SUSE Linux Enterprise Server

ONTAP SAN Host Utilities

NetApp
January 30, 2026

목차

SUSE Linux Enterprise Server

SUSE Linux Enterprise Server의 ONTAP 지원 및 기능에 대해 알아보세요.

NVMe over Fabrics(NVMe-oF)를 사용한 호스트 구성에서 지원되는 기능은 ONTAP 및 SUSE Linux Enterprise Server 버전에 따라 다릅니다.

특징	**SUSE Linux Enterprise Server** 호스트 버전	**ONTAP** 버전
NVMe/TCP는 ASA r2 시스템에 대한 모든 최적화된 경로를 보고합니다.	16	9.16.1 이상
NVMe/TCP는 전송 계층 보안(TLS) 1.3 암호화를 지원합니다.	15 SP6 이상	9.16.1 이상
RHEL 호스트와 ONTAP 컨트롤러 간의 NVMe/TCP를 통해 안전한 인밴드 인증이 지원됩니다.	15 SP4 이상	9.12.1 이상
고유한 검색 NQN을 사용하여 지속적 검색 컨트롤러(PDC)가 지원됩니다.	15 SP4 이상	9.11.1 이상
NVMe/TCP는 네이티브를 사용하여 네임스페이스를 제공합니다. `nvme-cli` 패키지	15 SP4 이상	9.10.1 이상
NVMe 및 SCSI 트래픽은 NVMe-oF 네임스페이스의 경우 NVMe 멀티패스를 사용하고 SCSI LUN의 경우 dm-multipath를 사용하여 동일한 호스트에서 지원됩니다.	15 SP1 이상	9.4 이상

ONTAP 시스템 설정에서 실행되는 ONTAP 버전에 관계없이 다음과 같은 SAN 호스트 기능을 지원합니다.

특징	**SUSE Linux Enterprise Server** 호스트 버전
NVMe/FC 프로토콜을 사용하여 SAN 부팅이 활성화됩니다.	15 SP7 이상
기본적으로 네이티브 NVMe 멀티패스가 활성화되어 있습니다.	15 SP1 이상
그만큼 `nvme-cli` 패키지에는 타사 스크립트가 필요 없는 자동 연결 스크립트가 포함되어 있습니다.	15 SP1 이상

지원되는 구성에 대한 자세한 내용은 다음을 참조하세요."상호 운용성 매트릭스 툴" .

다음은 무엇입니까?

SUSE Linux Enterprise Server 버전이 ...인 경우	다음 내용을 알아보세요...
16	"SUSE Linux Enterprise Server 16용 NVMe 구성"
15 SPx 시리즈	"SUSE Linux Enterprise Server 15 SPx용 NVMe 구성"

관련 정보

- "ASA r2 시스템에 대해 알아보세요"
- "NVMe 프로토콜 관리에 대해 알아보세요"

ONTAP 스토리지를 사용하여 NVMe-oF를 지원하도록 SUSE Linux Enterprise Server 16을 구성합니다.

SUSE Linux Enterprise Server 16 호스트는 비대칭 네임스페이스 액세스(ANA)를 지원하는 NVMe over Fibre Channel(NVMe/FC) 및 NVMe over TCP(NVMe/TCP) 프로토콜을 지원합니다. ANA는 iSCSI 및 FCP 환경의 비대칭 논리 장치 액세스(ALUA)와 동일한 멀티패싱 기능을 제공합니다.

SUSE Linux Enterprise Server 16용 NVMe over Fabrics(NVMe-oF) 호스트 구성 방법을 알아보세요. 더 자세한 지원 및 기능 정보는 "ONTAP 지원 및 기능"을 참조하십시오.

SUSE Linux Enterprise Server 16의 NVMe-oF에는 다음과 같은 알려진 제한 사항이 있습니다.

- 그만큼 `nvme disconnect-all` 이 명령을 실행하면 루트와 데이터 파일 시스템의 연결이 모두 끊어지고 시스템이 불안정해질 수 있습니다. NVMe-TCP 또는 NVMe-FC 네임스페이스를 통해 SAN에서 부팅하는 시스템에서는 이 명령을 실행하지 마세요.
- NetApp sanlun 호스트 유틸리티는 NVMe-oF를 지원하지 않습니다. 대신, 기본 제공되는 NetApp 플러그인을 사용할 수 있습니다. `nvme-cli` 모든 NVMe-oF 전송을 위한 패키지입니다.

1단계: 필요에 따라 SAN 부팅을 활성화합니다

SAN 부팅을 사용하도록 호스트를 구성하여 배포를 간소화하고 확장성을 개선할 수 있습니다. 사용하다"상호 운용성 매트릭스 툴" Linux OS, 호스트 버스 어댑터(HBA), HBA 펌웨어, HBA 부팅 BIOS 및 ONTAP 버전이 SAN 부팅을 지원하는지 확인하세요.

단계

1. "NVMe 네임스페이스를 생성하고 호스트에 매핑합니다." .
2. SAN 부팅 네임스페이스가 매핑된 포트에 대해 서버 BIOS에서 SAN 부팅을 활성화합니다.

   HBA BIOS를 활성화하는 방법에 대한 자세한 내용은 공급업체별 설명서를 참조하십시오.

3. 호스트를 재부팅하고 OS가 제대로 실행 중인지 확인하세요.

2단계: SUSE Linux Enterprise Server 및 NVMe 소프트웨어를 설치하고 구성을 확인합니다.

NVMe-oF를 사용하도록 호스트를 구성하려면 호스트 및 NVMe 소프트웨어 패키지를 설치하고, 멀티패싱을 활성화하고, 호스트의 NQN 구성을 확인해야 합니다.

단계

1. 서버에 SUSE Linux Enterprise Server 16을 설치하십시오. 설치가 완료되면 지정된 SUSE Linux Enterprise Server 16 커널이 실행 중인지 확인하십시오.

```
uname -r
```

SUSE Linux Enterprise Server 커널 버전 예:

```
6.12.0-160000.6-default
```

2. "NVMe-CLI" 패키지를 설치합니다.

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

다음 예에서는 다음을 보여줍니다. `nvme-cli` 패키지 버전:

```
nvme-cli-2.11+29.g35e62868-160000.1.1.x86_64
```

3. 를 설치합니다 `libnvme` 패키지:

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

다음 예에서는 다음을 보여줍니다. `libnvme` 패키지 버전:

```
libnvme1-1.11+17.g6d55624d-160000.1.1.x86_64
```

4. 호스트에서 hostnqn 문자열을 확인하세요. `/etc/nvme/hostnqn` :

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

다음 예에서는 다음을 보여줍니다. `hostnqn` 버전:

```
nqn.2014-08.org.nvmexpress:uuid:d3b581b4-c975-11e6-8425-0894ef31a074
```

5. ONTAP 시스템에서 다음 사항을 확인하십시오. `hostnqn` 문자열이 일치합니다 `hostnqn` ONTAP 어레이의 해당 서브시스템에 대한 문자열:

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

예제 보기

```
Vserver Subsystem Priority  Host NQN
------- --------- --------
-------------------------------------------------
vs_coexistence_emulex
        nvme1
                  regular   nqn.2014-
08.org.nvmexpress:uuid:d3b581b4-c975-11e6-8425-0894ef31a074
        nvme10
                  regular   nqn.2014-
08.org.nvmexpress:uuid:d3b581b4-c975-11e6-8425-0894ef31a074
        nvme11
                  regular   nqn.2014-
08.org.nvmexpress:uuid:d3b581b4-c975-11e6-8425-0894ef31a074
        nvme12
                  regular   nqn.2014-
08.org.nvmexpress:uuid:d3b581b4-c975-11e6-8425-0894ef31a074
4 entries were displayed.
```

를 누릅니다 `hostnqn` 문자열이 일치하지 않습니다. 를 사용하십시오 `vserver modify` 명령을 사용하여 를 업데이트합니다 `hostnqn` 와 일치하는 해당 ONTAP 배열 하위 시스템의 문자열입니다 `hostnqn` 문자열 시작 `/etc/nvme/hostnqn` 호스트.

3단계: NVMe/FC 및 NVMe/TCP 구성

Broadcom/Emulex 또는 Marvell/QLogic 어댑터를 사용하여 NVMe/FC를 구성하거나 수동 검색 및 연결 작업을 사용하여 NVMe/TCP를 구성합니다.

NVMe/FC - Broadcom/Emulex

Broadcom/Emulex FC 어댑터용 NVMe/FC를 구성합니다.

단계

1. 지원되는 어댑터 모델을 사용하고 있는지 확인합니다.

   a. 모델 이름을 표시합니다:

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

   다음과 같은 출력이 표시됩니다.

   ```
   SN37A92079
   SN37A92079
   ```

   b. 모델 설명을 표시합니다.

