



SUSE Linux Enterprise Server

SAN hosts and cloud clients

NetApp
December 06, 2024

목차

SUSE Linux Enterprise Server	1
SUSE Linux Enterprise Server 15	1

SUSE Linux Enterprise Server

SUSE Linux Enterprise Server 15

ONTAP를 사용하는 SUSE Linux Enterprise Server 15 SP6용 NVMe-oF 호스트 구성

NVMe/FC(NVMe over Fibre Channel) 및 기타 전송을 포함한 NVMe-oF(NVMe over Fabrics)는 ANA(Asymmetric Namespace Access)가 있는 SUSE Linux Enterprise Server 15 SP6에 대해 지원됩니다. NVMe-oF 환경의 경우, ANA는 iSCSI 및 FCP 환경에서 ALUA 다중 경로와 동일하며 커널 내 NVMe 다중 경로를 통해 구현됩니다.

ONTAP를 사용하는 SUSE Linux Enterprise Server 15 SP6의 NVMe-oF 호스트 구성에 대해 다음 지원을 사용할 수 있습니다.

- 동일한 기존 호스트에서 NVMe 및 SCSI 트래픽 실행 예를 들어, SCSI LUN에 대해 SCSI 장치에 대해 dm-multipath를 구성하고 NVMe multipath를 사용하여 호스트에 NVMe-oF 네임스페이스 장치를 구성할 수 mpath 있습니다.
- NVMe over TCP(NVMe/TCP) 및 NVMe/FC 지원 이 경우 네이티브 패키지의 NetApp 플러그인에 `nvme-cli` NVMe/FC 및 NVMe/TCP 네임스페이스 모두에 대한 ONTAP 세부 정보를 표시할 수 있습니다.

지원되는 구성에 대한 자세한 내용은 를 참조하십시오 "[NetApp 상호 운용성 매트릭스 툴](#)".

피쳐

- NVMe 보안 대역 내 인증 지원
- 고유한 검색 NQN을 사용하여 영구 검색 컨트롤러(PDC)를 지원합니다
- NVMe/TCP에 대한 TLS 1.3 암호화 지원

알려진 제한 사항

- 현재 NVMe-oF 프로토콜을 사용한 SAN 부팅은 지원되지 않습니다.
- SUSE Linux Enterprise Server 15 SP6 호스트에서는 NetApp `sanlun` 호스트 유틸리티를 NVMe-oF에 사용할 수 없습니다. 대신 모든 NVMe-oF 전송에 대해 기본 패키지에 포함된 NetApp 플러그인을 사용할 수 `nvme-cli` 있습니다.

NVMe/FC 구성

ONTAP 구성을 사용하는 SUSE Linux Enterprise Server 15 SP6의 경우 Broadcom/Emulex FC 또는 Marvell/Qlogic FC 어댑터를 사용하여 NVMe/FC를 구성할 수 있습니다.

Broadcom/Emulex

Broadcom/Emulex FC 어댑터용 NVMe/FC를 구성합니다.

단계

1. 권장 어댑터 모델을 사용하고 있는지 확인합니다.

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

예제 출력

```
LPe32002 M2  
LPe32002-M2
```

2. 어댑터 모델 설명을 확인합니다.

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

예제 출력

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

3. 권장되는 Emulex HBA(호스트 버스 어댑터) 펌웨어 버전을 사용하고 있는지 확인합니다.

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

예제 출력

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

4. 권장되는 lpfc 드라이버 버전을 사용하고 있는지 확인합니다.

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

예제 출력

```
0:14.4.0.1
```

5. 이니시에이터 포트를 볼 수 있는지 확인합니다.

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

예제 출력

```
0x10000090fae0ec88  
0x10000090fae0ec89
```

6. 이니시에이터 포트가 온라인 상태인지 확인합니다.

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

예제 출력

```
Online  
Online
```

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

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

다음 예에서는 이니시에이터 포트 하나가 사용하도록 설정되고 두 개의 타겟 LIF로 연결됩니다.

