.168.1.117
eflags: explicit discovery connections, duplicate discovery information
sectype: none
=====Discovery Log Entry 2===== trtype: tcp
adrfam: ipv4
subtype: current discovery subsystem treq: not specified
portid: 2
trsvcid: 8009 subnqn: nqn.1992-
08.com.netapp:sn.48391d66c0a611ecaaa5d039ea165514:discovery traddr:
192.168.2.116
eflags: explicit discovery connections, duplicate discovery information
sectype: none
=====Discovery Log Entry 3===== trtype: tcp
adrfam: ipv4
subtype: current discovery subsystem treq: not specified
portid: 3
trsvcid: 8009 subnqn: nqn.1992-
08.com.netapp:sn.48391d66c0a611ecaaa5d039ea165514:discovery traddr:
192.168.1.116
eflags: explicit discovery connections, duplicate discovery information
sectype: none
=====Discovery Log Entry 4===== trtype: tcp
```

```

adrfam: ipv4
subtype: nvme subsystem treq: not specified portid: 0
trsvcid: 4420 subnqn: nqn.1992-
08.com.netapp:sn.48391d66c0a611ecaaa5d039ea165514:subsystem.subsys_CLIENT16
traddr: 192.168.2.117 eflags: not specified sectype: none
=====Discovery Log Entry 5===== trtype: tcp
adrfam: ipv4
subtype: nvme subsystem treq: not specified portid: 1
trsvcid: 4420 subnqn: nqn.1992-
08.com.netapp:sn.48391d66c0a611ecaaa5d039ea165514:subsystem.subsys_CLIENT16
traddr: 192.168.1.117 eflags: not specified sectype: none
=====Discovery Log Entry 6===== trtype: tcp
adrfam: ipv4
subtype: nvme subsystem treq: not specified portid: 2
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.48391d66c0a611ecaaa5d039ea165514:subsystem.subsys_CLIENT16
traddr: 192.168.2.116 eflags: not specified sectype: none
=====Discovery Log Entry 7===== trtype: tcp
adrfam: ipv4
subtype: nvme subsystem treq: not specified portid: 3
trsvcid: 4420 subnqn: nqn.1992-
08.com.netapp:sn.48391d66c0a611ecaaa5d039ea165514:subsystem.subsys_CLIENT16
traddr: 192.168.1.116 eflags: not specified sectype: none

```

2. NVMe/TCPイニシエータとターゲットLIFの他のすべての組み合わせで、検出口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>
```

出力例：

```
# nvme discover -t tcp -w 192.168.1.4 -a 192.168.1.32
# nvme discover -t tcp -w 192.168.2.5 -a 192.168.2.36
# nvme discover -t tcp -w 192.168.2.5 -a 192.168.2.37
```

3. 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 -l  
<ctrl_loss_timeout_in_seconds>
```

出力例：

```
# nvme connect-all -t tcp -w 192.168.1.4 -a 192.168.1.31 -l -1  
# nvme connect-all -t tcp -w 192.168.1.4 -a 192.168.1.32 -l -1  
# nvme connect-all -t tcp -w 192.168.2.5 -a 192.168.1.36 -l -1  
# nvme connect-all -t tcp -w 192.168.2.5 -a 192.168.1.37 -l -1
```



NetAppでは、`ctrl-loss-tmo` オプションをに設定します `-1` これにより、パスが失われた場合にNVMe/TCPイニシエータが無期限に再接続を試行できるようになります。

NVMe-oF を検証します

NVMe-oFの検証には、次の手順を使用できます。

手順

1. カーネル内の 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  
Y
```

2. ホストのコントローラモデルがONTAP NVMeネームスペースに対応していることを確認します。

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

出力例：

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

3. それぞれのONTAP NVMe I/OコントローラのNVMe I/Oポリシーを確認します。

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

出力例：

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

4. ONTAPネームスペースがホストから認識されることを確認します。

```
nvme list -v
```

出力例：

```
Subsystem      Subsystem-NQN
Controllers
-----
-----  

-----  

nvme-subsy0    nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.unidir_dhcha
p  nvme0, nvme1, nvme2, nvme3
```

Device	SN	MN	
FR	TxPort Asdress	Subsystem	Namespaces
nvme0	81LGgBUqsI3AAAAAAAAE	NetApp ONTAP Controller	FFFFFFFFFF tcp
	traddr=192.168.2.214, trsvcid=4420, host_traddr=192.168.2.14	nvme-subsy0	
nvme0n1			
nvme1	81LGgBUqsI3AAAAAAAAE	NetApp ONTAP Controller	FFFFFFFFFF tcp
	traddr=192.168.2.215, trsvcid=4420, host_traddr=192.168.2.14	nvme-subsy0	
nvme0n1			
nvme2	81LGgBUqsI3AAAAAAAAE	NetApp ONTAP Controller	FFFFFFFFFF tcp
	traddr=192.168.1.214, trsvcid=4420, host_traddr=192.168.1.14	nvme-subsy0	
nvme0n1			
nvme3	81LGgBUqsI3AAAAAAAAE	NetApp ONTAP Controller	FFFFFFFFFF tcp
	traddr=192.168.1.215, trsvcid=4420, host_traddr=192.168.1.14	nvme-subsy0	
nvme0n1			

Device	Generic	NSID	Usage	Format
Controllers				
/dev/nvme0n1	/dev/ng0n1	0x1	1.07 GB / 1.07 GB	4 KiB + 0 B
nvme0, nvme1, nvme2, nvme3				

- 各バ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-subsy /dev/<subsystem_name>
```

NVMe/FC

出力例

```
# nvme list-subsy /dev/nvme1n1
nvme-subsy1 - NQN=nqn.1992-
08.com.netapp:sn.04ba0732530911ea8e8300a098dfdd91:subsystem.nvme_145
  _1
  \
+- nvme2 fc traddr=nn-0x208100a098dfdd91:pn-
  0x208200a098dfdd91,host_traddr=nn-0x200000109b579d5f:pn-
  0x100000109b579d5f live optimized
+- nvme3 fc traddr=nn-0x208100a098dfdd91:pn-
  0x208500a098dfdd91,host_traddr=nn-0x200000109b579d5e:pn-
  0x100000109b579d5e live optimized
+- nvme4 fc traddr=nn-0x208100a098dfdd91:pn-
  0x208400a098dfdd91,host_traddr=nn-0x200000109b579d5e:pn-
  0x100000109b579d5e live non-optimized
+- nvme6 fc traddr=nn-0x208100a098dfdd91:pn-
  0x208300a098dfdd91,host_traddr=nn-0x200000109b579d5f:pn-
  0x100000109b579d5f live non-optimized
```

NVMe/FC

出力例

```
# nvme list-subsy
nvme-subsy0 - NQN=nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.unidir_d
hchap
hostnqn=nqn.2014-08.org.nvmexpress:uuid:e58eca24-faff-11ea-8fee-
3a68dd3b5c5f
iopolicy=round-robin

  +- nvme0 tcp
  traddr=192.168.2.214,trsvcid=4420,host_traddr=192.168.2.14 live
    +- nvme1 tcp
    traddr=192.168.2.215,trsvcid=4420,host_traddr=192.168.2.14 live
      +- nvme2 tcp
      traddr=192.168.1.214,trsvcid=4420,host_traddr=192.168.1.14 live
        +- nvme3 tcp
        traddr=192.168.1.215,trsvcid=4420,host_traddr=192.168.1.14 live
```

6. 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
```

出力例：

Device	Vserver	Namespace Path	Size
NSID	UUID		
-----	-----	-----	-----
/dev/nvme0n1	vs_CLIENT114		
/vol/CLIENT114_vol_0_10/CLIENT114_ns10		1	c6586535-da8a-
40fa-8c20-759ea0d69d33	1.07GB		

JSON

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

出力例：

```
{
  "ONTAPdevices": [
    {
      "Device": "/dev/nvme0n1",
      "Vserver": "vs_CLIENT114",
      "Namespace_Path": "/vol/CLIENT114_vol_0_10/CLIENT114_ns10",
      "NSID": 1,
      "UUID": "c6586535-da8a-40fa-8c20-759ea0d69d33",
      "Size": "1.07GB",
      "LBA_Data_Size": 4096,
      "Namespace_Size": 262144
    }
  ]
}
```

永続的検出コントローラの作成

ONTAP 9.11.1以降では、SUSE Linux Enterprise Server 15 SP5ホスト用の永続的検出コントローラ（PDC）を作成できます。NVMeサブシステムの追加または削除のシナリオと検出口ログページのデータに対する変更を自動的に検出するには、PDCが必要です。

手順

1. 検出口ログページのデータが使用可能で、イニシエータポートとターゲットLIFの組み合わせから取得できることを確認します。

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

出力例を示します。

```
Discovery Log Number of Records 16, Generation counter 14
=====Discovery Log Entry 0=====
trtype: tcp
adrifam: ipv4
subtype: current discovery subsystem
treq: not specified
portid: 0
trsvcid: 8009
subnqn: nqn.1992-
08.com.netapp:sn.0501daf15dd11eeab68d039eaa7a232:discovery
traddr: 192.168.1.214
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: 0
trsvcid: 8009
subnqn: nqn.1992-
08.com.netapp:sn.0501daf15dd11eeab68d039eaa7a232:discovery
traddr: 192.168.1.215
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: 0
trsvcid: 8009
subnqn: nqn.1992-
08.com.netapp:sn.0501daf15dd11eeab68d039eaa7a232:discovery
traddr: 192.168.2.215
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: 0
trsvcid: 8009
subnqn: nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:discovery
traddr: 192.168.2.214
eflags: explicit discovery connections, duplicate discovery
information sectype: none
=====Discovery Log Entry 4=====
trtype: tcp
adrifam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 0
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.unidir_n
one
traddr: 192.168.1.214
eflags: none
sectype: none
=====Discovery Log Entry 5=====
trtype: tcp
adrifam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 0
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.unidir_n
one
traddr: 192.168.1.215
eflags: none
sectype: none
=====Discovery Log Entry 6=====
trtype: tcp
adrifam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 0
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.unidir_n
one
traddr: 192.168.2.215
eflags: none
sectype: none
```

```
=====Discovery Log Entry 7=====  
trtype: tcp  
adrifam: ipv4  
subtype: nvme subsystem  
treq: not specified  
portid: 0  
trsvcid: 4420  
subnqn: nqn.1992-  
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.unidir_n  
one  
traddr: 192.168.2.214  
eflags: none  
sectype: none  
=====Discovery Log Entry 8=====  
trtype: tcp  
adrifam: ipv4  
subtype: nvme subsystem  
treq: not specified  
portid: 0  
trsvcid: 4420  
subnqn: nqn.1992-  
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.subsys_C  
LIENT114  
traddr: 192.168.1.214  
eflags: none  
sectype: none  
=====Discovery Log Entry 9=====  
trtype: tcp  
adrifam: ipv4  
subtype: nvme subsystem  
treq: not specified  
portid: 0  
trsvcid: 4420  
subnqn: nqn.1992-  
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.subsys_C  
LIENT114  
traddr: 192.168.1.215  
eflags: none  
sectype: none  
=====Discovery Log Entry 10=====  
trtype: tcp  
adrifam: ipv4  
subtype: nvme subsystem  
treq: not specified  
portid: 0  
trsvcid: 4420
```

```
subnqn: nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.subsys_C
LIENT114
traddr: 192.168.2.215
eflags: none
sectype: none
=====Discovery Log Entry 11=====
trtype: tcp
adrifam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 0
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.subsys_C
LIENT114
traddr: 192.168.2.214
eflags: none
sectype: none
=====Discovery Log Entry 12=====
trtype: tcp
adrifam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 0
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.unidir_d
hchap
traddr: 192.168.1.214
eflags: none
sectype: none
=====Discovery Log Entry 13=====
trtype: tcp
adrifam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 0
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.unidir_d
hchap
traddr: 192.168.1.215
eflags: none
sectype: none
=====Discovery Log Entry 14=====
```

```
trtype: tcp
adrifam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 0
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.unidir_d
hchap
traddr: 192.168.2.215
eflags: none
sectype: none
=====Discovery Log Entry 15=====
trtype: tcp
adrifam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 0
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.unidir_d
hchap
traddr: 192.168.2.214
eflags: none
sectype: none
```

