



RHEL을 참조하십시오 ONTAP SAN Host Utilities

NetApp
January 29, 2026

목차

RHEL을 참조하십시오	1
RHEL 호스트에 대한 ONTAP 지원 및 기능에 대해 알아보세요.	1
다음은 무엇입니까?	1
ONTAP 스토리지를 사용하여 NVMe-oF에 대한 RHEL 10.x 구성	2
1단계: 필요에 따라 SAN 부팅을 활성화합니다.	2
2단계: RHEL 및 NVMe 소프트웨어 설치 및 구성 확인.	2
3단계: NVMe/FC 및 NVMe/TCP 구성	4
4단계: 선택적으로 udev 규칙에서 iopolicy를 수정합니다.	12
5단계: 선택적으로 NVMe/FC에 대해 1MB I/O를 활성화합니다.	13
6단계: NVMe 부팅 서비스 확인	14
7단계: 다중 경로 구성 확인	15
8단계: 안전한 인밴드 인증 설정	20
9단계: 알려진 문제를 검토합니다	26
ONTAP 스토리지를 사용하여 NVMe-oF에 대한 RHEL 9.x 구성	26
1단계: 필요에 따라 SAN 부팅을 활성화합니다.	26
2단계: RHEL 및 NVMe 소프트웨어 설치 및 구성 확인.	26
3단계: NVMe/FC 및 NVMe/TCP 구성	28
4단계: 선택적으로 udev 규칙에서 iopolicy를 수정합니다.	37
5단계: 선택적으로 NVMe/FC에 대해 1MB I/O를 활성화합니다.	38
6단계: NVMe 부팅 서비스 확인	39
7단계: 다중 경로 구성 확인	40
8단계: 안전한 인밴드 인증 설정	45
9단계: 알려진 문제를 검토합니다	51
ONTAP 스토리지를 사용하여 NVMe-oF에 대한 RHEL 8.x 구성	52
1단계: 필요에 따라 SAN 부팅을 활성화합니다.	52
2단계: RHEL 및 NVMe 소프트웨어 설치 및 구성 확인.	52
3단계: NVMe/FC 및 NVMe/TCP 구성	54
4단계: 선택적으로 NVMe/FC에 대해 1MB I/O를 활성화합니다.	60
5단계: 다중 경로 구성 확인	60
6단계: 알려진 문제를 검토합니다	65

RHEL을 참조하십시오

RHEL 호스트에 대한 ONTAP 지원 및 기능에 대해 알아보세요.

NVMe over Fabrics(NVMe-oF)를 사용한 호스트 구성에 지원되는 기능은 ONTAP 및 RHEL 버전에 따라 다릅니다.

특징	RHEL 호스트 버전	ONTAP 버전
RHEL 호스트와 ONTAP 컨트롤러 간의 NVMe/TCP를 통해 안전한 인밴드 인증이 지원됩니다.	9.3 이상	9.12.1 이상
NVMe/TCP는 네이티브를 사용하여 네임스페이스를 제공합니다. <code>nvme-cli</code> 패키지	8.2 이상	9.10.1 이상
NVMe/TCP는 완벽하게 지원되는 엔터프라이즈 기능입니다.	9.0 이상	9.10.1 이상
NVMe 및 SCSI 트래픽은 NVMe-oF 네임스페이스의 경우 NVMe 멀티패스를 사용하고 SCSI LUN의 경우 <code>dm-multipath</code> 를 사용하여 동일한 호스트에서 지원됩니다.	8.2 이상	9.4 이상

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

특징	RHEL 호스트 버전
기본적으로 네이티브 NVMe 멀티패스가 활성화되어 있습니다.	10.0 이상
<code>nvme-cli</code> 패키지의 네이티브 <code>udev</code> 규칙은 NVMe 멀티패싱을 위한 큐 깊이 기반 로드 밸런싱을 제공합니다	9.6 이상
NVMe/FC 프로토콜을 사용하여 SAN 부팅이 활성화됩니다.	9.4 이상
그만큼 <code>nvme-cli</code> 패키지에는 타사 스크립트가 필요 없는 자동 연결 스크립트가 포함되어 있습니다.	8.2 이상
<code>nvme-cli</code> 패키지의 네이티브 <code>udev</code> 규칙은 NVMe 멀티패싱을 위한 라운드 로빈 로드 밸런싱을 제공합니다	8.2 이상



지원되는 구성에 대한 자세한 내용은 다음을 참조하세요. ["상호 운용성 매트릭스 툴"](#).

다음은 무엇입니까?

사용 중인 RHEL 버전이 ...인 경우	다음 내용을 알아보세요...
10 시리즈	"RHEL 10.x에 대한 NVMe 구성"
9 시리즈	"RHEL 9.x에 대한 NVMe 구성"
8시리즈	"RHEL 8.x에 대한 NVMe 구성"

관련 정보

ONTAP 스토리지를 사용하여 NVMe-oF에 대한 RHEL 10.x 구성

Red Hat Enterprise Linux(RHEL) 호스트는 비대칭 네임스페이스 액세스(ANA)를 통해 NVMe over Fibre Channel(NVMe/FC) 및 NVMe over TCP(NVMe/TCP) 프로토콜을 지원합니다. ANA는 iSCSI 및 FCP 환경에서 ALUA(비대칭 논리 장치 액세스)와 동일한 다중 경로 기능을 제공합니다.

RHEL 10.x에 대해 NVMe over Fabrics(NVMe-oF) 호스트를 구성하는 방법을 알아보세요. 추가 지원 및 기능 정보는 다음을 참조하세요. ["RHEL ONTAP 지원 및 기능"](#).

RHEL 10.x를 사용하는 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.

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: RHEL 및 NVMe 소프트웨어 설치 및 구성 확인

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

단계

1. 서버에 RHEL 10.x를 설치합니다. 설치가 완료되면 필요한 RHEL 10.x 커널을 실행하고 있는지 확인하세요.

```
uname -r
```

RHEL 커널 버전 예:

```
6.12.0-124.8.1.el10_1.x86_64
```

2. "NVMe-CLI" 패키지를 설치합니다.