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 x2070d039ea359e4a WWNN x206bd039ea359e4a DID
x0a0a05 TARGET DISCSRV
ONLINE
NVME Statistics
LS: Xmt 00000003ba Cmpl 00000003ba Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 0000000014e3dfb8 Issue 0000000014e308db OutIO
ffffffffffff2923
  abort 00000845 noxri 00000000 nondlp 00000063 qdepth 00000000
wqerr 00000003 err 00000000
FCP CMPL: xb 00000847 Err 00027f33
NVME Initiator Enabled
XRI Dist lpfc1 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc1 WWPN x10000090fae0ec89 WWNN x20000090fae0ec89
DID x0a1200 ONLINE
NVME RPORT WWPN x2071d039ea359e4a WWNN x206bd039ea359e4a DID
x0a0305 TARGET DISCSRV
ONLINE
NVME Statistics
LS: Xmt 00000003ba Cmpl 00000003ba Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 0000000014e39f78 Issue 0000000014e2b832 OutIO
ffffffffffff18ba
  abort 0000082d noxri 00000000 nondlp 00000028 qdepth 00000000
wqerr 00000007 err 00000000
FCP CMPL: xb 0000082d Err 000283bb
```

Marvell/QLogic

SUSE Linux Enterprise Server 15 SP6 커널에 포함된 기본 받은 편지함 qla2xxx 드라이버에는 최신 수정 사항이 포함되어 있습니다. 이러한 수정 사항은 ONTAP 지원에 필수적입니다.

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
```

예제 출력

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

2. 를 확인합니다 `ql2xnvmeenable` 매개 변수는 1로 설정됩니다.

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

예상 값은 1입니다.

1MB I/O 크기 활성화(옵션)

ONTAP는 컨트롤러 식별 데이터에서 MDTs(MAX Data 전송 크기)를 8로 보고합니다. 이는 최대 I/O 요청 크기가 1MB까지 될 수 있음을 의미합니다. Broadcom NVMe/FC 호스트에 대해 1MB 크기의 I/O 요청을 발행하려면 매개 변수 값을 `lpfc_sg_seg_cnt` 기본값인 64에서 256으로 늘려야 합니다.



다음 단계는 Qlogic NVMe/FC 호스트에는 적용되지 않습니다.

단계

1. `lpfc_sg_seg_cnt` 매개 변수를 256으로 설정합니다.

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

```
options lpfc lpfc_sg_seg_cnt=256
```

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

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

예상 값은 256입니다.

NVMe 서비스 확인

SUSE Linux Enterprise Server 15 SP6부터 `nvme-fc-boot-connections.service` NVMe/FC 패키지에 포함된 및 `nvme-fc-autoconnect.service` 부팅 서비스가 `nvme-cli` 시스템 부팅 중에 자동으로 시작됩니다. 시스템 부팅이 완료되면 부팅 서비스가 활성화되었는지 확인해야 합니다.

단계

1. 가 활성화되어 있는지 `nvme-fc-autoconnect.service` 확인합니다.

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

예제 출력을 표시합니다

```
nvme-autoconnect.service - Connect NVMe-oF subsystems automatically
during boot
   Loaded: loaded (/usr/lib/systemd/system/nvme-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]: nvme-autoconnect.service: Deactivated successfully.
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

```
nvme-fc-boot-connections.service - Auto-connect to subsystems on FC-
NVME devices found during boot
   Loaded: loaded (/usr/lib/systemd/system/nvme-fc-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]: nvme-fc-boot-connections.service: Succeeded.
systemd[1]: Finished Auto-connect to subsystems on FC-NVME devices
found during boot.
```


NVMe/TCP를 구성합니다

NVMe/TCP에는 자동 연결 기능이 없습니다. 대신 NVMe/TCP 또는 `connect-all` 작업을 수동으로 수행하여 NVMe/TCP 하위 시스템과 네임스페이스를 검색할 수 `connect` 있습니다.

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

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

```

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

```

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 출력

```

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

```

3. 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 출력

```

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

```



SUSE Linux Enterprise Server 15 SP6부터 NVMe/TCP 시간 제한에 대한 기본 설정이 `ctrl-loss-tmo` 꺼집니다. 다시 시도 횟수에 제한이 없으며(무제한 다시 시도) 또는 `nvme connect-all` 명령(옵션 `-l`)을 사용할 때 특정 시간 초과 기간을 `nvme connect` 수동으로 구성할 필요가 `ctrl-loss-tmo` 없습니다. 또한 NVMe/TCP 컨트롤러는 경로 장애 발생 시 시간 초과가 발생하지 않으며 무기한 연결된 상태를 유지합니다.

NVMe-oF를 검증합니다

다음 절차에 따라 ONTAP 구성이 포함된 SUSE Linux Enterprise Server 15 SP6의 NVMe-oF를 검증합니다.

단계

1. 커널 내 NVMe 다중 경로가 활성화되었는지 확인:

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

예상 값은 "Y"입니다.

2. 호스트에 ONTAP NVMe 네임스페이스에 대한 올바른 컨트롤러 모델이 있는지 확인합니다.

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

예제 출력

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

3. 해당 ONTAP NVMe I/O 컨트롤러에 대한 NVMe I/O 정책을 확인합니다.

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

예제 출력

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

4. ONTAP 네임스페이스가 호스트에 표시되는지 확인합니다.