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

   다음과 같은 출력이 표시됩니다.

   ```
   Emulex SN37A92079 32Gb 2-Port Fibre Channel Adapter
   Emulex SN37A92079 32Gb 2-Port Fibre Channel Adapter
   ```

2. 권장 Broadcom을 사용하고 있는지 확인합니다 `lpfc` 펌웨어 및 받은 편지함 드라이버:

   a. 펌웨어 버전을 표시합니다.

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

   다음 예에서는 펌웨어 버전을 보여줍니다.

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

   b. 받은 편지함 드라이버 버전을 표시합니다.

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

다음 예에서는 드라이버 버전을 보여줍니다.

```
0:14.4.0.11
```

지원되는 어댑터 드라이버 및 펌웨어 버전의 현재 목록은 를 참조하십시오"상호 운용성 매트릭스 툴".

3. 의 예상 출력이 3 다음과 같이 설정되었는지 확인합니다 `lpfc_enable_fc4_type`.

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

4. 이니시에이터 포트를 볼 수 있는지 확인합니다.

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

다음과 유사한 출력이 표시됩니다.

```
0x100000109bdacc75
0x100000109bdacc76
```

5. 이니시에이터 포트가 온라인 상태인지 확인합니다.

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

다음과 같은 출력이 표시됩니다.

```
Online
Online
```

6. NVMe/FC 이니시에이터 포트가 활성화되었고 타겟 포트가 표시되는지 확인합니다.

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

예제 출력을 표시합니다

```
NVME Initiator Enabled
XRI Dist lpfc0 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc0 WWPN x100000109bdacc75 WWNN x200000109bdacc75
DID x060100 ONLINE
NVME RPORT       WWPN x2001d039ea951c45 WWNN x2000d039ea951c45
DID x080801 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2003d039ea951c45 WWNN x2000d039ea951c45
DID x080d01 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2024d039eab31e9c WWNN x2023d039eab31e9c
DID x020a09 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2026d039eab31e9c WWNN x2023d039eab31e9c
DID x020a08 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2003d039ea5cfc90 WWNN x2002d039ea5cfc90
DID x061b01 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2012d039ea5cfc90 WWNN x2011d039ea5cfc90
DID x061b05 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2005d039ea5cfc90 WWNN x2002d039ea5cfc90
DID x061201 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2014d039ea5cfc90 WWNN x2011d039ea5cfc90
DID x061205 TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 0000017242 Cmpl 0000017242 Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 0000000000378362 Issue 00000000003783c7 OutIO
0000000000000065
        abort 00000409 noxri 00000000 nondlp 0000003a qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 00000409 Err 0000040a

NVME Initiator Enabled
XRI Dist lpfc1 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc1 WWPN x100000109bdacc76 WWNN x200000109bdacc76
DID x062800 ONLINE
NVME RPORT       WWPN x2002d039ea951c45 WWNN x2000d039ea951c45
DID x080701 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2004d039ea951c45 WWNN x2000d039ea951c45
DID x081501 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2025d039eab31e9c WWNN x2023d039eab31e9c
DID x020913 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2027d039eab31e9c WWNN x2023d039eab31e9c
DID x020912 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2006d039ea5cfc90 WWNN x2002d039ea5cfc90
DID x061401 TARGET DISCSRVC ONLINE
```

```
NVME RPORT       WWPN x2015d039ea5cfc90 WWNN x2011d039ea5cfc90
DID x061405 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2004d039ea5cfc90 WWNN x2002d039ea5cfc90
DID x061301 TARGET DISCSRVC ONLINE
NVME RPORT       WWPN x2013d039ea5cfc90 WWNN x2011d039ea5cfc90
DID x061305 TARGET DISCSRVC ONLINE

NVME Statistics
LS: Xmt 0000017428 Cmpl 0000017428 Abort 00000000
LS XMIT: Err 00000000  CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 00000000003443be Issue 000000000034442a OutIO
000000000000006c
        abort 00000491 noxri 00000000 nondlp 00000086 qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 00000491 Err 00000494
```

NVMe/FC - Marvell/QLogic

Marvell/QLogic 어댑터용 NVMe/FC를 구성합니다.

단계

1. 지원되는 어댑터 드라이버 및 펌웨어 버전을 실행하고 있는지 확인합니다.

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

다음 예에서는 드라이버 및 펌웨어 버전을 보여줍니다.

```
QLE2772 FW:v9.15.06 DVR:v10.02.09.400-k-debug
QLE2772 FW:v9.15.06 DVR:v10.02.09.400-k-debug
```

2. 확인합니다 `ql2xnvmeenable` 가 설정됩니다. 그러면 Marvell 어댑터가 NVMe/FC Initiator로 작동할 수 있습니다.

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

예상 출력은 1입니다.

NVMe/TCP

NVMe/TCP 프로토콜은 자동 연결 작업을 지원하지 않습니다. 대신 NVMe/TCP를 수행하여 NVMe/TCP 하위 시스템과 네임스페이스를 검색할 수 있습니다. `connect` 또는 `connect-all` 수동으로 작업합니다.

단계

1. 이니시에이터 포트가 지원되는 NVMe/TCP LIF에서 검색 로그 페이지 데이터를 가져올 수 있는지 확인합니다.

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

예제 출력을 표시합니다

```
nvme discover -t tcp -w 192.168.38.20 -a 192.168.38.10
Discovery Log Number of Records 8, Generation counter 42
=====Discovery Log Entry 0======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  4
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.f8e2af201b7211f0ac2bd039eab67a95:discovery
traddr:  192.168.211.71
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:  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
adrfam:  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.samp
le_tcp_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.samp
le_tcp_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.samp
```

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

2. 다른 모든 NVMe/TCP 이니시에이터-타겟 LIF 조합이 검색 로그 페이지 데이터를 성공적으로 가져올 수 있는지 확인합니다.

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

예제 보기

```
nvme discover -t tcp -w 192.168.38.20 -a 192.168.38.10
nvme discover -t tcp -w 192.168.38.20 -a 192.168.38.11
nvme discover -t tcp -w 192.168.39.20 -a 192.168.39.10
nvme discover -t tcp -w 192.168.39.20 -a 192.168.39.11
```

3. 를 실행합니다 `nvme connect-all` 노드에 걸쳐 지원되는 모든 NVMe/TCP 이니시에이터-타겟 LIF에 대한 명령:

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

예제 보기

```
nvme    connect-all -t  tcp -w  192.168.38.20   -a
192.168.38.10
nvme    connect-all -t  tcp -w  192.168.38.20   -a
192.168.38.11
nvme    connect-all -t  tcp -w  192.168.39.20   -a
192.168.39.10
nvme    connect-all -t  tcp -w  192.168.39.20   -a
192.168.39.11
```

NVMe/TCP 설정 `ctrl_loss_tmo timeout` 자동으로 "꺼짐"으로 설정됩니다. 결과적으로:

- 재시도 횟수에 제한이 없습니다(무기한 재시도).
- 특정 항목을 수동으로 구성할 필요가 없습니다. `ctrl_loss_tmo timeout` 사용 시 지속 시간 `nvme connect` 또는 `nvme connect-all` 명령어(옵션 -l).
- NVMe/TCP 컨트롤러는 경로 장애가 발생해도 시간 초과가 발생하지 않으며 무기한 연결 상태를 유지합니다.

4단계: 선택적으로 **udev** 규칙에서 **iopolicy**를 수정합니다.

SUSE Linux Enterprise Server 16부터 NVMe-oF의 기본 iopolicy는 `queue-depth`로 설정되어 있습니다. iopolicy를 `round-robin`로 변경하려면 udev 규칙 파일을 다음과 같이 수정하십시오.

단계

1. 루트 권한으로 텍스트 편집기에서 udev 규칙 파일을 엽니다.

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

다음과 같은 출력이 표시됩니다.

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

2. 다음 예시 규칙에 나와 있는 것처럼 NetApp ONTAP 컨트롤러의 iopolicy를 설정하는 줄을 찾으십시오.

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

3. 규칙을 수정하여 `queue-depth`이(가) `round-robin`이(가) 되도록 합니다:

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

4. udev 규칙을 다시 로드하고 변경 사항을 적용합니다.

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

5. 하위 시스템의 현재 iopolicy를 확인하세요. 예를 들어 <하위 시스템>을 다음과 같이 바꾸세요. `nvme-subsys0` .

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

다음과 같은 출력이 표시됩니다.

```
round-robin
```

새로운 iopolicy는 일치하는 NetApp ONTAP 컨트롤러 장치에 자동으로 적용됩니다. 재부팅할 필요가 없습니다.

5단계: 선택적으로 NVMe/FC에 대해 1MB I/O를 활성화합니다.