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

다음 예에서는 다음을 보여줍니다. nvme-cli 패키지 버전:

```
nvme-cli-2.13-2.el10.x86_64
```

3. 를 설치합니다 libnvme 패키지:

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

다음 예에서는 다음을 보여줍니다. libnvme 패키지 버전:

```
libnvme-1.13-1.el10.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:4c4c4544-0056-5410-8048-c7c04f425633
```

5. ONTAP 시스템에서 다음을 확인하십시오. hostnqn 문자열이 일치합니다 hostnqn ONTAP 스토리지 시스템의 해당 하위 시스템에 대한 문자열:

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

```

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

```



만약 hostnqn 문자열이 일치하지 않으면 다음을 사용하세요. vserver modify 업데이트 명령 hostnqn 해당 ONTAP 스토리지 시스템 하위 시스템의 문자열을 일치시키세요. hostnqn 문자열에서 /etc/nvme/hostnqn 호스트에서.

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 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.

```
SN1700E2P  
SN1700E2P
```

b. 모델 설명을 표시합니다.

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

다음 예와 비슷한 출력이 표시되어야 합니다.

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

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

a. 펌웨어 버전을 표시합니다.

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

이 명령은 펌웨어 버전을 반환합니다.

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

b. 받은 편지함 드라이버 버전을 표시합니다.

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

다음 예에서는 드라이버 버전을 보여줍니다.

```
0:14.4.0.9
```

지원되는 어댑터 드라이버 및 펌웨어 버전의 현재 목록은 를 참조하십시오 ["상호 운용성 매트릭스 툴"](#).

3. 확인합니다 lpfc_enable_fc4_type 가 로 설정되어 있습니다 3:

```
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.

```
0x10005cba2cfca7de  
0x10005cba2cfca7df
```

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
```



```

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

```

```

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

```

```

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

```

```

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

```

NVMe/FC - Marvell/QLogic

Marvell/QLogic 어댑터용 NVMe/FC를 구성합니다.

단계

1. 지원되는 어댑터 드라이버와 펌웨어 버전을 사용하고 있는지 확인하세요.

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

다음 예에서는 드라이버와 펌웨어 버전을 보여줍니다.

```
QLE2872 FW:v9.15.06 DVR:v10.02.09.400-k  
QLE2872 FW:v9.15.06 DVR:v10.02.09.400-k
```

2. 확인합니다 `ql2xnvmeeenable` 가 설정됩니다. 그러면 Marvell 어댑터가 NVMe/FC Initiator로 작동할 수 있습니다.

```
cat /sys/module/qla2xxx/parameters/ql2xnvmeeenable
```

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. 지원되는 NVMe/TCP LIF를 통해 개시자 포트가 검색 로그 페이지 데이터를 가져올 수 있는지 확인하세요.

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

```

nvme discover -t tcp -w 192.168.20.21 -a 192.168.20.28
Discovery Log Number of Records 8, Generation counter 10
=====Discovery Log Entry 0=====
trtype: tcp
adrfam: ipv4
subtype: current discovery subsystem
treq: not specified
portid: 8
trsvcid: 8009
subnqn: nqn.1992-
08.com.netapp:sn.17e32b6e8c7f11f09545d039eac03c33:discovery
traddr: 192.168.21.29
eflags: explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 1=====
trtype: tcp
adrfam: ipv4
subtype: current discovery subsystem
treq: not specified
portid: 6
trsvcid: 8009
subnqn: nqn.1992-
08.com.netapp:sn.17e32b6e8c7f11f09545d039eac03c33:discovery
traddr: 192.168.20.29
eflags: explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 2=====
trtype: tcp
adrfam: ipv4
subtype: current discovery subsystem
treq: not specified
portid: 7
trsvcid: 8009
subnqn: nqn.1992-
08.com.netapp:sn.17e32b6e8c7f11f09545d039eac03c33:discovery
traddr: 192.168.21.28
eflags: explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 3=====
trtype: tcp

```

```

adrfam:  ipv4
subtype: current discovery subsystem
treq:    not specified
portid:  5
trsvcid: 8009
subnqn:  nqn.1992-
08.com.netapp:sn.17e32b6e8c7f11f09545d039eac03c33:discovery
traddr:  192.168.20.28
eflags:  explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 4=====
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  8
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.17e32b6e8c7f11f09545d039eac03c33:subsystem.Bidi
rectional_DHCP_1_0
traddr:  192.168.21.29
eflags:  none
sectype: none
=====Discovery Log Entry 5=====
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  6
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.17e32b6e8c7f11f09545d039eac03c33:subsystem.Bidi
rectional_DHCP_1_0
traddr:  192.168.20.29
eflags:  none
sectype: none
=====Discovery Log Entry 6=====
trtype:  tcp
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  7
trsvcid: 4420
subnqn:  nqn.1992-
08.com.netapp:sn.17e32b6e8c7f11f09545d039eac03c33:subsystem.Bidi

```

```

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

```

2. 다른 NVMe/TCP 개시자-대상 LIF 조합이 검색 로그 페이지 데이터를 성공적으로 검색할 수 있는지 확인합니다.

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

예제 보기

```

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

```

3. 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 192.168.20.21 -a 192.168.20.28
nvme connect-all -t tcp -w 192.168.21.21 -a 192.168.21.28
nvme connect-all -t tcp -w 192.168.20.21 -a 192.168.20.29
nvme connect-all -t tcp -w 192.168.21.21 -a 192.168.21.29
```

RHEL 9.4부터 NVMe/TCP 설정 `ctrl_loss_tmo` timeout 자동으로 "꺼짐"으로 설정됩니다. 그 결과,

- 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 -).
- NVMe/TCP 컨트롤러는 경로 장애가 발생해도 시간 초과가 발생하지 않으며 무기한 연결 상태를 유지합니다.

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

RHEL 10.0은 NVMe-oF에 대한 기본 `iopolicy`를 다음과 같이 설정합니다. `round-robin`. RHEL 10.0을 사용하고 `iopolicy`를 변경하려는 경우 `queue-depth`, `udev` 규칙 파일을 다음과 같이 수정합니다.

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 `iopolicy`를 설정하는 줄을 찾으십시오.

```
ACTION=="add", SUBSYSTEM=="nvme-subsystem", ATTR{subsysstype}=="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{subsysstype}=="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 값 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 `nvme-fc-boot-connections.service` 그리고 `nvmmf-autoconnect.service` NVMe/FC에 포함된 부팅 서비스 `nvme-cli` 패키지는 시스템이 부팅될 때 자동으로 활성화됩니다.

부팅이 완료된 후 다음을 확인하세요. `nvme-fc-boot-connections.service` 그리고 `nvmmf-autoconnect.service` 부팅 서비스가 활성화되었습니다.

단계

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

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

예제 출력을 표시합니다

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

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

2. 가 활성화되어 있는지 `nvme-fc-boot-connections.service` 확인합니다.