```
nvme list -v
```

예제 출력을 표시합니다

```
Subsystem          Subsystem-NQN
Controllers
-----
-----
nvme-subsys0      nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.unidir_d
hcha p nvme0, nvme1, nvme2, nvme3

Device    SN                      MN
FR        TxPort Address          Subsystem    Namespaces
-----
-----
nvme0     81LGgBUqsI3EAAAAAAAAE NetApp ONTAP Controller  FFFFFFFF tcp
traddr=192.168.111.66,trsvcid=4420,host_traddr=192.168.111.79 nvme-
subsys0 nvme0n1
nvme1     81LGgBUqsI3EAAAAAAAAE NetApp ONTAP Controller  FFFFFFFF tcp
traddr=192.168.111.67,trsvcid=4420,host_traddr=192.168.111.79 nvme-
subsys0 nvme0n1
nvme2     81LGgBUqsI3EAAAAAAAAE NetApp ONTAP Controller  FFFFFFFF tcp
traddr=192.168.211.66,trsvcid=4420,host_traddr=192.168.211.79 nvme-
subsys0 nvme0n1
nvme3     81LGgBUqsI3EAAAAAAAAE NetApp ONTAP Controller  FFFFFFFF tcp
traddr=192.168.211.67,trsvcid=4420,host_traddr=192.168.211.79 nvme-
subsys0 nvme0n1
Device      Generic    NSID      Usage      Format
Controllers
-----
-----
/dev/nvme0n1 /dev/ng0n1 0x1      1.07 GB / 1.07 GB 4 KiB +
0 B nvme0, nvme1, nvme2, nvme3
```

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

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

NVMe/FC

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

예제 출력을 표시합니다

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

NVMe/TCP

```
nvme list-subsys
```

예제 출력을 표시합니다

```
nvme-subsys1 - NQN=nqn.1992-08.com.netapp:sn.8b5ee9199ff411eea468d039ea36a106:subsystem.nvme_tcp_1
  hostnqn=nqn.2014-08.org.nvmexpress:uuid:4c4c4544-0035-5910-804b-b2c04f444d33
  iopolicy=round-robin
\
+- nvme4 tcp
traddr=192.168.111.66,trsvcid=4420,host_traddr=192.168.111.79,src_addr=192.168.111.79 live
+- nvme3 tcp
traddr=192.168.211.66,trsvcid=4420,host_traddr=192.168.211.79,src_addr=192.168.111.79 live
+- nvme2 tcp
traddr=192.168.111.67,trsvcid=4420,host_traddr=192.168.111.79,src_addr=192.168.111.79 live
+- nvme1 tcp
traddr=192.168.211.67,trsvcid=4420,host_traddr=192.168.211.79,src_addr=192.168.111.79 live
```

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

명

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

예제 출력

Device	Vserver	Namespace Path	Size
NSID UUID			

/dev/nvme0n1	vs_192	/vol/fcnvme_vol_1_1_0/fcnvme_ns	1
c6586535-da8a-40fa-8c20-759ea0d69d33		20GB	

JSON을 참조하십시오

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

예제 출력을 표시합니다

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

영구 검색 컨트롤러를 만듭니다

ONTAP 9.11.1부터 SUSE Linux Enterprise Server 15 SP6 호스트에 대한 영구 검색 컨트롤러(PDC)를 생성할 수 있습니다. PDC는 NVMe 하위 시스템의 추가 또는 제거 작업과 검색 로그 페이지 데이터의 변경 사항을 자동으로 감지하기 위해 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.8b5ee9199ff411eea468d039ea36a106:discovery
traddr: 192.168.211.67
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.8b5ee9199ff411eea468d039ea36a106:discovery
traddr: 192.168.111.67
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.8b5ee9199ff411eea468d039ea36a106:discovery
traddr: 192.168.211.66
eflags: explicit discovery connections, duplicate discovery
information
sectype: none
=====Discovery Log Entry 3=====
trtype: tcp
adrfam: ipv4
```

```
subtype: current discovery subsystem
treql: not specified
portid: 1
trsvcid: 8009
subnqn: nqn.1992-
08.com.netapp:sn.8b5ee9199ff411eea468d039ea36a106:discovery
traddr: 192.168.111.66
eflags: explicit discovery connections, duplicate discovery
information
sectype: none
====Discovery Log Entry 4====
trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treql: not specified
portid: 4
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.8b5ee9199ff411eea468d039ea36a106:subsystem.nvme_tcp
_1
traddr: 192.168.211.67
eflags: none
sectype: none
====Discovery Log Entry 5====
trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treql: not specified
portid: 2
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.8b5ee9199ff411eea468d039ea36a106:subsystem.nvme_tcp
_1
traddr: 192.168.111.67
eflags: none
sectype: none
====Discovery Log Entry 6====
trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treql: not specified
portid: 3
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.8b5ee9199ff411eea468d039ea36a106:subsystem.nvme_tcp
_1
```

```
traddr: 192.168.211.66
eflags: none
sectype: none
=====Discovery Log Entry 7=====
trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 1
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.8b5ee9199ff411eea468d039ea36a106:subsystem.nvme_tcp
_1
traddr: 192.168.111.66
eflags: none
sectype: none
```