ONTAP Identify Controller 데이터에서 최대 데이터 전송 크기(MDTS)를 8로 보고합니다. 즉, 최대 I/O 요청 크기는 1MB까지 가능합니다. Broadcom NVMe/FC 호스트에 대해 1MB 크기의 I/O 요청을 발행하려면 다음을 늘려야 합니다. `lpfc` 의 가치 `lpfc_sg_seg_cnt` 매개변수를 기본값 64에서 256으로 변경합니다.

이 단계는 Qlogic NVMe/FC 호스트에는 적용되지 않습니다.

단계

1. `lpfc_sg_seg_cnt`매개변수를 256으로 설정합니다.

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

다음 예와 비슷한 출력이 표시되어야 합니다.

```
options lpfc lpfc_sg_seg_cnt=256
```

2. `dracut -f`명령을 실행하고 호스트를 재부팅합니다.
3. 의 값이 256인지 `lpfc_sg_seg_cnt` 확인합니다.

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

6단계: NVMe 부팅 서비스 확인

그만큼 `nvmefc-boot-connections.service` 그리고 `nvmf-autoconnect.service` NVMe/FC에 포함된 부팅 서비스 `nvme-cli` 패키지는 시스템이 부팅될 때 자동으로 활성화됩니다.

부팅이 완료된 후 다음을 확인하세요. `nvmefc-boot-connections.service` 그리고 `nvmf-autoconnect.service` 부팅 서비스가 활성화되었습니다.

단계

1. 가 활성화되어 있는지 `nvmf-autoconnect.service` 확인합니다.

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

예제 출력을 표시합니다

```
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.
```

2. 가 활성화되어 있는지 `nvmefc-boot-connections.service` 확인합니다.

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

예제 출력을 표시합니다

```
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.
```

7단계: 다중 경로 구성 확인

커널 내 NVMe 다중 경로 상태, ANA 상태 및 ONTAP 네임스페이스가 NVMe-oF 구성에 적합한지 확인합니다.

단계

1. in-kernel NVMe multipath가 활성화되어 있는지 확인합니다.

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

다음과 같은 출력이 표시됩니다.

```
Y
```

2. 해당 ONTAP 네임스페이스에 대한 적절한 NVMe-oF 설정(예: 모델이 NetApp ONTAP 컨트롤러로 설정되고 로드 밸런싱 iopolicy가 queue-depth로 설정됨)이 호스트에 올바르게 반영되었는지 확인하십시오.

   a. 하위 시스템을 표시합니다.

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

다음과 같은 출력이 표시됩니다.

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

b. 정책을 표시합니다.

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

다음과 같은 출력이 표시됩니다.

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

3. 호스트에서 네임스페이스가 생성되고 올바르게 검색되는지 확인합니다.

```
nvme list
```

예제 보기

```
Node          SN                    Model
------------------------------------------------------------
/dev/nvme7n1 81Ix2BVuekWcAAAAAAAB   NetApp ONTAP Controller

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

4. 각 경로의 컨트롤러 상태가 라이브이고 올바른 ANA 상태인지 확인합니다.

```
nvme list-subsys /dev/<controller_ID>
```

ONTAP 9.16.1부터 NVMe/FC 및 NVMe/TCP는 ASA r2 시스템에서 최적화된 모든 경로를 보고합니다.

NVMe/FC

다음 예시 출력은 NVMe/FC를 사용하는 AFF, FAS, ASA 시스템 및 ASA r2 시스템을 위한 2노드 ONTAP 컨트롤러에서 호스팅되는 네임스페이스를 보여줍니다.

AFF, FAS 및 **ASA** 예시 출력 표시

```
 nvme-subsys114 - NQN=nqn.1992-
08.com.netapp:sn.9e30b9760a4911f08c87d039eab67a95:subsystem.sles
_161_27
                 hostnqn=nqn.2014-
08.org.nvmexpress: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
```

ASA r2 예제 출력 표시

```
nvme-subsys96 - NQN=nqn.1992-
08.om.netapp:sn.b351b2b6777b11f0b3c2d039ea5cfc91:subsystem.nvme2
4
                hostnqn=nqn.2014-
08.org.nvmexpress:uuid:d3b581b4-c975-11e6-8425-0894ef31a074
\
 +- nvme203 fc traddr=nn-0x2011d039ea5cfc90:pn-
0x2015d039ea5cfc90,host_traddr=nn-0x200000109bdacc76:pn-
0x100000109bdacc76 live optimized
 +- nvme25 fc traddr=nn-0x2011d039ea5cfc90:pn-
0x2014d039ea5cfc90,host_traddr=nn-0x200000109bdacc75:pn-
0x100000109bdacc75 live optimized
 +- nvme30 fc traddr=nn-0x2011d039ea5cfc90:pn-
0x2012d039ea5cfc90,host_traddr=nn-0x200000109bdacc75:pn-
0x100000109bdacc75 live optimized
 +- nvme32 fc traddr=nn-0x2011d039ea5cfc90:pn-
0x2013d039ea5cfc90,host_traddr=nn-0x200000109bdacc76:pn-
0x100000109bdacc76 live optimized
```

NVMe/TCP

다음 예시 출력은 NVMe/TCP를 사용하는 AFF, FAS, ASA 시스템 및 ASA r2 시스템을 위한 2노드 ONTAP 컨트롤러에서 호스팅되는 네임스페이스를 보여줍니다.

AFF, FAS 및 ASA 예시 출력 표시

```
nvme-subsys9 - NQN=nqn.1992-
08.com.netapp:sn.9927e165694211f0b4f4d039eab31e9d:subsystem.nvme
10
               hostnqn=nqn.2014-08.org.nvmexpress:uuid:4c4c4544-
0035-5910-804b-b7c04f444d33
\
 +- nvme105 tcp
traddr=192.168.39.10,trsvcid=4420,host_traddr=192.168.39.20,src_
addr=192.168.39.20 live optimized
 +- nvme153 tcp
traddr=192.168.39.11,trsvcid=4420,host_traddr=192.168.39.20,src_
addr=192.168.39.20 live non-optimized
 +- nvme57 tcp
traddr=192.168.38.11,trsvcid=4420,host_traddr=192.168.38.20,src_
addr=192.168.38.20 live non-optimized
 +- nvme9 tcp
traddr=192.168.38.10,trsvcid=4420,host_traddr=192.168.38.20,src_
addr=192.168.38.20 live optimized
```

ASA r2 예제 출력 표시

```
nvme-subsys4 - NQN=nqn.1992-
08.com.netapp:sn.17e32b6e8c7f11f09545d039eac03c33:subsystem.Bidi
rectional_DHCP_1_0
               hostnqn=nqn.2014-08.org.nvmexpress:uuid:4c4c4544-
0054-5110-8039-c3c04f523034
\
+- nvme4 tcp
traddr=192.168.20.28,trsvcid=4420,host_traddr=192.168.20.21,src_
addr=192.168.20.21 live optimized
+- nvme5 tcp
traddr=192.168.20.29,trsvcid=4420,host_traddr=192.168.20.21,src_
addr=192.168.20.21 live optimized
+- nvme6 tcp
traddr=192.168.21.28,trsvcid=4420,host_traddr=192.168.21.21,src_
addr=192.168.21.21 live optimized
+- nvme7 tcp
traddr=192.168.21.29,trsvcid=4420,host_traddr=192.168.21.21,src_
addr=192.168.21.21 live optimized
```

5. NetApp 플러그인에 각 ONTAP 네임스페이스 장치에 대한 올바른 값이 표시되는지 확인합니다.

열

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

예제 보기

```
Device           Vserver                   Namespace Path
NSID UUID                                   Size
---------------- ------------------------- ------------------
---- ---------------------------------------- ---------
/dev/nvme0n1     vs_coexistence_emulex     ns1                1
79510f05-7784-11f0-b3c2-d039ea5cfc91   21.47GB
```

JSON을 참조하십시오

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

예제 보기

```
{
"ONTAPdevices":[{
      "Device":"/dev/nvme0n1",
      "Vserver":"vs_coexistence_emulex",
      "Namespace_Path":"ns1",
      "NSID":1,
      "UUID":"79510f05-7784-11f0-b3c2-d039ea5cfc91",
      "Size":"21.47GB",
      "LBA_Data_Size":4096,
      "Namespace_Size":5242880
    }  ]
}
```

8단계: 지속적인 검색 컨트롤러 만들기

SUSE Linux Enterprise Server 16 호스트용 영구 검색 컨트롤러(PDC)를 생성할 수 있습니다. PDC는 NVMe 서브시스템 추가 또는 제거 작업과 검색 로그 페이지 데이터 변경 사항을 자동으로 감지하는 데 필요합니다.

단계

1. 검색 로그 페이지 데이터를 사용할 수 있고 이니시에이터 포트와 타겟 LIF 조합을 통해 검색할 수 있는지 확인합니다.

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

예제 출력을 표시합니다