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


예제 출력을 표시합니다

```
nvmeofc-boot-connections.service - Auto-connect to subsystems on FC-
NVME devices found during boot
   Loaded: loaded (/usr/lib/systemd/system/nvmeofc-boot-
connections.service; enabled; preset: enabled)
   Active: inactive (dead) since Sun 2025-10-12 19:40:33 IST; 1
day 1h ago
  Invocation: 0ec258a9f8c342ffb82408086d409bc6
    Main PID: 4151 (code=exited, status=0/SUCCESS)
   Mem peak: 2.9M
      CPU: 17ms

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

7단계: 다중 경로 구성 확인

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

단계

1. 호스트에서 각 ONTAP 네임스페이스에 대해 적절한 NVMe-oF 설정(예: 모델이 NetApp ONTAP 컨트롤러로 설정되고 로드 밸런싱 iopolicy가 queue-depth로 설정됨)이 올바르게 표시되는지 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
```

2. 호스트에서 네임스페이스가 생성되고 올바르게 검색되는지 확인합니다.

```
nvme list
```

예제 보기

```
Node          Generic          SN
Model
-----
/dev/nvme11n1 /dev/ng11n1      810cqJXhgWtsAAAAAAAI
NetApp ONTAP Controller

Namespace  Usage          Format          FW Rev
-----
0x1        951.90 MB / 21.47 GB  4 KiB + 0 B    9.18.1
```

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

NVMe/FC

```
nvme list-subsys /dev/nvme9n2
```

예제 보기

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

NVMe/TCP

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

예제 보기

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

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

명

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

예제 보기

Device	Vserver	Subsystem
Namespace Path		

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

JSON을 참조하십시오

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

예제 보기

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

8단계: 안전한 인밴드 인증 설정

RHEL 10.x 호스트와 ONTAP 컨트롤러 간의 NVMe/TCP를 통해 안전한 인밴드 인증이 지원됩니다.

각 호스트 또는 컨트롤러는 다음과 연결되어야 합니다. DH-HMAC-CHAP 보안 인증을 설정하는 키입니다. DH-HMAC-CHAP 키는 NVMe 호스트 또는 컨트롤러의 NQN과 관리자가 구성한 인증 비밀번호의 조합입니다. 피어를 인증하려면 NVMe 호스트 또는 컨트롤러가 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. RHEL 10.x 호스트에 대한 dhchap 키를 생성합니다.

다음 출력은 다음을 설명합니다. `gen-dhchap-key` 명령 매개변수:

```
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:4c4c4544-0054-5110-8039-c3c04f523034
DHHC-
1:03:AppJHkJygA6ZC4BxyQNTJST+4k4IOv47MAJk0xBITwFOHIC2nV/uE04RoSpy1z2
SXYqNW1bhLe9hJ+MDHigGexaG2Ig=:
```

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-subsys4/nvme*/dhchap_secret
DHHC-1:01:2G71sg9PMO00h1Wf1g4QtP0XT11kREz0qVuLm2xvZdbaWR/g:
DHHC-1:01:2G71sg9PMO00h1Wf1g4QtP0XT11kREz0qVuLm2xvZdbaWR/g:
DHHC-1:01:2G71sg9PMO00h1Wf1g4QtP0XT11kREz0qVuLm2xvZdbaWR/g:
DHHC-1:01:2G71sg9PMO00h1Wf1g4QtP0XT11kREz0qVuLm2xvZdbaWR/g:
```

b. 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-
subsys4/nvme*/dhchap_ctrl_secret
DHHC-
1:03:5CgWULVnU5HUOWP1MNg95pkiUAwayiO+IvrALZR8HpeJIHw3xyHdGlTnv
EJ81HDjBb+fGteUgIn0fj8ASHZigkuFIx8=:
DHHC-
1:03:5CgWULVnU5HUOWP1MNg95pkiUAwayiO+IvrALZR8HpeJIHw3xyHdGlTnv
EJ81HDjBb+fGteUgIn0fj8ASHZigkuFIx8=:
DHHC-
1:03:5CgWULVnU5HUOWP1MNg95pkiUAwayiO+IvrALZR8HpeJIHw3xyHdGlTnv
EJ81HDjBb+fGteUgIn0fj8ASHZigkuFIx8=:
DHHC-
1:03:5CgWULVnU5HUOWP1MNg95pkiUAwayiO+IvrALZR8HpeJIHw3xyHdGlTnv
EJ81HDjBb+fGteUgIn0fj8ASHZigkuFIx8=:
```


JSON을 참조하십시오

ONTAP 컨트롤러에서 여러 NVMe 하위 시스템을 사용할 수 있는 경우 다음을 사용할 수 있습니다.
`/etc/nvme/config.json` 파일로 저장 `nvme connect-all` 명령.

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

1. JSON 파일을 구성합니다.



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

```
[
  {
    "hostnqn": "nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0054-5110-8039-c3c04f523034",
    "hostid": "44454c4c-5400-1051-8039-c3c04f523034",
    "dhchap_key": "DHHC-1:01:2G7lsg9PM000h1Wf1g4QtP0XT11kREz0qVuLm2xvZdbaWR/g:",
    "subsystems": [
      {
        "nqn": "nqn.1992-08.com.netapp:sn.5857c8c9b22411f08d0ed039eac03c33:subsystem.Bidirectional_DHCP_1_0",
        "ports": [
          {
            "transport": "tcp",
            "traddr": "192.168.20.28",
            "host_traddr": "192.168.20.21",
            "trsvcid": "4420",
            "dhchap_ctrl_key": "DHHC-1:03:5CgWULVnU5HUOwPlMN95pkiUAwayiO+IvrALZR8HpeJIHw3xyHdGlTnvEJ81HDjBb+fGteUgIn0fj8ASHZigkuFIx8="
          },
          {
            "transport": "tcp",
            "traddr": "192.168.20.29",
            "host_traddr": "192.168.20.21",
            "trsvcid": "4420",
            "dhchap_ctrl_key": "DHHC-1:03:5CgWULVnU5HUOwPlMN95pkiUAwayiO+IvrALZR8HpeJIHw3xyHdGlTnvEJ81HDjBb+fGteUgIn0fj8ASHZigkuFIx8="
          },
          {
            "transport": "tcp",
            "traddr": "192.168.21.28",
            "host_traddr": "192.168.21.21",
            "trsvcid": "4420",
            "dhchap_ctrl_key": "DHHC-1:03:5CgWULVnU5HUOwPlMN95pkiUAwayiO+IvrALZR8HpeJIHw3xyHdGlTnvEJ81HDjBb+fGteUgIn0fj8ASHZigkuFIx8="
          },
          {
            "transport": "tcp",
            "traddr": "192.168.21.29",
```

```

        "host_traddr": "192.168.21.21",
        "trsvcid": "4420",
        "dhchap_ctrl_key": "DHHC-
1:03:5CgWULVnU5HUOWPlMN95pkiUAWayIO+IvrALZR8HpeJIHw3xyHdGlTnvEJ
81HDjBb+fGteUgIn0fj8ASHZigkuFIx8="
    }
}
}
]
}
]

```

2. config JSON 파일을 사용하여 ONTAP 컨트롤러에 연결합니다.