2. 検出サブシステムのPDCを作成します。

```
nvme discover -t <trtype> -w <host-traddr> -a <traddr> -p
```

出力例：

```
nvme discover -t tcp -w 192.168.1.16 -a 192.168.1.116 -p
```

3. ONTAPコントローラから、PDCが作成されたことを確認します。

```
vserver nvme show-discovery-controller -instance -vserver vserver_name
```

出力例：

```
vserver nvme show-discovery-controller -instance -vserver vs_nvme175
Vserver Name: vs_CLIENT116 Controller ID: 00C0h
Discovery Subsystem NQN: nqn.1992-
08.com.netapp:sn.48391d66c0a611ecaaa5d039ea165514:discovery Logical
Interface UUID: d23ccb0a-c0a6-11ec-9731-d039ea165abc Logical Interface:
CLIENT116_lif_4a_1
Node: A400-14-124
Host NQN: nqn.2014-08.org.nvmexpress:uuid:12372496-59c4-4d1b-be09-
74362c0clafc
Transport Protocol: nvme-tcp
Initiator Transport Address: 192.168.1.16
Host Identifier: 59de25be738348f08a79df4bce9573f3 Admin Queue Depth: 32
Header Digest Enabled: false Data Digest Enabled: false
Vserver UUID: 48391d66-c0a6-11ec-aaa5-d039ea165514
```

セキュアなインバンド認証のセットアップ

ONTAP 9 12.1以降では、SUSE Linux Enterprise Server 15 SP5ホストとONTAPコントローラの間で、NVMe/TCPおよびNVMe/FC経由でセキュアなインバンド認証がサポートされます。

セキュアな認証を設定するには、各ホストまたはコントローラを DH-HMAC-CHAP キー。NVMeホストまたはコントローラのNQNと管理者が設定した認証シークレットを組み合わせたものです。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。

CLIまたは設定JSONファイルを使用して、セキュアなインバンド認証を設定できます。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

手順

1. ホストNQNを取得します。

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

2. SUSE Linux Enterprise Server 15 SP5ホストのDHCHAPキーを生成します。

```
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:d3ca725a-ac8d-4d88-b46a-174ac235139b  
DHHC-  
1:03:J2UJQfj9f0pLnpF/ASDJRTyILKJRr5CougGpGdQSysPrLu6RW1fGl5VSjbeDF1n  
1DEh3nVBe19nQ/LxreSBeH/bx/pU=:
```

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
```

単方向設定の出力例：

```
# cat /sys/class/nvme-subsystem/nvme-subsy1/nvme*/dhchap_secret
DHHC-
1:03:je1nQCmjJLUKD62mpYbz1puw0OIws86NB96uNO/t3jbvhp7fjyR9bIRjOHg8
wQte1JCFSMkBQH3pTKGdYR1OV9gx00=:
DHHC-
1:03:je1nQCmjJLUKD62mpYbz1puw0OIws86NB96uNO/t3jbvhp7fjyR9bIRjOHg8
wQte1JCFSMkBQH3pTKGdYR1OV9gx00=:
DHHC-
1:03:je1nQCmjJLUKD62mpYbz1puw0OIws86NB96uNO/t3jbvhp7fjyR9bIRjOHg8
wQte1JCFSMkBQH3pTKGdYR1OV9gx00=:
DHHC-
```

- 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
```

双方向コンフィグレーションの出力例：

```
# cat /sys/class/nvme-subsystem/nvme-
subsys6/nvme*/dhchap_ctrl_secret
DHHC-
1:03:WorVEV83eY053kV4Ie150pphbX5LApH03F8fgH3913t1rkSGDBJTt3crXeTUB8f
CwGbPsEyz6CXxdQJi6kbn4IzmkFU=:
DHHC-
1:03:WorVEV83eY053kV4Ie150pphbX5LApH03F8fgH3913t1rkSGDBJTt3crXeTUB8f
CwGbPsEyz6CXxdQJi6kbn4IzmkFU=:
DHHC-
1:03:WorVEV83eY053kV4Ie150pphbX5LApH03F8fgH3913t1rkSGDBJTt3crXeTUB8f
CwGbPsEyz6CXxdQJi6kbn4IzmkFU=:
DHHC-
1:03:WorVEV83eY053kV4Ie150pphbX5LApH03F8fgH3913t1rkSGDBJTt3crXeTUB8f
CwGbPsEyz6CXxdQJi6kbn4IzmkFU=:
```

JSON ファイル

を使用できます /etc/nvme/config.json を含むファイル nvme connect-all ONTAPコントローラ構成で複数のNVMeサブシステムを使用できる場合のコマンド。

JSONファイルは次のコマンドを使用して生成できます。 -o オプション構文オプションの詳細については、nvme connect-allのマニュアルページを参照してください。

手順

1. JSON ファイルを設定します。

```
# cat /etc/nvme/config.json
[
  {
    "hostnqn": "nqn.2014-08.org.nvmexpress:uuid:12372496-59c4-4d1b-
be09-74362c0clafc",
    "hostid": "3ae10b42-21af-48ce-a40b-cfb5bad81839",
    "dhchap_key": "DHHC-
1:03:Cu3ZZfIz1WMlqZFnCMqpAgn/T6EV0cIFHez215U+Pow8jTgBF2UbNk3DK4wfk2E
ptWpnalrpwG5CndpOgxpRxh9m41w=",
  },
  {
    "hostnqn": "nqn.2014-08.org.nvmexpress:uuid:12372496-59c4-4d1b-
be09-74362c0clafc",
    "subsystems": [
      {
        "nqn": "nqn.1992-
08.com.netapp:sn.48391d66c0a611ecaaa5d039ea165514:subsystem.subsys_C
LIENT116",
        "ports": [

```

```

{
    "transport": "tcp",
    "traddr": "192.168.1.117",
    "host_traddr": "192.168.1.16",
    "trsvcid": "4420",
    "dhchap_ctrl_key": "DHHC-
1:01:0h58bcT/uu0rCpGsDYU6ZHvRuVqsYKuBRS0Nu0VPx5HEwaZ:"
},
{
    "transport": "tcp",
    "traddr": "192.168.1.116",
    "host_traddr": "192.168.1.16",
    "trsvcid": "4420",
    "dhchap_ctrl_key": "DHHC-
1:01:0h58bcT/uu0rCpGsDYU6ZHvRuVqsYKuBRS0Nu0VPx5HEwaZ:"
},
{
    "transport": "tcp",
    "traddr": "192.168.2.117",
    "host_traddr": "192.168.2.16",
    "trsvcid": "4420",
    "dhchap_ctrl_key": "DHHC-
1:01:0h58bcT/uu0rCpGsDYU6ZHvRuVqsYKuBRS0Nu0VPx5HEwaZ:"
},
{
    "transport": "tcp",
    "traddr": "192.168.2.116",
    "host_traddr": "192.168.2.16",
    "trsvcid": "4420",
    "dhchap_ctrl_key": "DHHC-
1:01:0h58bcT/uu0rCpGsDYU6ZHvRuVqsYKuBRS0Nu0VPx5HEwaZ:"
}
]
}
]
}
]

[NOTE]

```

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

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
```

出力例：

```
traddr=192.168.2.116 is already connected  
traddr=192.168.1.116 is already connected  
traddr=192.168.2.117 is already connected  
traddr=192.168.1.117 is already connected  
traddr=192.168.2.117 is already connected  
traddr=192.168.1.117 is already connected  
traddr=192.168.2.116 is already connected  
traddr=192.168.1.116 is already connected  
traddr=192.168.2.116 is already connected  
traddr=192.168.1.116 is already connected  
traddr=192.168.2.117 is already connected  
traddr=192.168.1.117 is already connected
```

3. 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
```

出力例：

```
DHHC-1:01:NunEWY7AZlXqxITGheByarwZdQvU4ebZg9HOjIr6nOHEkxJg:
```

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
```

出力例：

```
DHHC-  
1:03:2YJinsxa2v3+m8qqCiTnmgBZoH6mIT6G/6f0aGO8viVZB4VLNLH4z8CvK7pV  
YxN6S5fOAtaU3DNi12rieRMfdbg3704=:
```

既知の問題

SUSE Linux Enterprise Server 15 SP5 with ONTAPリリースには既知の問題はありません。

SUSE Linux Enterprise Server 15 SP4 with ONTAP向けのNVMe-oFホスト構成

NVMe over Fibre Channel (NVMe/FC) やその他のトランスポートを含むNVMe over Fabrics (NVMe-oF) は、非対称ネームスペースアクセス (ANA) を備えたSUSE Linux Enterprise Server (SLES) 15 SP4でサポートされます。NVMe-oF環境では、ANAはiSCSIおよびFCP環境のALUAマルチパスに相当し、カーネル内NVMeマルチパスで実装されます。

SUSE Linux Enterprise Server 15 SP4 with ONTAPのNVMe-oFホスト構成では、次のサポートが利用できます。

- NVMeとSCSIの両方のトラフィックを同じホストで実行できます。そのため、SCSI LUNの場合はSCSI multipathデバイスにdm-multipathを設定できますが、NVMeマルチパスを使用してホスト上のNVMe-oFネームスペースデバイスを設定することができます。
- NVMe/FCに加えて、NVMe over TCP (NVMe/TCP) もサポートされます。標準のNVMe-CLI/パッケージに含まれるNetAppプラグインには、NVMe/FCとNVMe/TCP両方のネームスペースのONTAPの詳細が表示されます。

サポートされる構成の詳細については、を参照してください["Interoperability Matrix Tool"](#)。

の機能

- NVMeセキュア、インバンド認証のサポート
- 一意の検出NQNを使用した永続的検出コントローラ (PDC) のサポート

既知の制限

- 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。
- NVMe-oFはsanlunではサポートされていません。そのため、SUSE Linux Enterprise Server 15 SP5ホストのNVMe-oFではホストユーティリティのサポートを利用できません。すべてのNVMe-oF転送では、標準のNVMe-CLIパッケージに含まれているNetAppプラグインを利用できます。

NVMe/FC を設定

NVMe/FCは、Broadcom/Emulex FCアダプタまたはMarvell/Qlogic FCアダプタに設定できます。

Broadcom / Emulex

手順

- 推奨されるアダプタモデルを使用していることを確認します。

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

出力例：