```
Discovery Log Number of Records 8, Generation counter 10
=====Discovery Log Entry 0======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  3
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.9927e165694211f0b4f4d039eab31e9d:discovery
traddr:  192.168.39.10
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.9927e165694211f0b4f4d039eab31e9d:discovery
traddr:  192.168.38.10
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:  4
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.9927e165694211f0b4f4d039eab31e9d:discovery
traddr:  192.168.39.11
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:  2
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.9927e165694211f0b4f4d039eab31e9d:discovery
traddr:  192.168.38.11
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 4======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  3
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.9927e165694211f0b4f4d039eab31e9d:subsystem.nvme1
traddr:  192.168.39.10
eflags:  none
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.9927e165694211f0b4f4d039eab31e9d:subsystem.nvme1
traddr:  192.168.38.10
eflags:  none
sectype: none
=====Discovery Log Entry 6======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  4
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.9927e165694211f0b4f4d039eab31e9d:subsystem.nvme1
traddr:  192.168.39.11
eflags:  none
sectype: none
```

```
=====Discovery Log Entry 7======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.9927e165694211f0b4f4d039eab31e9d:subsystem.nvme1
traddr:  192.168.38.11
eflags:  none
sectype: none
```

2. 검색 하위 시스템에 대한 PDC 생성:

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

다음과 같은 출력이 표시됩니다.

```
nvme discover -t tcp -w 192.168.39.20 -a 192.168.39.11 -p
```

3. ONTAP 컨트롤러에서 PDC가 생성되었는지 확인합니다.

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

예제 출력을 표시합니다

```
vserver nvme show-discovery-controller -instance -vserver
vs_tcp_sles16
Vserver Name: vs_tcp_sles16
               Controller ID: 0180h
     Discovery Subsystem NQN: nqn.1992-
08.com.netapp:sn.9927e165694211f0b4f4d039eab31e9d:discovery
           Logical Interface: lif3
                        Node: A400-12-171
                   Host NQN: nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0035-5910-804b-b7c04f444d33
          Transport Protocol: nvme-tcp
 Initiator Transport Address: 192.168.39.20
Transport Service Identifier: 8009
             Host Identifier: 4c4c454400355910804bb7c04f444d33
           Admin Queue Depth: 32
       Header Digest Enabled: false
         Data Digest Enabled: false
   Keep-Alive Timeout (msec): 30000
```

9단계: 안전한 인밴드 인증 설정

SUSE Linux Enterprise Server 16 호스트와 ONTAP 컨트롤러 간에 NVMe/TCP를 통한 보안 인밴드 인증이 지원됩니다.

각 호스트 또는 컨트롤러는 다음 항목과 연결되어야 합니다. `DH-HMAC-CHAP` 안전한 인증을 설정하는 핵심입니다. DH-HMAC-CHAP 키는 NVMe 호스트 또는 컨트롤러의 NQN과 관리자가 구성한 인증 비밀 키의 조합입니다. NVMe 호스트 또는 컨트롤러는 피어를 인증하기 위해 피어와 연결된 키를 인식해야 합니다.

단계

CLI나 구성 JSON 파일을 사용하여 안전한 인밴드 인증을 설정합니다. 서로 다른 하위 시스템에 대해 다른 dhchap 키를 지정해야 하는 경우 구성 JSON 파일을 사용해야 합니다.

CLI를 참조하십시오

CLI를 사용하여 보안 인밴드 인증을 설정합니다.

1. 호스트 NQN 가져오기:

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

2. 호스트에 대한 dhchap 키를 생성합니다.

   다음 출력에서는 명령 매개 변수에 대해 `gen-dhchap-key` 설명합니다.

```
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
```

   다음 예에서는 HMAC이 3(SHA-512)으로 설정된 임의의 dhchap 키가 생성됩니다.

```
nvme gen-dhchap-key -m 3 -n nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0035-5910-804b-b7c04f444d33
DHHC-
1:03:ohdxI1yIS8gBLwIOubcwl57rXcozYuRgBsoWaBvxEvpDlQHn/7dQ4JjFGwmhgwd
JWmVoripbWbMJy5eMAbCahN4hhYU=:
```

3. ONTAP 컨트롤러에서 호스트를 추가하고 두 dhchap 키를 모두 지정합니다.

```
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. 호스트는 단방향 및 양방향이라는 두 가지 유형의 인증 방법을 지원합니다. 호스트에서 ONTAP 컨트롤러에 연결하고 선택한 인증 방법에 따라 dhchap 키를 지정합니다.

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

5. 의 유효성을 검사합니다 `nvme connect authentication` 호스트 및 컨트롤러 dhchap 키를 확인하여 명령:

   a. 호스트 dhchap 키를 확인합니다.

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

   단방향 설정에 대한 출력 예제를 표시합니다

   ```
   # cat /sys/class/nvme-subsystem/nvme-
   subsys1/nvme*/dhchap_secret
   DHHC-1:01:wkwAKk8r9Ip7qECKt7V5aIo/7Y1CH7DWkUfLfMxmseg39DFb:
   DHHC-1:01:wkwAKk8r9Ip7qECKt7V5aIo/7Y1CH7DWkUfLfMxmseg39DFb:
   DHHC-1:01:wkwAKk8r9Ip7qECKt7V5aIo/7Y1CH7DWkUfLfMxmseg39DFb:
   DHHC-1:01:wkwAKk8r9Ip7qECKt7V5aIo/7Y1CH7DWkUfLfMxmseg39DFb:
   ```

   b. 컨트롤러 dhchap 키를 확인합니다.

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

에는 양방향 구성의 출력 예가 나와 있습니다

```
# cat /sys/class/nvme-subsystem/nvme-
subsys6/nvme*/dhchap_ctrl_secret
DHHC-
1:03:ohdxI1yIS8gBLwIOubcwl57rXcozYuRgBsoWaBvxEvpDlQHn/7dQ4JjFG
wmhgwdJWmVoripbWbMJy5eMAbCahN4hhYU=:
DHHC-
1:03:ohdxI1yIS8gBLwIOubcwl57rXcozYuRgBsoWaBvxEvpDlQHn/7dQ4JjFG
wmhgwdJWmVoripbWbMJy5eMAbCahN4hhYU=:
DHHC-
1:03:ohdxI1yIS8gBLwIOubcwl57rXcozYuRgBsoWaBvxEvpDlQHn/7dQ4JjFG
wmhgwdJWmVoripbWbMJy5eMAbCahN4hhYU=:
DHHC-
1:03:ohdxI1yIS8gBLwIOubcwl57rXcozYuRgBsoWaBvxEvpDlQHn/7dQ4JjFG
wmhgwdJWmVoripbWbMJy5eMAbCahN4hhYU=:
```

JSON을 참조하십시오

ONTAP 컨트롤러 구성에서 여러 NVMe 서브시스템을 사용할 수 있는 경우 파일을 명령과 함께 `nvme connect-all` 사용할 수 `/etc/nvme/config.json` 있습니다.

사용하다 `-o` JSON 파일을 생성하는 옵션입니다. 더 많은 구문 옵션은 NVMe connect-all 매뉴얼 페이지를 참조하세요.

1. JSON 파일 구성:

예제 출력을 표시합니다

```
# cat /etc/nvme/config.json
[
  {
    "hostnqn":"nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0035-
5910-804b-b7c04f444d33",
    "hostid":"4c4c4544-0035-5910-804b-b7c04f444d33",
    "dhchap_key":"DHHC-
1:01:wkwAKk8r9Ip7qECKt7V5aIo/7Y1CH7DWkUfLfMxmseg39DFb:",
    "subsystems":[
      {
        "nqn":"nqn.1992-
08.com.netapp:sn.9927e165694211f0b4f4d039eab31e9d:subsystem.inba
nd_bidirectional",
        "ports":[
          {
            "transport":"tcp",
            "traddr":"192.168.38.10",
            "host_traddr":"192.168.38.20",
            "trsvcid":"4420",
            "dhchap_ctrl_key":"DHHC-
1:03:ohdxI1yIS8gBLwIOubcwl57rXcozYuRgBsoWaBvxEvpDlQHn/7dQ4JjFGwm
hgwdJWmVoripbWbMJy5eMAbCahN4hhYU=:"
          },
          {
            "transport":"tcp",
            "traddr":"192.168.38.11",
            "host_traddr":"192.168.38.20",
            "trsvcid":"4420",
            "dhchap_ctrl_key":"DHHC-
1:03:ohdxI1yIS8gBLwIOubcwl57rXcozYuRgBsoWaBvxEvpDlQHn/7dQ4JjFGwm
hgwdJWmVoripbWbMJy5eMAbCahN4hhYU=:"
          },
          {
            "transport":"tcp",
            "traddr":"192.168.39.11",
            "host_traddr":"192.168.39.20",
            "trsvcid":"4420",
            "dhchap_ctrl_key":"DHHC-
1:03:ohdxI1yIS8gBLwIOubcwl57rXcozYuRgBsoWaBvxEvpDlQHn/7dQ4JjFGwm
hgwdJWmVoripbWbMJy5eMAbCahN4hhYU=:"
          },
          {
            "transport":"tcp",
```

```
            "traddr":"192.168.39.10",
            "host_traddr":"192.168.39.20",
            "trsvcid":"4420",
            "dhchap_ctrl_key":"DHHC-
1:03:ohdxI1yIS8gBLwIOubcwl57rXcozYuRgBsoWaBvxEvpDlQHn/7dQ4JjFGwm
hgwdJWmVoripbWbMJy5eMAbCahN4hhYU=:"
          }
        ]
      }
    ]
  }
]
```

다음 예에서, `dhchap_key` 에 대응하다 `dhchap_secret` 그리고 `dhchap_ctrl_key` 에 대응하다 `dhchap_ctrl_secret` .