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

예제 보기

```

traddr=192.168.20.28 is already connected
traddr=192.168.20.28 is already connected
traddr=192.168.20.29 is already connected
traddr=192.168.20.29 is already connected

```

3. 각 하위 시스템의 해당 컨트롤러에 대해 dhchap 비밀번호가 활성화되었는지 확인하세요.

a. 호스트 dhchap 키를 확인합니다.

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

다음 예에서는 dhchap 키를 보여줍니다.

```
DHHC-1:01:2G7lsg9PMO00h1Wf1g4QtP0XT11kREz0qVuLm2xvZdbaWR/g:
```

b. 컨트롤러 dhchap 키를 확인합니다.

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

다음 예와 비슷한 출력이 표시되어야 합니다.

DHHC-

1:03:5CgWULVnU5HUOWPlMN95pkiUAWayio+IvrALZR8HpeJIHw3xyHdGlTnvEJ81HD
jBb+fGteUgIn0fj8ASHZigkuFIx8=:

9단계: 알려진 문제를 검토합니다

알려진 문제가 없습니다.

ONTAP 스토리지를 사용하여 NVMe-oF에 대한 RHEL 9.x 구성

Red Hat Enterprise Linux(RHEL) 호스트는 비대칭 네임스페이스 액세스(ANA)를 통해 NVMe over Fibre Channel(NVMe/FC) 및 NVMe over TCP(NVMe/TCP) 프로토콜을 지원합니다. ANA는 iSCSI 및 FCP 환경에서 ALUA(비대칭 논리 장치 액세스)와 동일한 다중 경로 기능을 제공합니다.

RHEL 9.x에 대해 NVMe over Fabrics(NVMe-oF) 호스트를 구성하는 방법을 알아보세요. 추가 지원 및 기능 정보는 다음을 참조하세요. ["RHEL ONTAP 지원 및 기능"](#).

RHEL 9.x를 사용하는 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.

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: RHEL 및 NVMe 소프트웨어 설치 및 구성 확인

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

단계

1. 서버에 RHEL 9.x를 설치합니다. 설치가 완료되면 필요한 RHEL 9.x 커널을 실행하고 있는지 확인하세요.

```
uname -r
```

RHEL 커널 버전 예:

```
5.14.0-611.5.1.el9_7.x86_64
```

2. "NVMe-CLI" 패키지를 설치합니다.

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

다음 예에서는 다음을 보여줍니다. nvme-cli 패키지 버전:

```
nvme-cli-2.13-1.el9.x86_64
```

3. 를 설치합니다 libnvme 패키지:

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

다음 예에서는 다음을 보여줍니다. libnvme 패키지 버전:

```
libnvme-1.13-1.el9.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:4c4c4544-0056-5410-8048-b9c04f425633
```

5. ONTAP 시스템에서 다음을 확인하십시오. hostnqn 문자열이 일치합니다 hostnqn ONTAP 스토리지 시스템의 해당 하위 시스템에 대한 문자열:

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

```

Vserver Subsystem Priority Host NQN
-----
vs_188 Nvme1
        regular nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
        Nvme10
        regular nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
        Nvme11
        regular nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
        Nvme12
        regular nqn.2014-
08.org.nvmexpress:uuid:4c4c4544-0056-5410-8048-b9c04f425633
48 entries were displayed.

```



만약 hostnqn 문자열이 일치하지 않으면 다음을 사용하세요. vserver modify 업데이트 명령 hostnqn 해당 ONTAP 스토리지 시스템 하위 시스템의 문자열을 일치시키세요. hostnqn 문자열에서 /etc/nvme/hostnqn 호스트에서.

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 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
```

다음 예와 비슷한 출력이 표시되어야 합니다.

```
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
```

이 명령은 펌웨어 버전을 반환합니다.

```
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.

```
0:14.4.0.9
```

지원되는 어댑터 드라이버 및 펌웨어 버전의 현재 목록은 를 참조하십시오 ["상호 운용성 매트릭스 툴"](#).

3. 확인합니다 lpfc_enable_fc4_type 가 로 설정되어 있습니다 3:

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

4. 이니시에이터 포트를 볼 수 있는지 확인합니다.

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

다음 예에서는 포트 ID를 보여줍니다.

```
0x100000109bf044b1  
0x100000109bf044b2
```

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
```



```

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

```

```

NVME Statistics
LS: Xmt 0000000548 Cmpl 0000000548 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 00000000000001a68 Issue 00000000000001a68 OutIO
0000000000000000
          abort 00000000 noxri 00000000 nondlp 00000000 qdepth
00000000 wqerr 00000000 err 00000000
FCP CMPL: xb 00000000 Err 00000000

```

```

NVME Initiator Enabled
XRI Dist lpfc1 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc1 WWPN x100000109b954519 WWNN x200000109b954519
DID x020500 ONLINE
NVME RPORT          WWPN x2027d039eaa7dfc8 WWNN x2025d039eaa7dfc8
DID x020b01 TARGET DISCSRVC ONLINE

```

```

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

```

```

NVME Initiator Enabled
XRI Dist lpfc2 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc2 WWPN x100000109bf044b1 WWNN x200000109bf044b1
DID x022a00 ONLINE
NVME RPORT          WWPN x2027d039eaa7dfc8 WWNN x2025d039eaa7dfc8
DID x020b01 TARGET DISCSRVC ONLINE
NVME RPORT          WWPN x2011d039eaa7dfc8 WWNN x200fd039eaa7dfc8
DID x020b02 TARGET DISCSRVC ONLINE
NVME RPORT          WWPN x2002d039eaa7dfc8 WWNN x2000d039eaa7dfc8
DID x020b05 TARGET DISCSRVC ONLINE

```

```

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

```

NVME Statistics

```

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

```

NVME Initiator Enabled