```
LPe32002 M2  
LPe32002-M2
```

- アダプタモデル概要を確認します。

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

出力例：

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

- 推奨されるバージョンのEmulex Host Bus Adapter (HBA；ホストバスアダプタ) ファームウェアを使用していることを確認します。

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

出力例：

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

- 推奨バージョンのlpfcドライバを使用していることを確認します。

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

出力例：

```
0:14.2.0.6
```

5. 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
```

出力例：

```
0x100000109b579d5e  
0x100000109b579d5f
```

6. 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
```

出力例：

```
Online  
Online
```

7. 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
```

出力例：

次の例では、1つのイニシエータポートが有効になっており、2つのターゲットLIFで接続されています。

```

NVME Initiator Enabled
XRI Dist lpfc0 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc0 WWPN x100000109b579d5e WWNN x200000109b579d5e DID
x011c00 ONLINE
NVME RPORT WWPN x208400a098dfdd91 WWNN x208100a098dfdd91 DID x011503
TARGET DISCSRVC ONLINE
NVME RPORT WWPN x208500a098dfdd91 WWNN x208100a098dfdd91 DID x010003
TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 0000000e49 Cmpl 0000000e49 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 00000003ceb594f Issue 000000003ce65dbe OutIO
ffffffffffffb046f
abort 00000bd2 noxri 00000000 nondlp 00000000 qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 000014f4 Err 00012abd

NVME Initiator Enabled
XRI Dist lpfc1 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc1 WWPN x100000109b579d5f WWNN x200000109b579d5f DID
x011b00 ONLINE
NVME RPORT WWPN x208300a098dfdd91 WWNN x208100a098dfdd91 DID x010c03
TARGET DISCSRVC ONLINE
NVME RPORT WWPN x208200a098dfdd91 WWNN x208100a098dfdd91 DID x012a03
TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 0000000e50 Cmpl 0000000e50 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 00000003c9859ca Issue 000000003c93515e OutIO
fffffffffffffaf794
abort 00000b73 noxri 00000000 nondlp 00000000 qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 0000159d Err 000135c3

```

8. ホストをリブートします。

Marvell/QLogic

SUSE Linux Enterprise Server 15 SP4カーネルに含まれているネイティブの受信トレイqla2xxxドライバには、最新の修正が含まれています。これらの修正は、ONTAPのサポートに不可欠です。

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
```

出力例：

```
QLE2742 FW:v9.08.02 DVR:v10.02.07.800-k QLE2742 FW:v9.08.02  
DVR:v10.02.07.800-k
```

2. を確認します `ql2xnvmeenable` パラメータは1に設定されています。

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

1MB の I/O サイズを有効にする（オプション）

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
```

NVMeサービスを有効にする

には2つのNVMe/FCブートサービスが含まれています。`nvme-cli` パッケージ (*only*) `nvmefc-boot-connections.service` システム起動中に起動できるようになっています。`nvmf-`

`autoconnect.service` が有効になつていません。そのため、手動で有効にする必要があります。`nvmf-autoconnect.service` システム起動中に起動します。

手順

1. 有効にします `nvmf-autoconnect.service` :

```
# systemctl enable nvmf-autoconnect.service
Created symlink /etc/systemd/system/default.target.wants/nvmf-
autoconnect.service → /usr/lib/systemd/system/nvmf-autoconnect.service.
```

2. ホストをリブートします。
3. システムのブート後に `nvmefc-boot-connections.service` が実行されていることを確認し `nvmf-autoconnect.service` ます。

出力例：

```

# systemctl status nvmf-autoconnect.service
nvmf-autoconnect.service - Connect NVMe-oF subsystems automatically
during boot
   Loaded: loaded (/usr/lib/systemd/system/nvmf-autoconnect.service;
  enabled; vendor preset: disabled)
     Active: inactive (dead) since Thu 2023-05-25 14:55:00 IST; 11min
    ago
       Process: 2108 ExecStartPre=/sbin/modprobe nvme-fabrics (code=exited,
      status=0/SUCCESS)
       Process: 2114 ExecStart=/usr/sbin/nvme connect-all (code=exited,
      status=0/SUCCESS)
     Main PID: 2114 (code=exited, status=0/SUCCESS)

systemd[1]: Starting Connect NVMe-oF subsystems automatically during
boot...
nvme[2114]: traddr=nn-0x201700a098fd4ca6:pn-0x201800a098fd4ca6 is
already connected
systemd[1]: nvmf-autoconnect.service: Deactivated successfully.
systemd[1]: Finished Connect NVMe-oF subsystems automatically during
boot.

# systemctl status nvmefc-boot-connections.service
nvmefc-boot-connections.service - Auto-connect to subsystems on FC-NVME
devices found during boot
   Loaded: loaded (/usr/lib/systemd/system/nvmefc-boot-
connections.service; enabled; vendor preset: enabled)
     Active: inactive (dead) since Thu 2023-05-25 14:55:00 IST; 11min ago
    Main PID: 1647 (code=exited, status=0/SUCCESS)

systemd[1]: Starting Auto-connect to subsystems on FC-NVME devices found
during boot...
systemd[1]: nvmefc-boot-connections.service: Succeeded.
systemd[1]: Finished Auto-connect to subsystems on FC-NVME devices found
during boot.

```

NVMe/FC を設定

NVMe/TCPの設定には、次の手順を使用できます。

手順

1. イニシエータポートがサポートされているNVMe/TCP LIFの検出ログページの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>
```

出力例：

```
# nvme discover -t tcp -w 192.168.1.4 -a 192.168.1.31

Discovery Log Number of Records 8, Generation counter 18
=====Discovery Log Entry 0===== trtype: tcp
adrfam: ipv4
subtype: current discovery subsystem treq: not specified
portid: 0
trsvcid: 8009 subnqn: nqn.1992-
08.com.netapp:sn.48391d66c0a611ecaaa5d039ea165514:discovery traddr:
192.168.2.117
eflags: explicit discovery connections, duplicate discovery information
sectype: none
=====Discovery Log Entry 1===== trtype: tcp
adrfam: ipv4
subtype: current discovery subsystem treq: not specified
portid: 1
trsvcid: 8009 subnqn: nqn.1992-
08.com.netapp:sn.48391d66c0a611ecaaa5d039ea165514:discovery traddr:
192.168.1.117
eflags: explicit discovery connections, duplicate discovery information
sectype: none
=====Discovery Log Entry 2===== trtype: tcp
adrfam: ipv4
subtype: current discovery subsystem treq: not specified
portid: 2
trsvcid: 8009 subnqn: nqn.1992-
08.com.netapp:sn.48391d66c0a611ecaaa5d039ea165514:discovery traddr:
192.168.2.116
eflags: explicit discovery connections, duplicate discovery information
sectype: none
=====Discovery Log Entry 3===== trtype: tcp
adrfam: ipv4
subtype: current discovery subsystem treq: not specified
portid: 3
trsvcid: 8009 subnqn: nqn.1992-
08.com.netapp:sn.48391d66c0a611ecaaa5d039ea165514:discovery traddr:
192.168.1.116
eflags: explicit discovery connections, duplicate discovery information
sectype: none
=====Discovery Log Entry 4===== trtype: tcp
```

```

adrfam: ipv4
subtype: nvme subsystem treq: not specified portid: 0
trsvcid: 4420 subnqn: nqn.1992-
08.com.netapp:sn.48391d66c0a611ecaaa5d039ea165514:subsystem.subsys_CLIENT16
traddr: 192.168.2.117 eflags: not specified sectype: none
=====Discovery Log Entry 5===== trtype: tcp
adrfam: ipv4
subtype: nvme subsystem treq: not specified portid: 1
trsvcid: 4420 subnqn: nqn.1992-
08.com.netapp:sn.48391d66c0a611ecaaa5d039ea165514:subsystem.subsys_CLIENT16
traddr: 192.168.1.117 eflags: not specified sectype: none
=====Discovery Log Entry 6===== trtype: tcp
adrfam: ipv4
subtype: nvme subsystem treq: not specified portid: 2
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.48391d66c0a611ecaaa5d039ea165514:subsystem.subsys_CLIENT16
traddr: 192.168.2.116 eflags: not specified sectype: none
=====Discovery Log Entry 7===== trtype: tcp
adrfam: ipv4
subtype: nvme subsystem treq: not specified portid: 3
trsvcid: 4420 subnqn: nqn.1992-
08.com.netapp:sn.48391d66c0a611ecaaa5d039ea165514:subsystem.subsys_CLIENT16
traddr: 192.168.1.116 eflags: not specified sectype: none

```

2. NVMe/TCPイニシエータとターゲットLIFの他のすべての組み合わせで、検出口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>
```

出力例：

```
# nvme discover -t tcp -w 192.168.1.4 -a 192.168.1.32
# nvme discover -t tcp -w 192.168.2.5 -a 192.168.2.36
# nvme discover -t tcp -w 192.168.2.5 -a 192.168.2.37
```

3. 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 -l  
<ctrl_loss_timeout_in_seconds>
```

出力例：

```
# nvme connect-all -t tcp -w 192.168.1.4 -a 192.168.1.31 -l -1  
# nvme connect-all -t tcp -w 192.168.1.4 -a 192.168.1.32 -l -1  
# nvme connect-all -t tcp -w 192.168.2.5 -a 192.168.1.36 -l -1  
# nvme connect-all -t tcp -w 192.168.2.5 -a 192.168.1.37 -l -1
```



NetAppでは、`ctrl-loss-tmo` オプションをに設定します `-1` これにより、パスが失われた場合にNVMe/TCPイニシエータが無期限に再接続を試行できるようになります。

NVMe-oF を検証します

NVMe-oFの検証には、次の手順を使用できます。

手順

1. カーネル内の 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  
Y
```