2. config JSON 파일을 사용하여 ONTAP 컨트롤러에 연결합니다.

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

예제 출력을 표시합니다

```
traddr=192.168.38.10is already connected
traddr=192.168.39.10 is already connected
traddr=192.168.38.11 is already connected
traddr=192.168.39.11 is already connected
traddr=192.168.38.10is already connected
traddr=192.168.39.10 is already connected
traddr=192.168.38.11 is already connected
traddr=192.168.39.11 is already connected
traddr=192.168.38.10is already connected
traddr=192.168.39.10 is already connected
traddr=192.168.38.11 is already connected
traddr=192.168.39.11 is already connected
```

3. 각 하위 시스템에 대해 해당 컨트롤러에 대해 dhchap 암호가 활성화되어 있는지 확인합니다.

   a. 호스트 dhchap 키를 확인합니다.

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

다음 예에서는 dhchap 키를 보여줍니다.

```
DHHC-1:01:wkwAKk8r9Ip7qECKt7V5aIo/7Y1CH7DWkUfLfMxmseg39DFb:
```

b. 컨트롤러 dhchap 키를 확인합니다.

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

다음 예와 비슷한 출력이 표시되어야 합니다.

```
DHHC-
1:03:ohdxI1yIS8gBLwIOubcwl57rXcozYuRgBsoWaBvxEvpDlQHn/7dQ4JjFGwmhgwd
JWmVoripbWbMJy5eMAbCahN4hhYU=:
```

10단계: 전송 계층 보안 구성

전송 계층 보안(TLS)은 NVMe-oF 호스트와 ONTAP 어레이 간의 NVMe 연결에 대해 안전한 종단 간 암호화를 제공합니다. CLI와 구성된 사전 공유 키(PSK)를 사용하여 TLS 1.3을 구성할 수 있습니다.

ONTAP 컨트롤러에서 단계를 수행하도록 지정된 경우를 제외하고, 다음 단계를 SUSE Linux Enterprise Server 호스트에서 수행하십시오.

단계

1. 다음 사항이 있는지 확인하세요. `ktls-utils` , `openssl` , 그리고 `libopenssl` 호스트에 설치된 패키지:

a. 확인하다 `ktls-utils` :

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

다음과 같은 출력이 표시됩니다.

```
ktls-utils-0.10+33.g311d943-160000.2.2.x86_64
```

a. SSL 패키지를 확인하세요:

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

예제 출력을 표시합니다

```
libopenssl3-3.5.0-160000.3.2.x86_64
openssl-3.5.0-160000.2.2.noarch
openssl-3-3.5.0-160000.3.2.x86_64
libopenssl3-x86-64-v3-3.5.0-160000.3.2.x86_64
```

2. 다음에 대해 올바르게 설정되어 있는지 확인합니다 `/etc/tlshd.conf`.

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

예제 출력을 표시합니다

```
[debug]
loglevel=0
tls=0
nl=0

[authenticate]
#keyrings= <keyring>;<keyring>;<keyring>

[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. 시스템 부팅 시 시작 활성화 `tlshd`:

```
systemctl enable tlshd
```

4. 데몬이 실행 중인지 `tlshd` 확인합니다.

```
systemctl status tlshd
```

예제 출력을 표시합니다

```
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. 다음을 사용하여 TLS PSK를 `nvme gen-tls-key` 생성합니다.

   a. 호스트를 확인하세요:

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

   다음과 같은 출력이 표시됩니다.

   ```
   nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0035-5910-804b-b7c04f444d33
   ```

   b. 키를 확인하세요:

   ```
   nvme gen-tls-key --hmac=1 --identity=1 --subsysnqn= nqn.1992-
   08.com.netapp:sn.9927e165694211f0b4f4d039eab31e9d:subsystem.nvme1
   ```

   다음과 같은 출력이 표시됩니다.

```
NVMeTLSkey-1:01:C50EsaGtuOp8n5fGE9EuWjbBCtshmfoHx4XTqTJUmydf0gIj:
```

6. ONTAP 컨트롤러에서 TLS PSK를 ONTAP 하위 시스템에 추가합니다.

예제 출력을 표시합니다

```
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 PSK를 호스트 커널 키링에 삽입합니다.

```
nvme check-tls-key --identity=1 --subsysnqn=nqn.1992
-08.com.netapp:sn.9927e165694211f0b4f4d039eab31e9d:subsystem.nvme1
--keydata=NVMeTLSkey
-1:01:C50EsaGtuOp8n5fGE9EuWjbBCtshmfoHx4XTqTJUmydf0gIj: --insert
```

다음 TLS 키가 표시되어야 합니다.

```
Inserted TLS key 069f56bb
```

PSK는 다음과 같이 표시됩니다. `NVMe1R01` 왜냐하면 그것을 사용하기 때문이다 `identity v1` TLS 핸드셰이크 알고리즘에서. Identity v1은 ONTAP에서 지원하는 유일한 버전입니다.

8. TLS PSK가 올바르게 삽입되었는지 확인합니다.

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

예제 출력을 표시합니다

```
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.9927e165694211f0b4f4d039eab31e9d:subsystem.nvme1
oYVLelmiOwnvDjXKBmrnIgGVpFIBDJtc4hmQXE/36Sw=: 32
```

9. 삽입된 TLS PSK를 사용하여 ONTAP 하위 시스템에 연결합니다.

   a. TLS PSK를 확인하세요.

```
nvme connect -t tcp -w 192.168.38.20 -a 192.168.38.10  -n nqn.1992-
08.com.netapp:sn.9927e165694211f0b4f4d039eab31e9d:subsystem.nvme1
--tls_key=0x069f56bb –tls
```

다음과 같은 출력이 표시됩니다.

```
connecting to device: nvme0
```

a. list-subsys를 확인하세요:

```
nvme list-subsys
```

예제 출력을 표시합니다

```
nvme-subsys0 - NQN=nqn.1992-
08.com.netapp:sn.9927e165694211f0b4f4d039eab31e9d:subsystem.nvme1
               hostnqn=nqn.2014-08.org.nvmexpress:uuid:4c4c4544-
0035-5910-804b-b2c04f444d33
**
 +- nvme0 tcp
traddr=192.168.38.10,trsvcid=4420,host_traddr=192.168.38.20,src_a
ddr=192.168.38.20 live
```

10. 대상을 추가하고 지정된 ONTAP 하위 시스템에 대한 TLS 연결을 확인합니다.