```

XRI Dist lpfc3 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc3 WWPN x100000109bf044b2 WWNN x200000109bf044b2
DID x021b00 ONLINE
NVME RPORT          WWPN x2028d039eaa7dfc8 WWNN x2025d039eaa7dfc8
DID x020101 TARGET DISCSRV ONLINE
NVME RPORT          WWPN x2012d039eaa7dfc8 WWNN x200fd039eaa7dfc8
DID x020102 TARGET DISCSRV ONLINE
NVME RPORT          WWPN x2003d039eaa7dfc8 WWNN x2000d039eaa7dfc8
DID x020105 TARGET DISCSRV ONLINE
NVME RPORT          WWPN x2029d039eaa7dfc8 WWNN x2025d039eaa7dfc8
DID x022901 TARGET DISCSRV ONLINE
NVME RPORT          WWPN x2013d039eaa7dfc8 WWNN x200fd039eaa7dfc8
DID x022902 TARGET DISCSRV ONLINE
NVME RPORT          WWPN x2004d039eaa7dfc8 WWNN x2000d039eaa7dfc8
DID x022905 TARGET DISCSRV ONLINE

```

NVME Statistics

```

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

```

NVMe/FC - Marvell/QLogic

Marvell/QLogic 어댑터용 NVMe/FC를 구성합니다.

단계

1. 지원되는 어댑터 드라이버와 펌웨어 버전을 사용하고 있는지 확인하세요.

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

다음 예에서는 드라이버와 펌웨어 버전을 보여줍니다.

```
QLE2872 FW:v9.15.06 DVR:v10.02.09.400-k  
QLE2872 FW:v9.15.06 DVR:v10.02.09.400-k
```

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. 지원되는 NVMe/TCP LIF를 통해 개시자 포트가 검색 로그 페이지 데이터를 가져올 수 있는지 확인하세요.

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

```

nvme discover -t tcp -w 192.168.30.15 -a 192.168.30.48

Discovery Log Number of Records 8, Generation counter 18
=====Discovery Log Entry 0=====
trtype: tcp
adrfam: ipv4
subtype: current discovery subsystem
treq: not specified
portid: 8
trsvcid: 8009
subnqn: nqn.1992-
08.com.netapp:sn.51a3c9846e0c11f08f5dd039eaa7dfc9:discovery
traddr: 192.168.31.49
eflags: explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 1=====
trtype: tcp
adrfam: ipv4
subtype: current discovery subsystem
treq: not specified
portid: 7
trsvcid: 8009
subnqn: nqn.1992-
08.com.netapp:sn.51a3c9846e0c11f08f5dd039eaa7dfc9:discovery
traddr: 192.168.31.48
eflags: explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 2=====
trtype: tcp
adrfam: ipv4
subtype: current discovery subsystem
treq: not specified
portid: 6
trsvcid: 8009
subnqn: nqn.1992-
08.com.netapp:sn.51a3c9846e0c11f08f5dd039eaa7dfc9:discovery
traddr: 192.168.30.49
eflags: explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 3=====

```

```

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

```

```

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

```

2. 다른 NVMe/TCP 개시자-대상 LIF 조합이 검색 로그 페이지 데이터를 성공적으로 검색할 수 있는지 확인합니다.

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

예제 보기

```

nvme discover -t tcp -w 192.168.30.15 -a 192.168.30.48
nvme discover -t tcp -w 192.168.30.15 -a 192.168.30.49
nvme discover -t tcp -w 192.168.31.15 -a 192.168.31.48
nvme discover -t tcp -w 192.168.31.15 -a 192.168.31.49

```

3. 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 192.168.30.15 -a 192.168.30.48
nvme connect-all -t tcp -w 192.168.30.15 -a
192.168.30.49
nvme connect-all -t tcp -w 192.168.31.15 -a
192.168.31.48
nvme connect-all -t tcp -w 192.168.31.15 -a
192.168.31.49
```

RHEL 9.4부터 NVMe/TCP 설정 `ctrl_loss_tmo` timeout 자동으로 "꺼짐"으로 설정됩니다. 그 결과,

- 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 -).
- NVMe/TCP 컨트롤러는 경로 장애가 발생해도 시간 초과가 발생하지 않으며 무기한 연결 상태를 유지합니다.

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

RHEL 9.6은 NVMe-oF에 대한 기본 `iopolicy`를 다음과 같이 설정합니다. `round-robin`. RHEL 9.6을 사용하고 `iopolicy`를 변경하려는 경우 `queue-depth`, `udev` 규칙 파일을 다음과 같이 수정합니다.

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 `iopolicy`를 설정하는 줄을 찾으십시오.

```
ACTION=="add", SUBSYSTEM=="nvme-subsystem", ATTR{subsysstype}=="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{subsysname}=="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. lpfc의 값 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 `nvme-fc-boot-connections.service` 그리고 `nvmmf-autoconnect.service` NVMe/FC에 포함된 부팅 서비스 `nvme-cli` 패키지는 시스템이 부팅될 때 자동으로 활성화됩니다.

부팅이 완료된 후 다음을 확인하세요. `nvme-fc-boot-connections.service` 그리고 `nvmmf-autoconnect.service` 부팅 서비스가 활성화되었습니다.

단계

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

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

예제 출력을 표시합니다

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

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

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

2. 가 활성화되어 있는지 `nvme-fc-boot-connections.service` 확인합니다.

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

예제 출력을 표시합니다

```
nvmeofc-boot-connections.service - Auto-connect to subsystems on FC-
NVME devices found during boot
    Loaded: loaded (/usr/lib/systemd/system/nvmeofc-boot-
connections.service; enabled; preset:enabled)
    Active: inactive (dead) since Wed 2025-10-29 00:41:51 EDT; 6h
ago
    Main PID: 4652 (code=exited, status=0/SUCCESS)
    CPU: 13ms

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

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. 호스트에서 각 ONTAP 네임스페이스에 대해 적절한 NVMe-oF 설정(예: 모델이 NetApp ONTAP 컨트롤러로 설정되고 로드 밸런싱 iopolicy가 라운드 로빈으로 설정됨)이 올바르게 표시되는지 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		Generic	SN
Model			

/dev/nvme100n1	/dev/ng100n1	81LJCJYaKOHhAAAAAAaf	NetApp ONTAP
Controller			
Namespace Usage	Format	FW	Rev