2. ホストのコントローラモデルがONTAP NVMeネームスペースに対応していることを確認します。

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

出力例：

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

3. それぞれのONTAP NVMe I/OコントローラのNVMe I/Oポリシーを確認します。

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

出力例：

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

4. ONTAPネームスペースがホストから認識されることを確認します。

```
nvme list -v
```

出力例：

```
Subsystem      Subsystem-NQN
Controllers
-----
-----  

-----  

nvme-subsy0    nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.unidir_dhcha
p    nvme0, nvme1, nvme2, nvme3
```

Device	SN	MN	
FR	TxPort Asdress	Subsystem	Namespaces
nvme0	81LGgBUqsI3AAAAAAAAE	NetApp ONTAP Controller	FFFFFFFFFF tcp
	traddr=192.168.2.214, trsvcid=4420, host_traddr=192.168.2.14	nvme-subsy0	
nvme0n1			
nvme1	81LGgBUqsI3AAAAAAAAE	NetApp ONTAP Controller	FFFFFFFFFF tcp
	traddr=192.168.2.215, trsvcid=4420, host_traddr=192.168.2.14	nvme-subsy0	
nvme0n1			
nvme2	81LGgBUqsI3AAAAAAAAE	NetApp ONTAP Controller	FFFFFFFFFF tcp
	traddr=192.168.1.214, trsvcid=4420, host_traddr=192.168.1.14	nvme-subsy0	
nvme0n1			
nvme3	81LGgBUqsI3AAAAAAAAE	NetApp ONTAP Controller	FFFFFFFFFF tcp
	traddr=192.168.1.215, trsvcid=4420, host_traddr=192.168.1.14	nvme-subsy0	
nvme0n1			

Device	Generic	NSID	Usage	Format
Controllers				
/dev/nvme0n1	/dev/ng0n1	0x1	1.07 GB / 1.07 GB	4 KiB + 0 B
nvme0, nvme1, nvme2, nvme3				

5. 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-subsy /dev/<subsystem_name>
```

NVMe/FC

```
# nvme list-subsy /dev/nvme1n1
nvme-subsy - NQN=nqn.1992-
08.com.netapp:sn.04ba0732530911ea8e8300a098dfdd91:subsystem.nvme_145
  _1
  \
+- nvme2 fc traddr=nn-0x208100a098dfdd91:pn-
  0x208200a098dfdd91,host_traddr=nn-0x200000109b579d5f:pn-
  0x100000109b579d5f live optimized
+- nvme3 fc traddr=nn-0x208100a098dfdd91:pn-
  0x208500a098dfdd91,host_traddr=nn-0x200000109b579d5e:pn-
  0x100000109b579d5e live optimized
+- nvme4 fc traddr=nn-0x208100a098dfdd91:pn-
  0x208400a098dfdd91,host_traddr=nn-0x200000109b579d5e:pn-
  0x100000109b579d5e live non-optimized
+- nvme6 fc traddr=nn-0x208100a098dfdd91:pn-
  0x208300a098dfdd91,host_traddr=nn-0x200000109b579d5f:pn-
  0x100000109b579d5f live non-optimized
```

NVMe/FC

```
# nvme list-subsy
nvme-subsy0 - NQN=nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.unidir_d
hchap
hostnqn=nqn.2014-08.org.nvmexpress:uuid:e58eca24-faff-11ea-8fee-
3a68dd3b5c5f
iopolicy=round-robin

  +- nvme0 tcp
  traddr=192.168.2.214,trsvcid=4420,host_traddr=192.168.2.14 live
    +- nvme1 tcp
    traddr=192.168.2.215,trsvcid=4420,host_traddr=192.168.2.14 live
      +- nvme2 tcp
      traddr=192.168.1.214,trsvcid=4420,host_traddr=192.168.1.14 live
        +- nvme3 tcp
        traddr=192.168.1.215,trsvcid=4420,host_traddr=192.168.1.14 live
```

6. 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
```

出力例：

Device NSID UUID	Vserver	Namespace Path	Size
/dev/nvme0n1	vs_CLIENT114		
/vol/CLIENT114_vol_0_10/CLIENT114_ns10		1	c6586535-da8a-
40fa-8c20-759ea0d69d33	1.07GB		

JSON

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

出力例：

```
{  
    "ONTAPdevices": [  
        {  
            "Device": "/dev/nvme0n1",  
            "Vserver": "vs_CLIENT114",  
            "Namespace_Path": "/vol/CLIENT114_vol_0_10/CLIENT114_ns10",  
            "NSID": 1,  
            "UUID": "c6586535-da8a-40fa-8c20-759ea0d69d33",  
            "Size": "1.07GB",  
            "LBA_Data_Size": 4096,  
            "Namespace_Size": 262144  
        }  
    ]  
}
```

永続的検出コントローラの作成

ONTAP 9.11.1以降では、SUSE Linux Enterprise Server 15 SP4ホスト用の永続的検出コントローラ（PDC）を作成できます。NVMeサブシステムの追加または削除のシナリオと検出口ログページのデータに対する変更を自動的に検出するには、PDCが必要です。

手順

- 検出ログページのデータが使用可能で、イニシエータポートとターゲットLIFの組み合わせから取得できることを確認します。

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

出力例を示します。

```
Discovery Log Number of Records 16, Generation counter 14
=====Discovery Log Entry 0=====
trtype: tcp
adrifam: ipv4
subtype: current discovery subsystem
treq: not specified
portid: 0
trsvcid: 8009
subnqn: nqn.1992-
08.com.netapp:sn.0501daf15dd11eeab68d039eaa7a232:discovery
traddr: 192.168.1.214
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: 0
trsvcid: 8009
subnqn: nqn.1992-
08.com.netapp:sn.0501daf15dd11eeab68d039eaa7a232:discovery
traddr: 192.168.1.215
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: 0
trsvcid: 8009
subnqn: nqn.1992-
08.com.netapp:sn.0501daf15dd11eeab68d039eaa7a232:discovery
traddr: 192.168.2.215
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: 0
trsvcid: 8009
subnqn: nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:discovery
traddr: 192.168.2.214
eflags: explicit discovery connections, duplicate discovery
information sectype: none
=====Discovery Log Entry 4=====
trtype: tcp
adrifam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 0
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.unidir_n
one
traddr: 192.168.1.214
eflags: none
sectype: none
=====Discovery Log Entry 5=====
trtype: tcp
adrifam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 0
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.unidir_n
one
traddr: 192.168.1.215
eflags: none
sectype: none
=====Discovery Log Entry 6=====
trtype: tcp
adrifam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 0
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.unidir_n
one
traddr: 192.168.2.215
eflags: none
sectype: none
```

```
=====Discovery Log Entry 7=====  
trtype: tcp  
adrifam: ipv4  
subtype: nvme subsystem  
treq: not specified  
portid: 0  
trsvcid: 4420  
subnqn: nqn.1992-  
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.unidir_n  
one  
traddr: 192.168.2.214  
eflags: none  
sectype: none  
=====Discovery Log Entry 8=====  
trtype: tcp  
adrifam: ipv4  
subtype: nvme subsystem  
treq: not specified  
portid: 0  
trsvcid: 4420  
subnqn: nqn.1992-  
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.subsys_C  
LIENT114  
traddr: 192.168.1.214  
eflags: none  
sectype: none  
=====Discovery Log Entry 9=====  
trtype: tcp  
adrifam: ipv4  
subtype: nvme subsystem  
treq: not specified  
portid: 0  
trsvcid: 4420  
subnqn: nqn.1992-  
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.subsys_C  
LIENT114  
traddr: 192.168.1.215  
eflags: none  
sectype: none  
=====Discovery Log Entry 10=====  
trtype: tcp  
adrifam: ipv4  
subtype: nvme subsystem  
treq: not specified  
portid: 0  
trsvcid: 4420
```

```
subnqn: nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.subsys_C
LIENT114
traddr: 192.168.2.215
eflags: none
sectype: none
=====Discovery Log Entry 11=====
trtype: tcp
adrifam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 0
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.subsys_C
LIENT114
traddr: 192.168.2.214
eflags: none
sectype: none
=====Discovery Log Entry 12=====
trtype: tcp
adrifam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 0
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.unidir_d
hchap
traddr: 192.168.1.214
eflags: none
sectype: none
=====Discovery Log Entry 13=====
trtype: tcp
adrifam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 0
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.unidir_d
hchap
traddr: 192.168.1.215
eflags: none
sectype: none
=====Discovery Log Entry 14=====
```

```
trtype: tcp
adrifam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 0
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.unidir_d
hchap
traddr: 192.168.2.215
eflags: none
sectype: none
=====Discovery Log Entry 15=====
trtype: tcp
adrifam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 0
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.unidir_d
hchap
traddr: 192.168.2.214
eflags: none
sectype: none
```

2. 検出サブシステムのPDCを作成します。

```
nvme discover -t <trtype> -w <host-traddr> -a <traddr> -p
```

出力例：

```
nvme discover -t tcp -w 192.168.1.16 -a 192.168.1.116 -p
```

3. ONTAPコントローラから、PDCが作成されたことを確認します。

```
vserver nvme show-discovery-controller -instance -vserver vserver_name
```

出力例：

```
vserver nvme show-discovery-controller -instance -vserver vs_nvme175
Vserver Name: vs_CLIENT116 Controller ID: 00C0h
Discovery Subsystem NQN: nqn.1992-
08.com.netapp:sn.48391d66c0a611ecaaa5d039ea165514:discovery Logical
Interface UUID: d23ccb0a-c0a6-11ec-9731-d039ea165abc Logical Interface:
CLIENT116_lif_4a_1
Node: A400-14-124
Host NQN: nqn.2014-08.org.nvmexpress:uuid:12372496-59c4-4d1b-be09-
74362c0clafc
Transport Protocol: nvme-tcp
Initiator Transport Address: 192.168.1.16
Host Identifier: 59de25be738348f08a79df4bce9573f3 Admin Queue Depth: 32
Header Digest Enabled: false Data Digest Enabled: false
Vserver UUID: 48391d66-c0a6-11ec-aaa5-d039ea165514
```

セキュアなインバンド認証のセットアップ

ONTAP 9 12.1以降では、SUSE Linux Enterprise Server 15 SP4ホストとONTAPコントローラの間で、NVMe/TCPおよびNVMe/FC経由でセキュアなインバンド認証がサポートされます。

セキュアな認証を設定するには、各ホストまたはコントローラを DH-HMAC-CHAP キー。NVMeホストまたはコントローラのNQNと管理者が設定した認証シークレットを組み合わせたものです。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。