```
nvme subsystem controller show -vserver vs_tcp_sles16 -subsystem nvme1
-instance
```

예제 출력을 표시합니다

```
(vserver nvme subsystem controller show)
                         Vserver Name: vs_tcp_sles16
                            Subsystem: nvme1
                        Controller ID: 0040h
                    Logical Interface: lif1
                                 Node: A400-12-171
                             Host NQN: nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0035-5910-804b-b2c04f444d33
                   Transport Protocol: nvme-tcp
          Initiator Transport Address: 192.168.38.20
                      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: 62203cfd-826a-11f0-966e-
d039eab31e9d
                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.9927e165694211f0b4f4d039eab31e9d:subsystem.nvme1
oYVLelmiOwnvDjXKBmrnIgGVpFIBDJtc4hmQXE/36Sw=
                          TLS Cipher: TLS-AES-128-GCM-SHA256
```

11단계: 알려진 문제를 검토합니다

알려진 문제가 없습니다.

ONTAP 스토리지를 사용하여 NVMe-oF를 구성하기 위한 SUSE Linux Enterprise Server 15 SPx 구성

SUSE Linux Enterprise Server 15 SPx 호스트는 비대칭 네임스페이스 액세스(ANA)를

지원하는 NVMe over Fibre Channel(NVMe/FC) 및 NVMe over TCP(NVMe/TCP) 프로토콜을 지원합니다. ANA는 iSCSI 및 FCP 환경의 비대칭 논리 장치 액세스(ALUA)와 동일한 멀티패싱 기능을 제공합니다.

SUSE Linux Enterprise Server 15 SPx용 NVMe-oF(NVMe over Fabrics) 호스트를 구성하는 방법을 알아보세요. 더 자세한 지원 및 기능 정보는 다음을 참조하세요. "ONTAP 지원 및 기능".

SUSE Linux Enterprise Server 15 SPx에서 NVMe-oF를 사용할 때 다음과 같은 알려진 제한 사항이 있습니다.

- 그만큼 `nvme disconnect-all` 이 명령을 실행하면 루트와 데이터 파일 시스템의 연결이 모두 끊어지고 시스템이 불안정해질 수 있습니다. NVMe-TCP 또는 NVMe-FC 네임스페이스를 통해 SAN에서 부팅하는 시스템에서는 이 명령을 실행하지 마세요.
- NetApp sanlun 호스트 유틸리티는 NVMe-oF를 지원하지 않습니다. 대신, 기본 제공되는 NetApp 플러그인을 사용할 수 있습니다. `nvme-cli` 모든 NVMe-oF 전송을 위한 패키지입니다.
- SUSE Linux Enterprise Server 15 SP6 이하 버전에서는 NVMe-oF 프로토콜을 사용한 SAN 부팅이 지원되지 않습니다.

1단계: 필요에 따라 SAN 부팅을 활성화합니다

SAN 부팅을 사용하도록 호스트를 구성하여 배포를 간소화하고 확장성을 개선할 수 있습니다. 사용하다"상호 운용성 매트릭스 툴" Linux OS, 호스트 버스 어댑터(HBA), HBA 펌웨어, HBA 부팅 BIOS 및 ONTAP 버전이 SAN 부팅을 지원하는지 확인하세요.

단계

1. "NVMe 네임스페이스를 생성하고 호스트에 매핑합니다." .
2. SAN 부팅 네임스페이스가 매핑된 포트에 대해 서버 BIOS에서 SAN 부팅을 활성화합니다.

   HBA BIOS를 활성화하는 방법에 대한 자세한 내용은 공급업체별 설명서를 참조하십시오.
3. 호스트를 재부팅하고 OS가 제대로 실행 중인지 확인하세요.

2단계: SUSE Linux Enterprise Server 및 NVMe 소프트웨어를 설치하고 구성을 확인합니다.

NVMe-oF를 사용하도록 호스트를 구성하려면 호스트 및 NVMe 소프트웨어 패키지를 설치하고, 멀티패싱을 활성화하고, 호스트의 NQN 구성을 확인해야 합니다.

단계

1. 서버에 SUSE Linux Enterprise Server 15 SPx를 설치하십시오. 설치가 완료되면 지정된 SUSE Linux Enterprise Server 15 SPx 커널이 실행 중인지 확인하십시오.

   ```
   uname -r
   ```

   Rocky Linux 커널 버전 예:

   ```
   6.4.0-150700.53.3-default
   ```

2. "NVMe-CLI" 패키지를 설치합니다.

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

다음 예에서는 다음을 보여줍니다. `nvme-cli` 패키지 버전:

```
nvme-cli-2.11+22.gd31b1a01-150700.3.3.2.x86_64
```

3. 를 설치합니다 `libnvme` 패키지:

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

다음 예에서는 다음을 보여줍니다. `libnvme` 패키지 버전:

```
libnvme1-1.11+4.ge68a91ae-150700.4.3.2.x86_64
```

4. 호스트에서 hostnqn 문자열을 확인하세요. `/etc/nvme/hostnqn` :

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

다음 예에서는 다음을 보여줍니다. `hostnqn` 버전:

```
nqn.2014-08.org.nvmexpress:uuid:f6517cae-3133-11e8-bbff-7ed30aef123f
```

5. ONTAP 시스템에서 다음 사항을 확인하십시오. `hostnqn` 문자열이 일치합니다 `hostnqn` ONTAP 어레이의 해당 서브시스템에 대한 문자열:

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

예제 보기

```
Vserver Subsystem Priority  Host NQN
------- --------- --------
-------------------------------------------------
vs_coexistence_LPE36002
        nvme
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
        nvme_1
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
        nvme_2
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
        nvme_3
                  regular   nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
4 entries were displayed.
```

를 누릅니다 `hostnqn` 문자열이 일치하지 않습니다. 를 사용하십시오 `vserver modify` 명령을 사용하여 를 업데이트합니다 `hostnqn` 와 일치하는 해당 ONTAP 배열 하위 시스템의 문자열입니다 `hostnqn` 문자열 시작 `/etc/nvme/hostnqn` 호스트.

3단계: NVMe/FC 및 NVMe/TCP 구성

Broadcom/Emulex 또는 Marvell/QLogic 어댑터를 사용하여 NVMe/FC를 구성하거나 수동 검색 및 연결 작업을 사용하여 NVMe/TCP를 구성합니다.

NVMe/FC - Broadcom/Emulex

Broadcom/Emulex FC 어댑터용 NVMe/FC를 구성합니다.

단계

1. 지원되는 어댑터 모델을 사용하고 있는지 확인합니다.

   a. 모델 이름을 표시합니다:

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

   다음과 같은 출력이 표시됩니다.

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

   b. 모델 설명을 표시합니다.

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

   다음과 같은 출력이 표시됩니다.

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

2. 권장 Broadcom을 사용하고 있는지 확인합니다 `lpfc` 펌웨어 및 받은 편지함 드라이버:

   a. 펌웨어 버전을 표시합니다.

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

   다음 예에서는 펌웨어 버전을 보여줍니다.

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

   b. 받은 편지함 드라이버 버전을 표시합니다.

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

다음 예에서는 드라이버 버전을 보여줍니다.

```
0:14.4.0.8
```

지원되는 어댑터 드라이버 및 펌웨어 버전의 현재 목록은 를 참조하십시오"상호 운용성 매트릭스 툴".

3. 의 예상 출력이 3 다음과 같이 설정되었는지 확인합니다 `lpfc_enable_fc4_type`.

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

4. 이니시에이터 포트를 볼 수 있는지 확인합니다.

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

다음과 유사한 출력이 표시됩니다.

```
0x10000090fae0ec88
0x10000090fae0ec89
```

5. 이니시에이터 포트가 온라인 상태인지 확인합니다.

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

다음과 같은 출력이 표시됩니다.

```
Online
Online
```

6. NVMe/FC 이니시에이터 포트가 활성화되었고 타겟 포트가 표시되는지 확인합니다.

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

예제 출력을 표시합니다

```
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 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 lpfc1 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc1 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
fffffffffffffffd
        abort 0000018a noxri 00000000 nondlp 0000012e qdepth
00000000 wqerr 00000004 err 00000000
FCP CMPL: xb 0000018a Err 000005ca
```

NVMe/FC - Marvell/QLogic

Marvell/QLogic 어댑터용 NVMe/FC를 구성합니다.

단계

1. 지원되는 어댑터 드라이버 및 펌웨어 버전을 실행하고 있는지 확인합니다.

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

다음 예에서는 드라이버 및 펌웨어 버전을 보여줍니다.

```
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. 확인합니다 `ql2xnvmeenable` 가 설정됩니다. 그러면 Marvell 어댑터가 NVMe/FC Initiator로 작동할 수 있습니다.

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

예상 출력은 1입니다.

NVMe/TCP

NVMe/TCP 프로토콜은 자동 연결 작업을 지원하지 않습니다. 대신 NVMe/TCP를 수행하여 NVMe/TCP 하위 시스템과 네임스페이스를 검색할 수 있습니다. `connect` 또는 `connect-all` 수동으로 작업합니다.

단계

1. 이니시에이터 포트가 지원되는 NVMe/TCP LIF에서 검색 로그 페이지 데이터를 가져올 수 있는지 확인합니다.

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

예제 출력을 표시합니다

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

2. 다른 모든 NVMe/TCP 이니시에이터-타겟 LIF 조합이 검색 로그 페이지 데이터를 성공적으로 가져올 수 있는지 확인합니다.