0x1	1.19 GB /	5.37 GB 4 KiB + 0 B	9.18.1

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

NVMe/FC

```
nvme list-subsys /dev/nvme100n1
```

예제 보기

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

NVMe/TCP

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

예제 보기

```
nvme-subsys0 - NQN=nqn.1992-08.com.netapp:sn.51a3c9846e0c11f08f5dd039eaa7dfc9:subsystem.Nvme
1
hostnqn=nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0035-5910-804b-
b5c04f444d33
\
+- nvme0 tcp
traddr=192.168.30.48,trsvcid=4420,host_traddr=192.168.30.15,
src_addr=192.168.30.15 live optimized
+- nvme1 tcp
traddr=192.168.30.49,trsvcid=4420,host_traddr=192.168.30.15,
src_addr=192.168.30.15 live non-optimized
+- nvme2 tcp
traddr=192.168.31.48,trsvcid=4420,host_traddr=192.168.31.15,
src_addr=192.168.31.15 live optimized
+- nvme3 tcp
traddr=192.168.31.49,trsvcid=4420,host_traddr=192.168.31.15,
src_addr=192.168.31.15 live non-optimized
```

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

명

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

예제 보기

Device NSID	Vserver	Subsystem	Namespace Path
----- ----	-----	-----	-----
/dev/nvme0n1 1	vs_iscsi_tcp	Nvme1	/vol/Nvmevol1/ns1
UUID		Size	
-----		-----	
d8efef7d-4dde-447f-b50e-b2c009298c66		26.84GB	

JSON을 참조하십시오

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

예제 보기

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

8단계: 안전한 인밴드 인증 설정

RHEL 9.x 호스트와 ONTAP 컨트롤러 간의 NVMe/TCP를 통해 안전한 인밴드 인증이 지원됩니다.

각 호스트 또는 컨트롤러는 다음과 연결되어야 합니다. DH-HMAC-CHAP 보안 인증을 설정하는 키입니다. DH-HMAC-CHAP 키는 NVMe 호스트 또는 컨트롤러의 NQN과 관리자가 구성한 인증 비밀번호의 조합입니다. 피어를 인증하려면 NVMe 호스트 또는 컨트롤러가 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. RHEL 9.x 호스트에 대한 dhchap 키를 생성합니다.

다음 출력은 다음을 설명합니다. `gen-dhchap-key` 명령 매개변수:

```
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:wSpuuKbBHTzC0W9JZxMBSYd9JFV8Si9aDh22k2BR/4m852vH7KGlrJeMpzhmyjD
W0o0PJJM6yZsTeEpGkDHMHQ255+g=:
```

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

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-  
subsys*/nvme*/dhchap_ctrl_secret  
  
DHHC-  
1:03:ZCRrP9MQOeXhFitT7Fvvf/3P6K/qY1HfSmSfM8nLjESJdOjbjK/J6m00y  
gJgjm0VrRlgrnHzjtWJmsnoVBO3rPDGEk=:  
DHHC-  
1:03:ZCRrP9MQOeXhFitT7Fvvf/3P6K/qY1HfSmSfM8nLjESJdOjbjK/J6m00y  
gJgjm0VrRlgrnHzjtWJmsnoVBO3rPDGEk=:  
DHHC-  
1:03:ZCRrP9MQOeXhFitT7Fvvf/3P6K/qY1HfSmSfM8nLjESJdOjbjK/J6m00y  
gJgjm0VrRlgrnHzjtWJmsnoVBO3rPDGEk=:  
DHHC-  
1:03:ZCRrP9MQOeXhFitT7Fvvf/3P6K/qY1HfSmSfM8nLjESJdOjbjK/J6m00y  
gJgjm0VrRlgrnHzjtWJmsnoVBO3rPDGEk=:
```

JSON을 참조하십시오

ONTAP 컨트롤러에서 여러 NVMe 하위 시스템을 사용할 수 있는 경우 다음을 사용할 수 있습니다.
/etc/nvme/config.json 파일로 저장 nvme connect-all 명령.

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

1. JSON 파일을 구성합니다.



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

```

cat /etc/nvme/config.json
[
{
  "hostnqn":"nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0035-5910-
804b-b5c04f444d33",
  "hostid":"4c4c4544-0035-5910-804b-b5c04f444d33",
  "dhchap_key":"DHHC-
1:01:GhgaLS+0h0W/IxKhSa0iaMHg17SOHRTzBduPzoJ6LKEJs3/f:",
  "subsystems":[
    {
      "nqn":"nqn.1992-
08.com.netapp:sn.2c0c80d9873a11f0bc60d039eab6cb6d:subsystem.istp
MNTC_subsys",
      "ports":[
        {
          "transport":"tcp",
          "traddr":"192.168.30.44",
          "host_traddr":"192.168.30.15",
          "trsvcid":"4420",
          "dhchap_ctrl_key":"DHHC-
1:03:GaraCO84o/uM0jF4rKJlgTy22bVoV0dRn1M+9QDfQRNVwJDHfPu2LrK5Y+/
XG8iGcRtBCdm3
fYm3ZmO6NiepCORoY5Q=:"
        },
        {
          "transport":"tcp",
          "traddr":"192.168.30.45"
          "host_traddr":"192.168.30.15",
          "trsvcid":"4420",
          "dhchap_ctrl_key":"DHHC-
1:03:GaraCO84o/uM0jF4rKJlgTy22bVoV0dRn1M+9QDfQRNVwJDHfPu2LrK5Y+/
XG8iGcRtBCdm3
fYm3ZmO6NiepCORoY5Q=:"
        },
        {
          "transport":"tcp",
          "traddr":"192.168.31.44",
          "host_traddr":"192.168.31.15",
          "trsvcid":"4420",
          "dhchap_ctrl_key":"DHHC-
1:03:
GaraCO84o/uM0jF4rKJlgTy22bVoV0dRn1M+9QDfQRNVwJDHfPu2LrK5Y+/XG8iG
c

```

```

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

```

2. config JSON 파일을 사용하여 ONTAP 컨트롤러에 연결합니다.

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

예제 보기

```

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

```

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

a. 호스트 dhchap 키를 확인합니다.

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

다음 예에서는 dhchap 키를 보여줍니다.

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

b. 컨트롤러 dhchap 키를 확인합니다.

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

다음 예와 비슷한 출력이 표시되어야 합니다.