2. 검색 하위 시스템에 대한 PDC 생성:

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

예제 출력

```
nvme discover -t tcp -w 192.168.111.79 -a 192.168.111.666 -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_nvme79
Vserver Name: vs_CLIENT116 Controller ID: 00C0h
Discovery Subsystem NQN: nqn.1992-08.com.netapp:sn.48391d66c0a611ecaaa5d039ea165514:discovery Logical
Interface UUID: d23cbb0a-c0a6-11ec-9731-d039ea165abc Logical
Interface:
CLIENT116_lif_4a_1
Node: A400-14-124
Host NQN: nqn.2014-08.org.nvmexpress:uuid:12372496-59c4-4d1b-be09-74362c0c1afc
Transport Protocol: nvme-tcp
Initiator Transport Address: 192.168.1.16
Host Identifier: 59de25be738348f08a79df4bce9573f3 Admin Queue Depth:
32
Header Digest Enabled: false Data Digest Enabled: false
Vserver UUID: 48391d66-c0a6-11ec-aaa5-d039ea165514
```

보안 대역내 인증을 설정합니다

ONTAP 9.12.1부터 SUSE Linux Enterprise Server 15 SP6 호스트와 ONTAP 컨트롤러 간에 NVMe/TCP 및 NVMe/FC를 통해 보안 인밴드 인증이 지원됩니다.

보안 인증을 설정하려면 각 호스트 또는 컨트롤러가 에 연결되어 있어야 합니다 DH-HMAC-CHAP 키 - NVMe 호스트 또는 컨트롤러의 NQN과 관리자가 구성한 인증 비밀의 조합입니다. 피어를 인증하려면 NVMe 호스트 또는 컨트롤러가 피어와 연결된 키를 인식해야 합니다.

CLI 또는 구성 JSON 파일을 사용하여 보안 대역 내 인증을 설정할 수 있습니다. 서로 다른 하위 시스템에 대해 다른 dhchap 키를 지정해야 하는 경우 구성 JSON 파일을 사용해야 합니다.

CLI를 참조하십시오

CLI를 사용하여 보안 인밴드 인증을 설정합니다.

단계

1. 호스트 NQN 가져오기:

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

2. SUSE Linux Enterprise Server 15 SP6 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:d3ca725a- ac8d-4d88-b46a-174ac235139b
DHHC-
1:03:J2UJQfj9f0pLnpF/ASDJRTyILKJRr5CougGpGdQSysPrLu6RW1fG15VSjbeDF1n
1DEh3nVBe19nQ/LxreSBeH/bx/pU=:
```

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-
1:03:je1nQCmjJLUKD62mpYbz1puw00Iws86NB96uNO/t3jbvhp7fjyR9bIRjO
Hg8wQtYe1JCFsmkBQH3pTKGdYR1OV9gx00=:
DHHC-
1:03:je1nQCmjJLUKD62mpYbz1puw00Iws86NB96uNO/t3jbvhp7fjyR9bIRjO
Hg8wQtYe1JCFsmkBQH3pTKGdYR1OV9gx00=:
DHHC-
1:03:je1nQCmjJLUKD62mpYbz1puw00Iws86NB96uNO/t3jbvhp7fjyR9bIRjO
Hg8wQtYe1JCFsmkBQH3pTKGdYR1OV9gx00=:
DHHC-
1:03:je1nQCmjJLUKD62mpYbz1puw00Iws86NB96uNO/t3jbvhp7fjyR9bIRjO
Hg8wQtYe1JCFsmkBQH3pTKGdYR1OV9gx00=:
```