CLIまたは設定JSONファイルを使用して、セキュアなインバンド認証を設定できます。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

手順

1. ホストNQNを取得します。

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

2. SUSE Linux Enterprise Server 15 SP4ホストのDHCHAPキーを生成します。

```
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.nvmeexpress:uuid:d3ca725a-ac8d-4d88-b46a-174ac235139b
DHCH-
1:03:J2UJQfj9f0pLnpF/ASDJRTyILKJRr5CougGpGdQSsPrLu6RW1fG15VSjbeDF1n1DE
h3nVBe19nQ/LxreSBeH/bx/pU=:
```

1. 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}
```

2. 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>
```

3. 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-subsyssX>/nvme*/dhchap_secret
```

単方向設定の出力例：

```
SR650-14-114:~ # cat /sys/class/nvme-subsystem/nvme-  
subsys1/nvme*/dhchap_secret  
DHHC-  
1:03:je1nQCmjJLUKD62mpYbz1puw0OIws86NB96uNO/t3jbvhp7fjyR9bIRjOHg8  
wQtye1JCFSMkBQH3pTKGdYR1OV9gx00=:  
DHHC-  
1:03:je1nQCmjJLUKD62mpYbz1puw0OIws86NB96uNO/t3jbvhp7fjyR9bIRjOHg8  
wQtye1JCFSMkBQH3pTKGdYR1OV9gx00=:  
DHHC-  
1:03:je1nQCmjJLUKD62mpYbz1puw0OIws86NB96uNO/t3jbvhp7fjyR9bIRjOHg8  
wQtye1JCFSMkBQH3pTKGdYR1OV9gx00=:  
DHHC-  
1:03:je1nQCmjJLUKD62mpYbz1puw0OIws86NB96uNO/t3jbvhp7fjyR9bIRjOHg8  
wQtye1JCFSMkBQH3pTKGdYR1OV9gx00=:
```

- 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
```

双方向コンフィグレーションの出力例：

```
SR650-14-114:~ # cat /sys/class/nvme-subsystem/nvme-
subsys6/nvme*/dhchap_ctrl_secret
DHHC-
1:03:WorVEV83eY053kV4Ie15OpphbX5LAphO3F8fgH3913t1rkSGDBJTt3crXeTUB8f
CwGbPsEyz6CXxdQJi6kbn4IzmkFU=:
DHHC-
1:03:WorVEV83eY053kV4Ie15OpphbX5LAphO3F8fgH3913t1rkSGDBJTt3crXeTUB8f
CwGbPsEyz6CXxdQJi6kbn4IzmkFU=:
DHHC-
1:03:WorVEV83eY053kV4Ie15OpphbX5LAphO3F8fgH3913t1rkSGDBJTt3crXeTUB8f
CwGbPsEyz6CXxdQJi6kbn4IzmkFU=:
DHHC-
1:03:WorVEV83eY053kV4Ie15OpphbX5LAphO3F8fgH3913t1rkSGDBJTt3crXeTUB8f
CwGbPsEyz6CXxdQJi6kbn4IzmkFU=:
```

JSON ファイル

を使用できます /etc/nvme/config.json を含むファイル nvme connect-all ONTAPコントローラ構成で複数のNVMeサブシステムを使用できる場合のコマンド。

JSONファイルは次のコマンドを使用して生成できます。 -o オプション構文オプションの詳細については、nvme connect-allのマニュアルページを参照してください。

手順

1. JSON ファイルを設定します。

```
# cat /etc/nvme/config.json
[
  {
    "hostnqn": "nqn.2014-08.org.nvmexpress:uuid:12372496-59c4-4d1b-
be09-74362c0clafc",
    "hostid": "3ae10b42-21af-48ce-a40b-cfb5bad81839",
    "dhchap_key": "DHHC-
1:03:Cu3ZZfIz1WMlqZFnCMqpAgn/T6EV0cIFHez215U+Pow8jTgBF2UbNk3DK4wfk2E
ptWpnalrpwG5CndpOgxpRxh9m41w=",
  },
  {
    "hostnqn": "nqn.2014-08.org.nvmexpress:uuid:12372496-59c4-4d1b-
be09-74362c0clafc",
    "subsystems": [
      {
        "nqn": "nqn.1992-
08.com.netapp:sn.48391d66c0a611ecaaa5d039ea165514:subsystem.subsys_C
LIENT116",
        "ports": [

```

```

{
    "transport": "tcp",
    "traddr": "192.168.1.117",
    "host_traddr": "192.168.1.16",
    "trsvcid": "4420",
    "dhchap_ctrl_key": "DHHC-
1:01:0h58bcT/uu0rCpGsDYU6ZHvRuVqsYKuBRS0Nu0VPx5HEwaZ:"
},
{
    "transport": "tcp",
    "traddr": "192.168.1.116",
    "host_traddr": "192.168.1.16",
    "trsvcid": "4420",
    "dhchap_ctrl_key": "DHHC-
1:01:0h58bcT/uu0rCpGsDYU6ZHvRuVqsYKuBRS0Nu0VPx5HEwaZ:"
},
{
    "transport": "tcp",
    "traddr": "192.168.2.117",
    "host_traddr": "192.168.2.16",
    "trsvcid": "4420",
    "dhchap_ctrl_key": "DHHC-
1:01:0h58bcT/uu0rCpGsDYU6ZHvRuVqsYKuBRS0Nu0VPx5HEwaZ:"
},
{
    "transport": "tcp",
    "traddr": "192.168.2.116",
    "host_traddr": "192.168.2.16",
    "trsvcid": "4420",
    "dhchap_ctrl_key": "DHHC-
1:01:0h58bcT/uu0rCpGsDYU6ZHvRuVqsYKuBRS0Nu0VPx5HEwaZ:"
}
]
}
]
}
]

[NOTE]

```

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

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
```

出力例：

```
traddr=192.168.2.116 is already connected  
traddr=192.168.1.116 is already connected  
traddr=192.168.2.117 is already connected  
traddr=192.168.1.117 is already connected  
traddr=192.168.2.117 is already connected  
traddr=192.168.1.117 is already connected  
traddr=192.168.2.116 is already connected  
traddr=192.168.1.116 is already connected  
traddr=192.168.2.116 is already connected  
traddr=192.168.1.116 is already connected  
traddr=192.168.2.117 is already connected  
traddr=192.168.1.117 is already connected
```

3. 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
```

出力例：

```
DHHC-1:01:NunEWY7AZlXqxITGheByarwZdQvU4ebZg9HOjIr6nOHEkxJg:
```

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
```

出力例：

```
DHHC-  
1:03:2YJinsxa2v3+m8qqCiTnmgBZoH6mIT6G/6f0aGO8viVZB4VLNLH4z8CvK7pVYxN  
6S5foAtaU3Dni12rieRMfdbg3704=:
```

既知の問題

SUSE Linux Enterprise Server 15 SP4 with ONTAPリリースには既知の問題はありません。

ONTAP を搭載した SUSE Linux Enterprise Server 15 SP3 用 NVMe-oF ホスト構成

SUSE Linux Enterprise Server 15 SP3 with ANA (Asymmetric Namespace Access) では、NVMe over FabricsまたはNVMe-oF (NVMe/FCおよびその他のトランスポートを含む) がサポートされます。ANAはNVMe-oF環境におけるALUAに相当し、現在はカーネル内NVMeマルチパスで実装されています。この手順を使用すると、SUSE Linux Enterprise Server 15 SP3およびONTAP上のANAをターゲットとして使用して、カーネル内NVMeマルチパスでNVMe-oFを有効にすることができます。

サポートされている構成の詳細については、を参照してください["Interoperability Matrix Tool"](#)。

の機能

- SUSE Linux Enterprise Server 15 SP3は、NVMe/FCおよびその他のトランスポートをサポートしています。
- sanlun にも対応していません。そのため、SUSE Linux Enterprise Server 15 SP3では、Linux Host UtilitiesでNVMe-oFがサポートされません。NVMe-oFの標準のNVMe-CLIパッケージに含まれているNetAppプラグインを利用できます。すべてのNVMe-oF転送がサポートされている必要があります。
- NVMeとSCSIの両方のトラフィックを同じホストで実行できます。実際、これは、お客様に一般的に導入されるホスト構成であると想定されています。そのため、SCSIの場合はSCSI LUNを通常どおりに設定し`dm-multipath`でmpathデバイスを構成できますが、NVMeマルチパスを使用してホストでNVMe-oFマルチパスデバイスを構成できます。

既知の制限

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。

カーネル内の NVMe マルチパスを有効にします

SUSE Linux Enterprise Server 15 SP3などのSUSE Linux Enterprise Serverホストでは、カーネル内NVMeマルチパスがすでにデフォルトで有効になっています。そのため、ここでは追加の設定は必要ありません。サポートされている構成の詳細については、を参照してください["Interoperability Matrix Tool"](#)。

NVMe-oF イニシエータパッケージ

サポートされている構成の詳細については、を参照してください["Interoperability Matrix Tool"](#)。

1. 必要なカーネルとnvme-cli MUパッケージがSUSE Linux Enterprise Server 15 SP3 MUホストにインストールされていることを確認します。

例

```
# uname -r  
5.3.18-59.5-default  
  
# rpm -qa|grep nvme-cli  
nvme-cli-1.13-3.3.1.x86_64
```

上記の nvme-cli MU パッケージには、次のものが含まれています。

- * NVMe/FC 自動接続スクリプト * - ネームスペースへの基盤となるパスがリストアされた場合やホストのリブート中に NVMe/FC 自動接続（再接続）で必要：

```
# rpm -ql nvme-cli-1.13-3.3.1.x86_64  
/etc/nvme  
/etc/nvme/hostid  
/etc/nvme/hostnqn  
/usr/lib/systemd/system/nvmefc-boot-connections.service  
/usr/lib/systemd/system/nvmefc-connect.target  
/usr/lib/systemd/system/nvmefc-connect@.service  
...
```

- * 環境 udev ルール * - 新しい udev ルールで、 NVMe マルチパスラウンドロビンロードバランサのデフォルト ONTAP All ONTAP ネームスペースを確保する。