```
DHHC-  
1:03:ZCRrP9MQOeXhFitT7Fvvf/3P6K/qY1HfSmSfM8nLjESJdOjbjK/J6m00ygJgjm0  
VrRlgrnHzjtWJmsnoVBO3rPDGEk=:
```

9단계: 알려진 문제를 검토합니다

알려진 문제는 다음과 같습니다.

NetApp 버그 ID	제목	설명
1503468	RHEL 9.1에서는 <code>nvme list-subsys</code> 명령은 주어진 하위 시스템에 대한 반복된 NVME 컨트롤러 목록을 반환합니다.	그만큼 <code>nvme list-subsys</code> 명령은 주어진 하위 시스템에 대한 NVMe 컨트롤러 목록을 반환합니다. RHEL 9.1에서 이 명령은 하위 시스템의 모든 네임스페이스에 대한 ANA 상태와 함께 컨트롤러를 표시합니다. ANA 상태는 네임스페이스별 속성이므로 명령은 각 네임스페이스의 경로 상태를 포함하는 고유한 컨트롤러 항목을 표시해야 합니다.
"1479047"	RHEL 9.0 NVMe-oF 호스트는 중복된 PDC(Persistent Discovery Controller)를 생성합니다.	NVMe-oF 호스트에서 <code>nvme discover -p</code> 명령을 사용하여 PDC를 생성할 수 있습니다. 이 명령을 사용하면 개시자-대상 조합당 하나의 PDC만 생성되어야 합니다. 그러나 NVMe-oF 호스트를 사용하여 ONTAP 9.10.1 및 RHEL 9.0을 실행하는 경우 <code>nvme discover -p</code> 가 실행될 때마다 중복 PDC가 생성됩니다. 이로 인해 호스트와 대상 모두에서 불필요한 리소스 사용이 발생합니다.

ONTAP 스토리지를 사용하여 NVMe-oF에 대한 RHEL 8.x 구성

Red Hat Enterprise Linux(RHEL) 호스트는 비대칭 네임스페이스 액세스(ANA)를 통해 NVMe over Fibre Channel(NVMe/FC) 및 NVMe over TCP(NVMe/TCP) 프로토콜을 지원합니다. ANA는 iSCSI 및 FCP 환경에서 ALUA(비대칭 논리 장치 액세스)와 동일한 다중 경로 기능을 제공합니다.

RHEL 8.x에 대해 NVMe over Fabrics(NVMe-oF) 호스트를 구성하는 방법을 알아보세요. 추가 지원 및 기능 정보는 다음을 참조하세요. "[RHEL ONTAP 지원 및 기능](#)".

RHEL 8.x를 사용하는 NVMe-oF에는 다음과 같은 알려진 제한 사항이 있습니다.

- NVMe-oF 프로토콜을 사용한 SAN 부팅은 현재 지원되지 않습니다.
- RHEL 8.x의 NVMe-oF 호스트에서는 커널 내부 NVMe 멀티패스가 기본적으로 비활성화되어 있습니다. 수동으로 활성화해야 합니다.
- 알려진 문제로 인해 NVMe/TCP는 기술 미리보기로 제공됩니다.

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: RHEL 및 NVMe 소프트웨어 설치 및 구성 확인

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

단계

1. 서버에 RHEL 8.x를 설치합니다. 설치가 완료되면 필요한 RHEL 8.x 커널을 실행하고 있는지 확인하세요.

```
uname -r
```

RHEL 커널 버전 예:

```
4.18.0-553.el8_10.x86_64
```

2. "NVMe-CLI" 패키지를 설치합니다.

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

다음 예에서는 nvme-cli 패키지 버전을 보여줍니다.

```
nvme-cli-1.16-9.el8.x86_64
```

3. 를 설치합니다 libnvme 패키지:

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

다음 예제는 libnvme 패키지 버전을 보여줍니다.

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

4. 인커널 NVMe 다중 경로 지원:

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

5. RHEL 8.x 호스트에서 다음을 확인하세요. hostnqn 문자열 /etc/nvme/hostnqn :

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

다음 예에서는 다음을 보여줍니다. hostnqn 버전:

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

6. ONTAP 시스템에서 다음을 확인하십시오. hostnqn 문자열이 일치합니다 hostnqn ONTAP 스토리지 시스템의 해당 하위 시스템에 대한 문자열:

```
::> vserver nvme subsystem host show -vserver vs_fc_nvme_141
```

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



만약 `hostnqn` 문자열이 일치하지 않으면 다음을 사용하세요. `vserver modify` 업데이트 명령 `hostnqn` 해당 ONTAP 스토리지 시스템 하위 시스템의 문자열을 일치시키세요. `hostnqn` 문자열에서 `/etc/nvme/hostnqn` 호스트에서.

7. 호스트를 재부팅합니다.

동일한 호스트에서 NVMe와 SCSI 트래픽을 모두 실행하려면 NetApp ONTAP 네임스페이스에 커널 내 NVMe 멀티패스를 사용하고 ONTAP LUN에 `dm-multipath`를 사용할 것을 권장합니다. `dm-multipath`가 ONTAP 네임스페이스 장치를 청구하지 못하도록 하려면 다음을 추가하여 제외합니다. `enable_foreign` 설정 `/etc/multipath.conf` 파일:



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

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 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.

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

b. 모델 설명을 표시합니다.

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

다음 예와 비슷한 출력이 표시되어야 합니다.

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

이 명령은 펌웨어 버전을 반환합니다.

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

b. 받은 편지함 드라이버 버전을 표시합니다.

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

다음 예에서는 드라이버 버전을 보여줍니다.

```
0:14.0.0.21
```

지원되는 어댑터 드라이버 및 펌웨어 버전의 현재 목록은 ["상호 운용성 매트릭스 툴"](#)을 참조하십시오.

3. 확인합니다 lpfc_enable_fc4_type 가 로 설정되어 있습니다 3:

```
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 보기

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

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

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

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

NVMe/FC - Marvell/QLogic

Marvell/QLogic 어댑터용 NVMe/FC를 구성합니다.

단계

1. 지원되는 어댑터 드라이버와 펌웨어 버전을 사용하고 있는지 확인하세요.

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

다음 예에서는 드라이버와 펌웨어 버전을 보여줍니다.

```
QLE2742 FW: v9.10.11 DVR: v10.02.08.200-k  
QLE2742 FW: v9.10.11 DVR: v10.02.08.200-k
```

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. 지원되는 NVMe/TCP LIF를 통해 개시자 포트가 검색 로그 페이지 데이터를 가져올 수 있는지 확인하세요.