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:WorVEV83eY053kV4Iel50pphbX5LAph03F8fgH3913t1rkSGDBJTt3crX  
eTUB8fCwGbPsEyz6CXxdQJi6kbn4IzmkFU=:  
DHHC-  
1:03:WorVEV83eY053kV4Iel50pphbX5LAph03F8fgH3913t1rkSGDBJTt3crX  
eTUB8fCwGbPsEyz6CXxdQJi6kbn4IzmkFU=:  
DHHC-  
1:03:WorVEV83eY053kV4Iel50pphbX5LAph03F8fgH3913t1rkSGDBJTt3crX  
eTUB8fCwGbPsEyz6CXxdQJi6kbn4IzmkFU=:  
DHHC-  
1:03:WorVEV83eY053kV4Iel50pphbX5LAph03F8fgH3913t1rkSGDBJTt3crX  
eTUB8fCwGbPsEyz6CXxdQJi6kbn4IzmkFU=:
```

JSON 파일

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

JSON 파일을 생성하려면 `-o` 옵션을 사용합니다. 자세한 구문 옵션은 NVMe Connect - 모든 설명서 페이지를 참조하십시오.

단계

1. JSON 파일 구성:

예제 출력을 표시합니다

```
# cat /etc/nvme/config.json
[
  {
    "hostnqn": "nqn.2014-08.org.nvmexpress:uuid:12372496-59c4-4d1b-be09-74362c0c1afc",
    "hostid": "3ae10b42-21af-48ce-a40b-cfb5bad81839",
    "dhchap_key": "DHHC-1:03:Cu3ZZfIz1Wm1qZFncMqpAgn/T6EVOcIFHez215U+Pow8jTgBF2UbNk3DK4wfk2EptWpna1rpwG5CndpOgxpRxxh9m41w="
  },
  {
    "hostnqn": "nqn.2014-08.org.nvmexpress:uuid:12372496-59c4-4d1b-be09-74362c0c1afc",
    "subsystems": [
      {
        "nqn": "nqn.1992-08.com.netapp:sn.48391d66c0a611e00000000000000000:subsystem.subsys_CLIENT116",
        "ports": [
          {
            "transport": "tcp",
            "traddr": " 192.168.111.66 ",
            "host_traddr": " 192.168.111.79",
            "trsvcid": "4420",
            "dhchap_ctrl_key": "DHHC-1:01:0h58bcT/uu0rCpGsDYU6ZHZvRuVqsYKuBRS0Nu0VPx5HEwaZ:"
          },
          {
            "transport": "tcp",
            "traddr": " 192.168.111.66 ",
            "host_traddr": " 192.168.111.79",
            "trsvcid": "4420",
            "dhchap_ctrl_key": "DHHC-1:01:0h58bcT/uu0rCpGsDYU6ZHZvRuVqsYKuBRS0Nu0VPx5HEwaZ:"
          },
          {
            "transport": "tcp",
            "traddr": " 192.168.111.66 ",
            "host_traddr": " 192.168.111.79",
            "trsvcid": "4420",
            "dhchap_ctrl_key": "DHHC-1:01:0h58bcT/uu0rCpGsDYU6ZHZvRuVqsYKuBRS0Nu0VPx5HEwaZ:"
          }
        ]
      }
    ]
  }
]
```

```

        {
            "transport": "tcp",
            "traddr": " 192.168.111.66 ",
            "host_traddr": " 192.168.111.79",
            "trsvcid": "4420",
            "dhchap_ctrl_key": "DHHC-
1:01:0h58bcT/uu0rCpGsDYU6ZHZvRuVqsYKuBRS0Nu0VPx5HEwaZ:"
        }
    ]
}
]
]

```

를 누릅니다



위의 예제에서는 dhchap_key 에 해당하고 에 dhchap_secret dhchap_ctrl_key dhchap_ctrl_secret 해당합니다.

2. config JSON 파일을 사용하여 ONTAP 컨트롤러에 연결합니다.

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

예제 출력을 표시합니다

```

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

```

3. 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
```

예제 출력

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

b. 컨트롤러 dhchap 키를 확인합니다.