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

예제 보기

```
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. 를 실행합니다 `nvme connect-all` 노드에 걸쳐 지원되는 모든 NVMe/TCP 이니시에이터-타겟 LIF에 대한 명령:

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

예제 보기

```
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 SP6부터 NVMe/TCP 설정이 변경되었습니다. `ctrl_loss_tmo timeout` 자동으로 "꺼짐"으로 설정됩니다. 결과적으로:

- 재시도 횟수에 제한이 없습니다(무기한 재시도).
- 특정 항목을 수동으로 구성할 필요가 없습니다. `ctrl_loss_tmo timeout` 사용 시 지속 시간 `nvme connect` 또는 `nvme connect-all` 명령어(옵션 -l).
- NVMe/TCP 컨트롤러는 경로 장애가 발생해도 시간 초과가 발생하지 않으며 무기한 연결 상태를 유지합니다.

4단계: 선택적으로 **udev** 규칙에서 **iopolicy**를 수정합니다.

SUSE Linux Enterprise Server 15 SP6부터 NVMe-oF의 기본 iopolicy는 다음과 같이 설정됩니다. `round-robin`. iopolicy를 변경하려면 `queue-depth`udev 규칙 파일을 다음과 같이 수정하십시오.

단계

1. 루트 권한으로 텍스트 편집기에서 udev 규칙 파일을 엽니다.

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

   다음과 같은 출력이 표시됩니다.

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

2. 다음 예시 규칙에 나와 있는 것처럼 NetApp ONTAP 컨트롤러의 iopolicy를 설정하는 줄을 찾으십시오.

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

3. 규칙을 다음과 같이 수정하세요. `round-robin` 된다 `queue-depth` :

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

4. udev 규칙을 다시 로드하고 변경 사항을 적용합니다.

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

5. 하위 시스템의 현재 iopolicy를 확인하세요. 예를 들어 <하위 시스템>을 다음과 같이 바꾸세요. `nvme-subsys0` .

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

다음과 같은 출력이 표시됩니다.

```
queue-depth.
```

새로운 iopolicy는 일치하는 NetApp ONTAP 컨트롤러 장치에 자동으로 적용됩니다. 재부팅할 필요가 없습니다.

5단계: 선택적으로 NVMe/FC에 대해 1MB I/O를 활성화합니다.

ONTAP Identify Controller 데이터에서 최대 데이터 전송 크기(MDTS)를 8로 보고합니다. 즉, 최대 I/O 요청 크기는 1MB까지 가능합니다. Broadcom NVMe/FC 호스트에 대해 1MB 크기의 I/O 요청을 발행하려면 다음을 늘려야 합니다. `lpfc` 의 가치 `lpfc_sg_seg_cnt` 매개변수를 기본값 64에서 256으로 변경합니다.

이 단계는 Qlogic NVMe/FC 호스트에는 적용되지 않습니다.

단계

1. `lpfc_sg_seg_cnt`매개변수를 256으로 설정합니다.

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

다음 예와 비슷한 출력이 표시되어야 합니다.

```
options lpfc lpfc_sg_seg_cnt=256
```

2. `dracut -f`명령을 실행하고 호스트를 재부팅합니다.
3. 의 값이 256인지 `lpfc_sg_seg_cnt` 확인합니다.

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

6단계: NVMe 부팅 서비스 확인

그만큼 `nvmefc-boot-connections.service` 그리고 `nvmf-autoconnect.service` NVMe/FC에 포함된 부팅 서비스 `nvme-cli` 패키지는 시스템이 부팅될 때 자동으로 활성화됩니다.

부팅이 완료된 후 다음을 확인하세요. `nvmefc-boot-connections.service` 그리고 `nvmf-autoconnect.service` 부팅 서비스가 활성화되었습니다.

단계

1. 가 활성화되어 있는지 `nvmf-autoconnect.service` 확인합니다.

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

예제 출력을 표시합니다

```
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. 가 활성화되어 있는지 `nvmefc-boot-connections.service` 확인합니다.

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

예제 출력을 표시합니다

```
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.
```

7단계: 다중 경로 구성 확인

커널 내 NVMe 다중 경로 상태, ANA 상태 및 ONTAP 네임스페이스가 NVMe-oF 구성에 적합한지 확인합니다.

단계

1. in-kernel NVMe multipath가 활성화되어 있는지 확인합니다.

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

다음과 같은 출력이 표시됩니다.

```
Y
```

2. 해당 ONTAP 네임스페이스에 대한 적절한 NVMe-oF 설정(예: 모델이 NetApp ONTAP 컨트롤러로 설정되고 로드 밸런싱 iopolicy가 queue-depth로 설정됨)이 호스트에 올바르게 반영되었는지 확인하십시오.

   a. 하위 시스템을 표시합니다.

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

다음과 같은 출력이 표시됩니다.

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

b. 정책을 표시합니다.

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

다음과 같은 출력이 표시됩니다.

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

3. 호스트에서 네임스페이스가 생성되고 올바르게 검색되는지 확인합니다.

```
nvme list
```

예제 보기

```
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. 각 경로의 컨트롤러 상태가 라이브이고 올바른 ANA 상태인지 확인합니다.

NVMe/FC

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

예제 출력을 표시합니다

```
nvme-subsys114 - NQN=nqn.1992-
08.com.netapp:sn.9e30b9760a4911f08c87d039eab67a95:subsystem.sles
_161_27
                 hostnqn=nqn.2014-
08.org.nvmexpress: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/TCP

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

예제 출력을 표시합니다

```
nvme-subsys9 - NQN=nqn.1992-
08.com.netapp:sn.f8e2af201b7211f0ac2bd039eab67a95:subsystem.with
_inband_with_json hostnqn=nqn.2014-
08.org.nvmexpress: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. NetApp 플러그인에 각 ONTAP 네임스페이스 장치에 대한 올바른 값이 표시되는지 확인합니다.

열

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

예제 보기

```
Device            Vserver                    Namespace Path
NSID UUID                                     Size
----------------- -------------------------
------------------------------------------------------ ----
---------------------------------------- ---------
/dev/nvme0n1      vs_161
/vol/fc_nvme_vol1/fc_nvme_ns1                        1
32fd92c7-0797-428e-a577-fdb3f14d0dc3   5.37GB
```

JSON을 참조하십시오

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

예제 보기

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

8단계: 지속적인 검색 컨트롤러 만들기

SUSE Linux Enterprise Server 15 SPx 호스트용 영구 검색 컨트롤러(PDC)를 생성할 수 있습니다. PDC는 NVMe 서브시스템 추가 또는 제거 작업과 검색 로그 페이지 데이터 변경 사항을 자동으로 감지하는 데 필요합니다.

단계

1. 검색 로그 페이지 데이터를 사용할 수 있고 이니시에이터 포트와 타겟 LIF 조합을 통해 검색할 수 있는지 확인합니다.

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

예제 출력을 표시합니다

```
Discovery Log Number of Records 8, Generation counter 18
=====Discovery Log Entry 0======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  4
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.4f7af2bd221811f0afadd039eab0dadd:discovery
traddr:  192.168.111.66
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 1======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  2
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.4f7af2bd221811f0afadd039eab0dadd:discovery
traddr:  192.168.211.66
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 2======
trtype:  tcp
adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  3
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.4f7af2bd221811f0afadd039eab0dadd:discovery
traddr:  192.168.111.67
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.4f7af2bd221811f0afadd039eab0dadd:discovery
traddr:  192.168.211.67
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.4f7af2bd221811f0afadd039eab0dadd:subsystem.pdc
traddr:  192.168.111.66
eflags:  none
sectype: none
=====Discovery Log Entry 5======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  2
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.4f7af2bd221811f0afadd039eab0dadd:subsystem.pdc
traddr:  192.168.211.66
eflags:  none
sectype: none
=====Discovery Log Entry 6======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  3
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.4f7af2bd221811f0afadd039eab0dadd:subsystem.pdc
traddr:  192.168.111.67
eflags:  none
sectype: none
```

```
=====Discovery Log Entry 7======
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.4f7af2bd221811f0afadd039eab0dadd:subsystem.pdc
traddr:  192.168.211.67
eflags:  none
sectype: none
```

2. 검색 하위 시스템에 대한 PDC 생성:

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

다음과 같은 출력이 표시됩니다.

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

3. ONTAP 컨트롤러에서 PDC가 생성되었는지 확인합니다.

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

예제 출력을 표시합니다

```
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
```

9단계: 안전한 인밴드 인증 설정

SUSE Linux Enterprise Server 15 SPx 호스트와 ONTAP 컨트롤러 간에 NVMe/TCP를 통한 안전한 대역 내 인증이 지원됩니다.

각 호스트 또는 컨트롤러는 다음 항목과 연결되어야 합니다. `DH-HMAC-CHAP` 안전한 인증을 설정하는 핵심입니다. DH-HMAC-CHAP 키는 NVMe 호스트 또는 컨트롤러의 NQN과 관리자가 구성한 인증 비밀 키의 조합입니다. NVMe 호스트 또는 컨트롤러는 피어를 인증하기 위해 피어와 연결된 키를 인식해야 합니다.