```
# rpm -ql nvme-cli-1.13-3.3.1.x86_64  
/etc/nvme  
/etc/nvme/hostid  
/etc/nvme/hostnqn  
/usr/lib/systemd/system/nvmefc-boot-connections.service  
/usr/lib/systemd/system/nvme-autoconnect.service  
/usr/lib/systemd/system/nvme-connect.target  
/usr/lib/systemd/system/nvme-connect@.service  
/usr/lib/udev/rules.d/70-nvme-autoconnect.rules  
/usr/lib/udev/rules.d/71-nvme-iopolicy-netapp.rules  
...  
# cat /usr/lib/udev/rules.d/71-nvme-iopolicy-netapp.rules  
# Enable round-robin for NetApp ONTAP and NetApp E-Series  
ACTION=="add", SUBSYSTEM=="nvme-subsystem", ATTR{model}=="NetApp  
ONTAP Controller", ATTR{iopolicy}="round-robin"  
ACTION=="add", SUBSYSTEM=="nvme-subsystem", ATTR{model}=="NetApp E-  
Series", ATTR{iopolicy}="round-robin"
```

- * ONTAP デバイス用ネットアッププラグイン * - ONTAP ネームスペースも処理できるように、既存のネットアッププラグインが変更されました。

2. ホストの /etc/nvme/hostnqn' で hostnqn 文字列を確認し、ONTAP アレイの対応するサブシステムの hostnqn 文字列と適切に一致していることを確認します。例：

```
# cat /etc/nvme/hostnqn
nqn.2014-08.org.nvmexpress:uuid:3ca559e1-5588-4fc4-b7d6-5ccfb0b9f054
::> vserver nvme subsystem host show -vserver vs_fcnvme_145
Vserver      Subsystem      Host NQN
-----
vs_nvme_145  nvme_145_1  nqn.2014-08.org.nvmexpress:uuid:c7b07b16-a22e-
41a6-a1fd-cf8262c8713f
          nvme_145_2  nqn.2014-08.org.nvmexpress:uuid:c7b07b16-a22e-
41a6-a1fd-cf8262c8713f
          nvme_145_3  nqn.2014-08.org.nvmexpress:uuid:c7b07b16-a22e-
41a6-a1fd-cf8262c8713f
          nvme_145_4  nqn.2014-08.org.nvmexpress:uuid:c7b07b16-a22e-
41a6-a1fd-cf8262c8713f
          nvme_145_5  nqn.2014-08.org.nvmexpress:uuid:c7b07b16-a22e-
41a6-a1fd-cf8262c8713f
5 entries were displayed.
```

ホストで使用している FC アダプタに応じて、次の手順を実行します。

NVMe/FC を設定

Broadcom / Emulex

1. 推奨されるアダプタとファームウェアのバージョンが揃っていることを確認します。例：

```
# cat /sys/class/scsi_host/host*/modelname
LPe32002-M2
LPe32002-M2
# cat /sys/class/scsi_host/host*/modeldesc
Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
# cat /sys/class/scsi_host/host*/fwrev
12.8.340.8, sli-4:2:c
12.8.840.8, sli-4:2:c
```

- 新しい lpfcc ドライバ（インボックスとアウトボックスの両方）には 'lpfc_enable_fc4_type' デフォルトが 3 に設定されていますが、'/etc/modprobe.d/lpfc.conf' でこれを明示的に設定しなくても 'initrd' を再作成できます。lpfc nvme のサポートは 'デフォルト' すでに有効になっています

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

- 既存のネイティブインボックス lpfc ドライバは、 NVMe / FC に対応した最新のバージョンです。そのため、 lpfc OOB ドライバをインストールする必要はありません。

```
# cat /sys/module/lpfc/version  
0:12.8.0.10
```

2. イニシエータポートが動作していることを確認します。

```
# cat /sys/class/fc_host/host*/port_name  
0x100000109b579d5e  
0x100000109b579d5f  
# cat /sys/class/fc_host/host*/port_state  
Online  
Online
```

3. NVMe/FCイニシエータポートが有効になっていること、ターゲットポートが表示されていること、およびすべてのポートが稼働していることを確認します。+次の例では、イニシエータポートが1つだけ有効になっており、2つのターゲットLIFで接続されています。

```

# cat /sys/class/scsi_host/host*/nvme_info
NVME Initiator Enabled
XRI Dist lpfc0 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc0 WWPN x100000109b579d5e WWNN x200000109b579d5e DID
x011c00 ONLINE
NVME RPORT WWPN x208400a098dfdd91 WWNN x208100a098dfdd91 DID x011503
TARGET DISCSRVC ONLINE
NVME RPORT WWPN x208500a098dfdd91 WWNN x208100a098dfdd91 DID x010003
TARGET DISCSRVC ONLINE
NVME Statistics
LS: Xmt 0000000e49 Cmpl 0000000e49 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 00000003ceb594f Issue 00000003ce65dbe OutIO
fffffffffffffb046f
abort 00000bd2 noxri 00000000 nondlp 00000000 qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 000014f4 Err 00012abd
NVME Initiator Enabled
XRI Dist lpfc1 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc1 WWPN x100000109b579d5f WWNN x200000109b579d5f DID
x011b00 ONLINE
NVME RPORT WWPN x208300a098dfdd91 WWNN x208100a098dfdd91 DID x010c03
TARGET DISCSRVC ONLINE
NVME RPORT WWPN x208200a098dfdd91 WWNN x208100a098dfdd91 DID x012a03
TARGET DISCSRVC ONLINE
NVME Statistics
LS: Xmt 0000000e50 Cmpl 0000000e50 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 00000003c9859ca Issue 00000003c93515e OutIO
fffffffffffffaf794
abort 00000b73 noxri 00000000 nondlp 00000000 qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 0000159d Err 000135c3

```

4. ホストをリブートします。

1MB I/O サイズを有効にする（オプション）

ONTAP は Identify コントローラデータに MDT（MAX Data 転送サイズ）8 を報告します。つまり、最大 I/O 要求サイズは最大 1 MB でなければなりません。ただし、Broadcom NVMe/FC ホストにサイズ 1MB の I/O 要求を発行するには、`lpfc` パラメータを `lpfc_sg_seg_cnt` デフォルト値の 64 から 256 まで増やす必要があります。次の手順を使用して実行します。

1. それぞれの「`modprobe lpfc.conf`」ファイルに値 256 を追加します。

```
# 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. 再起動後、対応する sysfs 値を確認して、上記の設定が適用されていることを確認します。

```
# cat /sys/module/lpfc/parameters/lpfc_sg_seg_cnt
256
```

これで、Broadcom NVMe/FC ホストは ONTAP ネームスペースデバイスに 1MB の I/O 要求を送信できるようになります。

Marvell/QLogic

新しいSUSE Linux Enterprise Server 15 SP3 MUカーネルに含まれているネイティブの受信トレイqla2xxx ドライバには、最新のアップストリーム修正が適用されています。これらの修正は、ONTAPのサポートに不可欠です。

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。次に例を示します。

```
# cat /sys/class/fc_host/host*/symbolic_name
QLE2742 FW:v9.06.02 DVR:v10.02.00.106-k
QLE2742 FW:v9.06.02 DVR:v10.02.00.106-k
```

2. Marvell アダプタが NVMe/FC イニシエータとして機能できるように、「ql2xnvmeenable」が設定されていることを確認します。

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

NVMe/FC を設定

NVMe/FC とは異なり、NVMe/FC は自動接続機能を備えていません。これにより、Linux NVMe/FC ホストには次の 2 つの大きな制限があります。

- * パスが復活した後の自動再接続は行われない * NVMe/TCP は、パスダウン後 10 分間のデフォルトの「Ctrl-loss-TTMO」タイマーを超えて復活したパスに自動的に再接続することはできません。
- * ホストの起動時に自動接続が行われない * ホストの起動時に NVMe/FC が自動的に接続されることもありません。

タイムアウトを防ぐには、フェイルオーバーイベントの再試行期間を30分以上に設定する必要があります。Ctrl_loss_TMOタイマーの値を大きくすると、再試行期間を延長できます。詳細は次のとおりです。

手順

1. サポートされている NVMe/FC LIF の検出口グページデータをイニシエータポートが読み始めたかどうかを確認します。

```
# nvme discover -t tcp -w 192.168.1.8 -a 192.168.1.51
Discovery Log Number of Records 10, Generation counter 119
=====Discovery Log Entry 0=====
trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 0
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.56e362e9bb4f11ebbad039ea165abc:subsystem.nvme_118_tcp
_1
traddr: 192.168.2.56
sectype: none
=====Discovery Log Entry 1=====
trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 1
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.56e362e9bb4f11ebbad039ea165abc:subsystem.nvme_118_tcp
_1
traddr: 192.168.1.51
sectype: none
=====Discovery Log Entry 2=====
trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 0
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.56e362e9bb4f11ebbad039ea165abc:subsystem.nvme_118_tcp
_2
traddr: 192.168.2.56
sectype: none
...
...
```

2. 他の NVMe/FC イニシエータターゲット LIF のコンボファイルが検出口グページデータを正常に取得できることを確認します。例：

```
# nvme discover -t tcp -w 192.168.1.8 -a 192.168.1.52
# nvme discover -t tcp -w 192.168.2.9 -a 192.168.2.56
# nvme discover -t tcp -w 192.168.2.9 -a 192.168.2.57
```

3. を実行します `nvme connect-all` ノード間でサポートされるすべてのNVMe/FCイニシエータターゲットLIFに対して実行するコマンド。設定時間が長いことを確認してください `ctrl_loss_tmo` タイマー再試行期間（30分など、から設定できます）`-l 1800`) `connect-all`中に、パス損失が発生した場合に長期間再試行されるようにします。例：

```
# nvme connect-all -t tcp -w 192.168.1.8 -a 192.168.1.51 -l 1800
# nvme connect-all -t tcp -w 192.168.1.8 -a 192.168.1.52 -l 1800
# nvme connect-all -t tcp -w 192.168.2.9 -a 192.168.2.56 -l 1800
# nvme connect-all -t tcp -w 192.168.2.9 -a 192.168.2.57 -l 1800
```

NVMe-oF を検証します

1. 次のチェックボックスをオンにして、カーネル内の NVMe マルチパスが実際に有効になっていることを確認

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

2. 各 ONTAP ネームスペースの適切な NVMe-oF 設定（「NetApp ONTAP Controller」に設定された「model」や「loadbalancing iopolicy」が「ラウンドロビン」に設定されているなど）がホストに正しく反映されていることを確認します。

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

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

3. ONTAP ネームスペースがホストに正しく反映されていることを確認します。例：

```
# nvme list
Node SN Model Namespace
-----
/dev/nvme0n1 81CZ5BQuUNfGAAAAAAAB NetApp ONTAP Controller 1

Usage Format FW Rev
-----
85.90 GB / 85.90 GB 4 KiB + 0 B FFFFFFFF
```

別の例：

```
# nvme list
Node SN Model Namespace
-----
/dev/nvme0n1 81CYrBQuTHQFAAAAAAAC NetApp ONTAP Controller 1

Usage Format FW Rev
-----
85.90 GB / 85.90 GB 4 KiB + 0 B FFFFFFFF
```