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

예제 보기

```
nvme discover -t tcp -w 192.168.2.31 -a 192.168.2.25

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

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

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

4단계: 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. lpfc의 값 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
```

5단계: 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. 각 ONTAP 네임스페이스에 대한 적절한 NVMe-oF 설정(예: NetApp ONTAP 컨트롤러로 설정된 모델 및 라운드 로빈으로 설정된 로드 밸런싱 IPolicy가 호스트에 올바르게 반영되는지 확인합니다.

- 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.

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

3. 호스트에서 네임스페이스가 생성되고 올바르게 검색되는지 확인합니다.

```
nvme list
```

Node	SN	Model	

/dev/nvme4n1	81Ix2BVuekWcAAAAAAB	NetApp ONTAP Controller	
Namespace Usage	Format	FW	Rev

1	21.47 GB / 21.47 GB	4 KiB + 0 B	FFFFFFFF

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

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

NVMe/FC 예시 보기

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


NVMe/TCP 예제 보기

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

5. 호스트에서 네임스페이스가 생성되고 올바르게 검색되는지 확인합니다.

```
nvme list
```

예제 보기

Node	SN	Model		

/dev/nvme4n1	81Ix2BVuekWcAAAAAAB	NetApp ONTAP Controller		
Namespace	Usage	Format	FW	Rev

1		21.47 GB / 21.47 GB	4 KiB + 0 B	FFFFFFFF

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

명

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

예제 보기

Device	Vserver	Namespace Path
/dev/nvme0n1	tcpiscsi_129	/vol/tcpcnvme_1_0_0/tcpcnvme_ns

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

JSON을 참조하십시오

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

예제 보기

```
{
  "ONTAPdevices": [
    {
      "Device": "/dev/nvme0n1",
      "Vserver": "tcpiscsi_129",
      "Namespace Path": "/vol/tcpcnvme_1_0_0/tcpcnvme_ns ",
      "NSID": 1,
      "UUID": " 05c2c351-5d7f-41d7-9bd8-1a56c160c80b ",
      "Size2": "21.47GB",
      "LBA_Data_Size": 4096,
      "Namespace Size" : 5242880
    },
  ]
}
```

6단계: 알려진 문제를 검토합니다

알려진 문제는 다음과 같습니다.

NetApp 버그 ID	제목	설명
"1479047"	RHEL 8.x NVMe-oF 호스트는 중복된 PDC(지속적 검색 컨트롤러)를 생성합니다.	NVMe-oF 호스트에서는 "nvme discover -p" 명령을 사용하여 PDC를 생성할 수 있습니다. 이 명령을 사용하면 개시자-대상 조합당 하나의 PDC만 생성되어야 합니다. 하지만 NVMe-oF 호스트에서 RHEL 8.x를 실행하는 경우 "nvme discover -p"를 실행할 때마다 중복된 PDC가 생성됩니다. 이로 인해 호스트와 대상 모두에서 불필요한 리소스 사용이 발생합니다.

저작권 정보

Copyright © 2026 NetApp, Inc. All Rights Reserved. 미국에서 인쇄된 본 문서의 어떠한 부분도 저작권 소유자의 사전 서면 승인 없이는 어떠한 형식이나 수단(복사, 녹음, 녹화 또는 전자 검색 시스템에 저장하는 것을 비롯한 그래픽, 전자적 또는 기계적 방법)으로도 복제될 수 없습니다.

NetApp이 저작권을 가진 자료에 있는 소프트웨어에는 아래의 라이선스와 고지사항이 적용됩니다.

본 소프트웨어는 NetApp에 의해 '있는 그대로' 제공되며 상품성 및 특정 목적에의 적합성에 대한 명시적 또는 묵시적 보증을 포함하여(이에 제한되지 않음) 어떠한 보증도 하지 않습니다. NetApp은 대체품 또는 대체 서비스의 조달, 사용 불능, 데이터 손실, 이익 손실, 영업 중단을 포함하여(이에 국한되지 않음), 이 소프트웨어의 사용으로 인해 발생하는 모든 직접 및 간접 손해, 우발적 손해, 특별 손해, 징벌적 손해, 결과적 손해의 발생에 대하여 그 발생 이유, 책임론, 계약 여부, 엄격한 책임, 불법 행위(과실 또는 그렇지 않은 경우)와 관계없이 어떠한 책임도 지지 않으며, 이와 같은 손실의 발생 가능성이 통지되었다 하더라도 마찬가지입니다.

NetApp은 본 문서에 설명된 제품을 언제든지 예고 없이 변경할 권리를 보유합니다. NetApp은 NetApp의 명시적인 서면 동의를 받은 경우를 제외하고 본 문서에 설명된 제품을 사용하여 발생하는 어떠한 문제에도 책임을 지지 않습니다. 본 제품의 사용 또는 구매의 경우 NetApp에서는 어떠한 특허권, 상표권 또는 기타 지적 재산권이 적용되는 라이선스도 제공하지 않습니다.

본 설명서에 설명된 제품은 하나 이상의 미국 특허, 해외 특허 또는 출원 중인 특허로 보호됩니다.

제한적 권리 표시: 정부에 의한 사용, 복제 또는 공개에는 DFARS 252.227-7013(2014년 2월) 및 FAR 52.227-19(2007년 12월)의 기술 데이터-비상업적 품목에 대한 권리(Rights in Technical Data -Noncommercial Items) 조항의 하위 조항 (b)(3)에 설명된 제한사항이 적용됩니다.

여기에 포함된 데이터는 상업용 제품 및/또는 상업용 서비스(FAR 2.101에 정의)에 해당하며 NetApp, Inc.의 독점 자산입니다. 본 계약에 따라 제공되는 모든 NetApp 기술 데이터 및 컴퓨터 소프트웨어는 본질적으로 상업용이며 개인 비용만으로 개발되었습니다. 미국 정부는 데이터가 제공된 미국 계약과 관련하여 해당 계약을 지원하는 데에만 데이터에 대한 전 세계적으로 비독점적이고 양도할 수 없으며 재사용이 불가능하며 취소 불가능한 라이선스를 제한적으로 가집니다. 여기에 제공된 경우를 제외하고 NetApp, Inc.의 사전 서면 승인 없이는 이 데이터를 사용, 공개, 재생산, 수정, 수행 또는 표시할 수 없습니다. 미국 국방부에 대한 정부 라이선스는 DFARS 조항 252.227-7015(b)(2014년 2월)에 명시된 권한으로 제한됩니다.

상표 정보

NETAPP, NETAPP 로고 및 <http://www.netapp.com/TM>에 나열된 마크는 NetApp, Inc.의 상표입니다. 기타 회사 및 제품 이름은 해당 소유자의 상표일 수 있습니다.