단계

CLI나 구성 JSON 파일을 사용하여 안전한 인밴드 인증을 설정합니다. 서로 다른 하위 시스템에 대해 다른 dhchap 키를 지정해야 하는 경우 구성 JSON 파일을 사용해야 합니다.

CLI를 참조하십시오

CLI를 사용하여 보안 인밴드 인증을 설정합니다.

1. 호스트 NQN 가져오기:

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

2. 호스트에 대한 dhchap 키를 생성합니다.

   다음 출력에서는 명령 매개 변수에 대해 `gen-dhchap-key` 설명합니다.

```
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
```

   다음 예에서는 HMAC이 3(SHA-512)으로 설정된 임의의 dhchap 키가 생성됩니다.

```
nvme gen-dhchap-key -m 3 -n nqn.2014-
08.org.nvmexpress:uuid:e6dade64-216d-11ec-b7bb-7ed30a5482c3
DHHC-
1:03:1CFivw9ccz58gAcOUJrM7Vs98hd2ZHSr+iw+Amg6xZPl5D2Yk+HDTZiUAg1iGgx
TYqnxukqvYedA55Bw3wtz6sJNpR4=:
```

3. ONTAP 컨트롤러에서 호스트를 추가하고 두 dhchap 키를 모두 지정합니다.

```
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. 호스트는 단방향 및 양방향이라는 두 가지 유형의 인증 방법을 지원합니다. 호스트에서 ONTAP 컨트롤러에 연결하고 선택한 인증 방법에 따라 dhchap 키를 지정합니다.

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

5. 의 유효성을 검사합니다 `nvme connect authentication` 호스트 및 컨트롤러 dhchap 키를 확인하여 명령:

   a. 호스트 dhchap 키를 확인합니다.

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

   단방향 설정에 대한 출력 예제를 표시합니다

   ```
   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. 컨트롤러 dhchap 키를 확인합니다.

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

   에는 양방향 구성의 출력 예가 나와 있습니다

   ```
   cat /sys/class/nvme-subsystem/nvme-
   subsys6/nvme*/dhchap_ctrl_secret
   DHHC-
   1:03:1CFivw9ccz58gAcOUJrM7Vs98hd2ZHSr+iw+Amg6xZPl5D2Yk+HDTZiUA
   g1iGgxTYqnxukqvYedA55Bw3wtz6sJNpR4=:
   DHHC-
   1:03:1CFivw9ccz58gAcOUJrM7Vs98hd2ZHSr+iw+Amg6xZPl5D2Yk+HDTZiUA
   g1iGgxTYqnxukqvYedA55Bw3wtz6sJNpR4=:
   DHHC-
   1:03:1CFivw9ccz58gAcOUJrM7Vs98hd2ZHSr+iw+Amg6xZPl5D2Yk+HDTZiUA
   g1iGgxTYqnxukqvYedA55Bw3wtz6sJNpR4=:
   DHHC-
   1:03:1CFivw9ccz58gAcOUJrM7Vs98hd2ZHSr+iw+Amg6xZPl5D2Yk+HDTZiUA
   g1iGgxTYqnxukqvYedA55Bw3wtz6sJNpR4=:
   ```

JSON을 참조하십시오

ONTAP 컨트롤러 구성에서 여러 NVMe 서브시스템을 사용할 수 있는 경우 파일을 명령과 함께 `nvme connect-all` 사용할 수 `/etc/nvme/config.json` 있습니다.

사용하다 `-o` JSON 파일을 생성하는 옵션입니다. 더 많은 구문 옵션은 NVMe connect-all 매뉴얼 페이지를 참조하세요.

1. JSON 파일 구성:

예제 출력을 표시합니다

```
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.samp
le_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+XqAf2X6twr9ngBpr2n0MGWbmZIZq4PieKZCoilKGef8lAvh
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+XqAf2X6twr9ngBpr2n0MGWbmZIZq4PieKZCoilKGef8lAvh
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+XqAf2X6twr9ngBpr2n0MGWbmZIZq4PieKZCoilKGef8lAvh
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=:"
                }
            ]
        }
    ]
  }
]
```

다음 예에서, `dhchap_key` 에 대응하다 `dhchap_secret` 그리고 `dhchap_ctrl_key` 에 대응하다 `dhchap_ctrl_secret` .

2. config JSON 파일을 사용하여 ONTAP 컨트롤러에 연결합니다.

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

예제 출력을 표시합니다

```
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. 각 하위 시스템에 대해 해당 컨트롤러에 대해 dhchap 암호가 활성화되어 있는지 확인합니다.

   a. 호스트 dhchap 키를 확인합니다.

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

다음 예에서는 dhchap 키를 보여줍니다.

```
DHHC-1:01:i4i789R11sMuHLCY27RVI8XloC/GzjRwyhxip5hmIELsHrBq:
```

b. 컨트롤러 dhchap 키를 확인합니다.

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

다음 예와 비슷한 출력이 표시되어야 합니다.

```
DHHC-
1:03:jqgYcJSKp73+XqAf2X6twr9ngBpr2n0MGWbmZIZq4PieKZCoilKGef8lAvhYS0P
NK7T+04YD5CRPjh+m3qjJU++yR8s=:
```

10단계: 전송 계층 보안 구성

전송 계층 보안(TLS)은 NVMe-oF 호스트와 ONTAP 어레이 간의 NVMe 연결에 대해 안전한 종단 간 암호화를 제공합니다. CLI와 구성된 사전 공유 키(PSK)를 사용하여 TLS 1.3을 구성할 수 있습니다.

ONTAP 컨트롤러에서 단계를 수행하도록 지정된 경우를 제외하고, 다음 단계를 SUSE Linux Enterprise Server 호스트에서 수행하십시오.

단계

1. 다음 사항이 있는지 확인하세요. `ktls-utils` , `openssl` , 그리고 `libopenssl` 호스트에 설치된 패키지:

a. 확인하다 `ktls-utils` :

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

다음과 같은 출력이 표시됩니다.

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

a. SSL 패키지를 확인하세요:

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

예제 출력을 표시합니다

```
libopenssl3-3.2.3-150700.3.20.x86_64
openssl-3-3.2.3-150700.3.20.x86_64
libopenssl1_1-1.1.1w-150700.9.37.x86_64
```

2. 다음에 대해 올바르게 설정되어 있는지 확인합니다 `/etc/tlshd.conf`.

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

예제 출력을 표시합니다

```
[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. 시스템 부팅 시 시작 활성화 `tlshd`:

```
systemctl enable tlshd
```

4. 데몬이 실행 중인지 `tlshd` 확인합니다.

```
systemctl status tlshd
```

예제 출력을 표시합니다

```
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. 다음을 사용하여 TLS PSK를 `nvme gen-tls-key` 생성합니다.

   a. 호스트를 확인하세요:

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

   다음과 같은 출력이 표시됩니다.

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

   b. 키를 확인하세요:

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

   다음과 같은 출력이 표시됩니다.

```
NVMeTLSkey-1:01:C50EsaGtuOp8n5fGE9EuWjbBCtshmfoHx4XTqTJUmydf0gIj:
```

6. ONTAP 컨트롤러에서 TLS PSK를 ONTAP 하위 시스템에 추가합니다.

예제 출력을 표시합니다

```
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 PSK를 호스트 커널 키링에 삽입합니다.

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

다음 TLS 키가 표시되어야 합니다.

```
Inserted TLS key 22152a7e
```

PSK는 다음과 같이 표시됩니다. `NVMe1R01` 왜냐하면 그것을 사용하기 때문이다 `identity v1` TLS 핸드셰이크 알고리즘에서. Identity v1은 ONTAP에서 지원하는 유일한 버전입니다.

8. TLS PSK가 올바르게 삽입되었는지 확인합니다.

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

예제 출력을 표시합니다

```
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. 삽입된 TLS PSK를 사용하여 ONTAP 하위 시스템에 연결합니다.

   a. TLS PSK를 확인하세요.

```
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
```

다음과 같은 출력이 표시됩니다.

```
connecting to device: nvme0
```

a. list-subsys를 확인하세요:

```
nvme list-subsys
```

예제 출력을 표시합니다

```
nvme-subsys0 - NQN=nqn.1992-
08.com.netapp:sn.a2d41235b78211efb57dd039eab67a95:subsystem.nvme1
               hostnqn=nqn.2014-08.org.nvmexpress: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. 대상을 추가하고 지정된 ONTAP 하위 시스템에 대한 TLS 연결을 확인합니다.

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

예제 출력을 표시합니다

```
(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
```

11단계: 알려진 문제를 검토합니다

알려진 문제가 없습니다.