4. 各バスのコントローラの状態がライブで、適切な ANA ステータスであることを確認します。例：

```
# nvme list-subsy /dev/nvme0n1
nvme-subsy1 - NQN=nqn.1992-
08.com.netapp:sn.04ba0732530911ea8e8300a098dfdd91:subsystem.nvme_145_1 \
+- nvme2 fc traddr=nn-0x208100a098dfdd91:pn-0x208200a098dfdd91
host_traddr=nn-0x200000109b579d5f:pn-0x100000109b579d5f live non-optimized
+- nvme3 fc traddr=nn-0x208100a098dfdd91:pn-0x208500a098dfdd91
host_traddr=nn-0x200000109b579d5e:pn-0x100000109b579d5e live non-optimized
+- nvme4 fc traddr=nn-0x208100a098dfdd91:pn-0x208400a098dfdd91
host_traddr=nn-0x200000109b579d5e:pn-0x100000109b579d5e live optimized
+- nvme6 fc traddr=nn-0x208100a098dfdd91:pn-0x208300a098dfdd91
host_traddr=nn-0x200000109b579d5f:pn-0x100000109b579d5f live optimized
```

別の例：

```
#nvme list-subsy /dev/nvme0n1
nvme-subsy0 - NQN=nqn.1992-
08.com.netapp:sn.37ba7d9cbfb11eba35dd039ea165514:subsystem.nvme_114_tcp
_1
 \
+- nvme0 tcp traddr=192.168.2.36 trsvcid=4420 host_traddr=192.168.1.4
live optimized
+- nvme1 tcp traddr=192.168.1.31 trsvcid=4420 host_traddr=192.168.1.4
live optimized
+- nvme10 tcp traddr=192.168.2.37 trsvcid=4420 host_traddr=192.168.1.4
live non-optimized
+- nvme11 tcp traddr=192.168.1.32 trsvcid=4420 host_traddr=192.168.1.4
live non-optimized
+- nvme20 tcp traddr=192.168.2.36 trsvcid=4420 host_traddr=192.168.2.5
live optimized
+- nvme21 tcp traddr=192.168.1.31 trsvcid=4420 host_traddr=192.168.2.5
live optimized
+- nvme30 tcp traddr=192.168.2.37 trsvcid=4420 host_traddr=192.168.2.5
live non-optimized
+- nvme31 tcp traddr=192.168.1.32 trsvcid=4420 host_traddr=192.168.2.5
live non-optimized
```

5. ネットアッププラグインにONTAPネームスペースデバイスごとに適切な値が表示されていることを確認します。例：

```

# nvme netapp ontapdevices -o column
Device          Vserver          Namespace Path
-----  -----  -----
----- /dev/nvme1n1 vserver_fcnvme_145 /vol/fcnvme_145_vol_1_0_0/fcnvme_145_ns

NSID  UUID                                Size
----  -----                                -----
1     23766b68-e261-444e-b378-2e84dbe0e5e1  85.90GB

# nvme netapp ontapdevices -o json
{
  "ONTAPdevices" : [
    {
      "Device" : "/dev/nvme1n1",
      "Vserver" : "vserver_fcnvme_145",
      "Namespace_Path" : "/vol/fcnvme_145_vol_1_0_0/fcnvme_145_ns",
      "NSID" : 1,
      "UUID" : "23766b68-e261-444e-b378-2e84dbe0e5e1",
      "Size" : "85.90GB",
      "LBA_Data_Size" : 4096,
      "Namespace_Size" : 20971520
    }
  ]
}

```

別の例：

```
# nvme netapp ontapdevices -o column
Device          Vserver          Namespace Path
-----
-----
/dev/nvme0n1  vs_tcp_114      /vol/tcpnvme_114_1_0_1/tcpnvme_114_ns

NSID  UUID                      Size
----  -----                      -----
1     a6aee036-e12f-4b07-8e79-4d38a9165686  85.90GB

# nvme netapp ontapdevices -o json
{
  "ONTAPdevices" : [
    {
      "Device" : "/dev/nvme0n1",
      "Vserver" : "vs_tcp_114",
      "Namespace_Path" : "/vol/tcpnvme_114_1_0_1/tcpnvme_114_ns",
      "NSID" : 1,
      "UUID" : "a6aee036-e12f-4b07-8e79-4d38a9165686",
      "Size" : "85.90GB",
      "LBA_Data_Size" : 4096,
      "Namespace_Size" : 20971520
    }
  ]
}
```

既知の問題

既知の問題はありません。

ONTAP を搭載した SUSE Linux Enterprise Server 15 SP2 用 NVMe/FC ホスト構成

NVMe/FCは、SUSE Linux Enterprise Server 15 SP2でONTAP 9.6以上でサポートされています。SUSE Linux Enterprise Server 15 SP2ホストでは、同じファイバチャネルリンクシエータアダプタポートを介してNVMe/FCトラフィックとFCPトラフィックの両方を実行できます。サポートされるFCアダプタとコントローラの一覧については、を参照してください "[Hardware Universe](#)"。

サポートされる構成とバージョンの最新のリストについては、を参照してください "[Interoperability Matrix Tool](#)"。



この手順で説明する構成設定を使用して、およびに接続されているクラウドクライアントを構成でき "Cloud Volumes ONTAP" "ONTAP 対応の Amazon FSX" ます。

既知の制限

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。

SUSE Linux Enterprise Server 15 SP2 で NVMe/FC を有効にする

- 推奨される SUSE Linux Enterprise Server 15 SP2 MU カーネルバージョンにアップグレードします。
- ネイティブの nvme-CLI パッケージをアップグレードします。

このネイティブの nvme-CLI パッケージには、NVMe/FC 自動接続スクリプト、ONTAP udev ルールが含まれています。このルールでは、NVMe マルチパスでのラウンドロビンロードバランシングや、ONTAP ネームスペース用のネットアッププラグインが有効になります。

```
# rpm -qa | grep nvme-cli  
nvme-cli-1.10-2.38.x86_64
```

- SUSE Linux Enterprise Server 15 SP2 ホストで、のホスト NQN 文字列を確認し /etc/nvme/hostnqn 、ONTAP アレイの対応するサブシステムのホスト NQN 文字列と一致することを確認します。例：

```
# cat /etc/nvme/hostnqn  
nqn.2014-08.org.nvmexpress:uuid:3ca559e1-5588-4fc4-b7d6-5ccfb0b9f054
```

```
::> vserver nvme subsystem host show -vserver vs_fcnvme_145  
Vserver Subsystem Host NQN  
-----  
-----  
vs_fcnvme_145  
nvme_145_1  
nqn.2014-08.org.nvmexpress:uuid:c7b07b16-a22e-41a6-a1fd-cf8262c8713f  
nvme_145_2  
nqn.2014-08.org.nvmexpress:uuid:c7b07b16-a22e-41a6-a1fd-cf8262c8713f  
nvme_145_3  
nqn.2014-08.org.nvmexpress:uuid:c7b07b16-a22e-41a6-a1fd-cf8262c8713f  
nvme_145_4  
nqn.2014-08.org.nvmexpress:uuid:c7b07b16-a22e-41a6-a1fd-cf8262c8713f  
nvme_145_5  
nqn.2014-08.org.nvmexpress:uuid:c7b07b16-a22e-41a6-a1fd-cf8262c8713f  
5 entries were displayed.
```

- ホストをリブートします。

Broadcom FC アダプタを NVMe/FC 用に設定します

1. サポートされているアダプタを使用していることを確認します。サポートされているアダプタの現在のリストについては、を参照してください["Interoperability Matrix Tool"](#)。

```
# cat /sys/class/scsi_host/host*/modelname
LPe32002-M2
LPe32002-M2
```

```
# cat /sys/class/scsi_host/host*/modeldesc
Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
```

2. 推奨される Broadcom lpfc ファームウェアとネイティブインボックスドライバのバージョンを使用していることを確認します。

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

```
# cat /sys/module/lpfc/version
0:12.8.0.2
```

3. lpfc_enable_fc4_type が 3 に設定されていることを確認します

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

4. イニシエータポートが動作していることを確認します。

```
# cat /sys/class/fc_host/host*/port_name
0x100000109b579d5e
0x100000109b579d5f
```

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

5. NVMe/FC イニシエータポートが有効になっており、実行中で、ターゲット LIF を認識できることを確認

します。

```
# cat /sys/class/scsi_host/host*/nvme_info
NVME Initiator Enabled
XRI Dist lpfc0 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc0 WWPN x100000109b579d5e WWNN x200000109b579d5e DID
x011c00 ONLINE
NVME RPORT WWPN x208400a098dfdd91 WWNN x208100a098dfdd91 DID x011503
TARGET DISCSRVC ONLINE
NVME RPORT WWPN x208500a098dfdd91 WWNN x208100a098dfdd91 DID x010003
TARGET DISCSRVC ONLINE
NVME Statistics
LS: Xmt 0000000e49 Cmpl 0000000e49 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 00000003ceb594f Issue 00000003ce65dbe OutIO
fffffffffffffb046f
abort 00000bd2 noxri 00000000 nondlp 00000000 qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 000014f4 Err 00012abd
NVME Initiator Enabled
XRI Dist lpfc1 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc1 WWPN x100000109b579d5f WWNN x200000109b579d5f DID
x011b00 ONLINE
NVME RPORT WWPN x208300a098dfdd91 WWNN x208100a098dfdd91 DID x010c03
TARGET DISCSRVC ONLINE
NVME RPORT WWPN x208200a098dfdd91 WWNN x208100a098dfdd91 DID x012a03
TARGET DISCSRVC ONLINE
NVME Statistics
LS: Xmt 0000000e50 Cmpl 0000000e50 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 00000003c9859ca Issue 00000003c93515e OutIO
fffffffffffffaf794
abort 00000b73 noxri 00000000 nondlp 00000000 qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 0000159d Err 000135c3
```

NVMe/FC を検証

1. 以下の NVMe/FC 設定を確認してください。

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

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

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

2. ネームスペースが作成されたことを確認します。

```
# nvme list  
Node SN Model Namespace Usage Format FW Rev  
-----  
-----  
-----  
/dev/nvme1n1 814vWBNRwfBGAAAAAAAB NetApp ONTAP Controller 1 85.90 GB /  
85.90 GB 4 KiB + 0 B FFFFFFFF
```

3. ANA パスのステータスを確認します。

```
# nvme list-subsy /dev/nvme1n1  
nvme-subsy1 - NQN=nqn.1992-  
08.com.netapp:sn.04ba0732530911ea8e8300a098dfdd91:subsystem.nvme_145_1  
\  
+- nvme2 fc traddr=nn-0x208100a098dfdd91:pn-0x208200a098dfdd91  
host_traddr=nn-0x200000109b579d5f:pn-0x100000109b579d5f live  
inaccessible  
+- nvme3 fc traddr=nn-0x208100a098dfdd91:pn-0x208500a098dfdd91  
host_traddr=nn-0x200000109b579d5e:pn-0x100000109b579d5e live  
inaccessible  
+- nvme4 fc traddr=nn-0x208100a098dfdd91:pn-0x208400a098dfdd91  
host_traddr=nn-0x200000109b579d5e:pn-0x100000109b579d5e live optimized  
+- nvme6 fc traddr=nn-0x208100a098dfdd91:pn-0x208300a098dfdd91  
host_traddr=nn-0x200000109b579d5f:pn-0x100000109b579d5f live optimized
```

4. ONTAP デバイス用ネットアッププラグインを確認します。

```

# nvme netapp ontapdevices -o column
Device Vserver Namespace Path NSID UUID Size
-----
-----
-----
/dev/nvme1n1 vserver_fcnvme_145 /vol/fcnvme_145_vol_1_0_0/fcnvme_145_ns
1 23766b68-e261-444e-b378-2e84dbe0e5e1 85.90GB

# nvme netapp ontapdevices -o json
{
  "ONTAPdevices" : [
    {
      "Device" : "/dev/nvme1n1",
      "Vserver" : "vserver_fcnvme_145",
      "Namespace_Path" : "/vol/fcnvme_145_vol_1_0_0/fcnvme_145_ns",
      "NSID" : 1,
      "UUID" : "23766b68-e261-444e-b378-2e84dbe0e5e1",
      "Size" : "85.90GB",
      "LBA_Data_Size" : 4096,
      "Namespace_Size" : 20971520
    },
  ]
}

```

既知の問題

既知の問題はありません。

Broadcom NVMe/FC の 1MB I/O サイズを有効にします

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
```

lpfc 詳細ログ

NVMe/FC用のLPFC ドライバを設定します。

手順

1. を設定します lpfc_log_verbose NVMe/FCイベントをログに記録するためのドライバ設定は次のいずれかです。

```
#define LOG_NVME 0x00100000 /* NVME general events. */
#define LOG_NVME_DISC 0x00200000 /* NVME Discovery/Connect events. */
#define LOG_NVME_ABTS 0x00400000 /* NVME ABTS events. */
#define LOG_NVME_IOERR 0x00800000 /* NVME IO Error events. */
```

2. 値を設定したら、を実行します dracut-f コマンドを実行し、ホストをリブートします。
3. 設定を確認します。

```
# cat /etc/modprobe.d/lpfc.conf options lpfc lpfc_log_verbose=0xf00083
# cat /sys/module/lpfc/parameters/lpfc_log_verbose 15728771
```

SUSE Linux Enterprise Server 15 SP1 と ONTAP の NVMe/FC ホスト構成

NVMe over Fibre Channel (NVMe/FC) は、SUSE Linux Enterprise Server 15 SP1およびONTAPを実行するホストにターゲットとして設定できます。

ONTAP 9.6以降では、次のバージョンのSUSE Linux Enterprise ServerでNVMe/FCがサポートされます。

- SUSE Linux Enterprise Server 15 SP1

SUSE Linux Enterprise Server 15 SP1ホストは、同じファイバチャネルイニシエータアダプタポートを介してNVMe/FCトラフィックとFCPトラフィックの両方を実行できます。サポートされるFCアダプタとコントローラの一覧については、を参照してください "Hardware Universe" 。

サポートされる構成とバージョンの最新のリストについては、を参照してください["Interoperability Matrix Tool"](#)。

- NVMe/FC 標準の自動接続スクリプトは、 nvme-CLI パッケージに含まれています。SUSE Linux Enterprise Server 15 SP1では、ネイティブの受信トレイlpfcドライバを使用できます。

既知の制限

NVMe-oFプロトコルを使用したSANポートは現在サポートされていません。

SUSE Linux Enterprise Server 15 SP1でのNVMe/FCの有効化

1. 推奨されるSUSE Linux Enterprise Server 15 SP2 MUカーネルへのアップグレード
2. 推奨される nvme-CLI MU バージョンにアップグレードします。

このNVMe-CLIパッケージにはNVMe/FC自動接続スクリプトが標準で含まれているため、Broadcomが提供する外部NVMe/FC自動接続スクリプトをSUSE Linux Enterprise Server 15 SP1ホストにインストールする必要はありません。このパッケージには、ONTAPのudevルールも含まれています。このルールでは、NVMeマルチパスでのラウンドロビンロードバランシングや、ONTAPデバイス用ネットアッププラグインが有効になります。

```
# rpm -qa | grep nvme-cli  
nvme-cli-1.8.1-6.9.1.x86_64
```

3. SUSE Linux Enterprise Server 15 SP1ホストで、のホストNQN文字列を確認し /etc/nvme/hostnqn 、ONTAPアレイの対応するサブシステムのホストNQN文字列と一致することを確認します。例：

```
# cat /etc/nvme/hostnqn  
nqn.2014-08.org.nvmexpress:uuid:75953f3b-77fe-4e03-bf3c-09d5a156fbcd
```

```
*> vserver nvme subsystem host show -vserver vs_nvme_10  
Vserver Subsystem Host NQN  
-----  
sles_117_nvme_ss_10_0  
nqn.2014-08.org.nvmexpress:uuid:75953f3b-77fe-4e03-bf3c-09d5a156fbcd
```

4. ホストをリブートします。

Broadcom FC アダプタを NVMe/FC 用に設定します

1. サポートされているアダプタを使用していることを確認します。サポートされているアダプタの現在のリストについては、を参照してください["Interoperability Matrix Tool"](#)。

```
# cat /sys/class/scsi_host/host*/modelname  
LPe32002-M2  
LPe32002-M2
```

```
# cat /sys/class/scsi_host/host*/modeldesc  
Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter  
Emulex LightPulse LPe32002-M2 2-Port 32Gb Fibre Channel Adapter
```

2. 推奨される Broadcom lpfc フームウェアとネイティブインボックスドライバのバージョンを使用していることを確認します。

```
# cat /sys/class/scsi_host/host*/fwrev  
12.4.243.17, sil-4.2.c  
12.4.243.17, sil-4.2.c
```

```
# cat /sys/module/lpfc/version  
0:12.6.0.0
```

3. lpfc_enable_fc4_type が 3 に設定されていることを確認します

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

4. イニシエータポートが動作していることを確認します。

```
# cat /sys/class/fc_host/host*/port_name  
0x10000090fae0ec61  
0x10000090fae0ec62
```

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

5. NVMe/FC イニシエータポートが有効になっており、実行中で、ターゲット LIF を認識できることを確認します。

```
# cat /sys/class/scsi_host/host*/nvme_info
NVME Initiator Enabled
XRI Dist lpfc0 Total 6144 NVME 2947 SCSI 2977 ELS 250
NVME LPORT lpfc0 WWPN x10000090fae0ec61 WWNN x20000090fae0ec61 DID
x012000 ONLINE
NVME RPORT WWPN x202d00a098c80f09 WWNN x202c00a098c80f09 DID x010201
TARGET DISCSRVC ONLINE
NVME RPORT WWPN x203100a098c80f09 WWNN x202c00a098c80f09 DID x010601
TARGET DISCSRVC ONLINE
NVME Statistics
...
```

NVMe/FC を検証

- 以下の NVMe/FC 設定を確認してください。

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

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

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

- ネームスペースが作成されたことを確認します。

```
# nvme list
Node SN Model Namespace Usage Format FW Rev
-----
/dev/nvme0n1 80BADBKnB/JvAAAAAAAC NetApp ONTAP Controller 1 53.69 GB /
53.69 GB 4 KiB + 0 B FFFFFFFF
```

- ANA パスのステータスを確認します。

```
# nvme list-subsy /dev/nvme0n1
Nvme-subsyf0 - NQN=nqn.1992-
08.com.netapp:sn.341541339b9511e8a9b500a098c80f09:subsystem.sles_117_nvm
e_ss_10_0
\
+- nvme0 fc traddr=nn-0x202c00a098c80f09:pn-0x202d00a098c80f09
host_traddr=nn-0x20000090fae0ec61:pn-0x10000090fae0ec61 live optimized
+- nvme1 fc traddr=nn-0x207300a098dfdd91:pn-0x207600a098dfdd91
host_traddr=nn-0x20000109b1c1204:pn-0x10000109b1c1204 live
inaccessible
+- nvme2 fc traddr=nn-0x207300a098dfdd91:pn-0x207500a098dfdd91
host_traddr=nn-0x20000109b1c1205:pn-0x10000109b1c1205 live optimized
+- nvme3 fc traddr=nn-0x207300a098dfdd91:pn-0x207700a098dfdd91 host
traddr=nn-0x20000109b1c1205:pn-0x10000109b1c1205 live inaccessible
```

4. ONTAP デバイス用ネットアッププラグインを確認します。

```
# nvme netapp ontapdevices -o column
Device Vserver Namespace Path NSID UUID Size
----- -----
/dev/nvme0n1 vs_nvme_10 /vol/sles_117_vol_10_0/sles_117_ns_10_0
1 55baf453-f629-4a18-9364-b6aee3f50dad 53.69GB

# nvme netapp ontapdevices -o json
{
  "ONTAPdevices" : [
    {
      "Device" : "/dev/nvme0n1",
      "Vserver" : "vs_nvme_10",
      "Namespace_Path" : "/vol/sles_117_vol_10_0/sles_117_ns_10_0",
      "NSID" : 1,
      "UUID" : "55baf453-f629-4a18-9364-b6aee3f50dad",
      "Size" : "53.69GB",
      "LBA_Data_Size" : 4096,
      "Namespace_Size" : 13107200
    }
  ]
}
```

既知の問題

既知の問題はありません。

Broadcom NVMe/FC の 1MB I/O サイズを有効にします

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
```

lpfc 詳細ログ

NVMe/FC 用の LPFC ドライバを設定します。

手順

1. を設定します `lpfc_log_verbose` NVMe/FC イベントをログに記録するための ドライバ 設定は次のいずれかです。

```
#define LOG_NVME 0x00100000 /* NVME general events. */
#define LOG_NVME_DISC 0x00200000 /* NVME Discovery/Connect events. */
#define LOG_NVME_ABTS 0x00400000 /* NVME ABTS events. */
#define LOG_NVME_IOERR 0x00800000 /* NVME IO Error events. */
```

2. 値を設定したら、を実行します `dracut-f` コマンドを実行し、ホストをリブートします。
3. 設定を確認します。

```
# cat /etc/modprobe.d/lpfc.conf options lpfc lpfc_log_verbose=0xf00083  
# cat /sys/module/lpfc/parameters/lpfc_log_verbose 15728771
```

著作権に関する情報

Copyright © 2025 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。