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

예제 출력

```
DHHC-  
1:03:2YJinsxa2v3+m8qqCiTnmgBZoH6mIT6G/6f0aGO8viVZB4VLNLH4z8CvK7pV  
YxN6S5fOAtaU3Dni12rieRMfdbg3704=:
```

전송 계층 보안을 구성합니다

TLS(전송 계층 보안)는 NVMe-oF 호스트와 ONTAP 어레이 간 NVMe 연결을 위한 안전한 엔드 투 엔드 암호화를 제공합니다. ONTAP 9.16.1부터 CLI 및 구성된 사전 공유 키(PSK)를 사용하여 TLS 1.3을 구성할 수 있습니다.

이 작업에 대해

SUSE Linux Enterprise Server 15 SP6 호스트에서 이 절차의 단계를 수행합니다. 단, ONTAP 컨트롤러에서 단계를 수행하도록 지정된 경우는 예외입니다.

단계

1. 호스트에 다음 `ktls-utils`, `openssl` 및 `libopenssl` 패키지가 설치되어 있는지 확인합니다.

a. `rpm -qa | grep ktls`

예제 출력

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

b. `rpm -qa | grep ssl`

예제 출력

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

2. 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. # cat /etc/nvme/hostnqn

예제 출력

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

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

예제 출력

```
NVMeTLSkey-1:01:dNcby017axByCko8GivzOO9zGlgHDXJCN6KLzvYoA+NpT1uD:
```

6. ONTAP 컨트롤러에서 TLS PSK를 ONTAP 하위 시스템에 추가합니다.

```
# nvme subsystem host add -vserver sles15_tls -subsystem sles15 -host
-nqn nqn.2014-08.org.nvmexpress:uuid:ffa0c815-e28b-4bb1-8d4c-
7c6d5e610bfc -tls-configured-psk NVMeTLSkey-
1:01:dNcby017axByCko8GivzOO9zGlgHDXJCN6KLzvYoA+NpT1uD:
```

7. TLS PSK를 호스트 커널 키링에 삽입합니다.

```
# nvme check-tls-key --identity=1 --subsysnqn=nqn.2014-08.org.nvmexpress:uuid:ffa0c815-e28b-4bb1-8d4c-7c6d5e610bfc --keydata=NVMETLSkey -1:01:dNcby017axByCko8GivzOO9zGlgHDXJCN6KLzvYoA+NpT1uD: --insert
```

예제 출력

```
Inserted TLS key 22152a7e
```



PSK는 TLS 핸드셰이크 알고리즘의 "identity v1"을 사용하기 때문에 "NVMe1R01"으로 표시됩니다. Identity v1은 ONTAP에서 지원하는 유일한 버전입니다.

8. TLS PSK가 올바르게 삽입되었는지 확인합니다.

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

예제 출력

```
22152a7e I--Q---      1 perm 3b010000      0      0 psk      NVMe1R01
nqn.2014-08.org.nvmexpress:uuid:ffa0c815-e28b-4bb1-8d4c-7c6d5e610bfc
nqn.1992-
08.com.netapp:sn.1d59a6b2416b11ef9ed5d039ea50acb3:subsystem.sles15
UoP9dEfvuCUzzpS0DYxnshKDapZYmvA0/RJJ8JAqmAo=: 32
```

9. 삽입된 TLS PSK를 사용하여 ONTAP 하위 시스템에 연결합니다.

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

예제 출력

```
connecting to device: nvme0
```

```
b. # nvme list-subsys
```

예제 출력

```
nvme-subsys0 - NQN=nqn.1992-08.com.netapp:sn.1d59a6b2416b11ef9ed5d039ea50acb3:subsystem.sles15
                hostnqn=nqn.2014-08.org.nvmexpress:uuid:ffa0c815-e28b-4bb1-8d4c-7c6d5e610bfc
                iopolicy=round-robin
\
+- nvme0 tcp
traddr=20.20.10.14,trsvcid=4420,host_traddr=20.20.10.80,src_addr=20.20.10.80 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: sles15_tls
      Subsystem: sles15
      Controller ID: 0040h
      Logical Interface: sles15t_e1a_1
      Node: A900-17-174
      Host NQN: nqn.2014-
08.org.nvmexpress:uuid:ffa0c815-e28b-4bb1-8d4c-7c6d5e610bfc
      Transport Protocol: nvme-tcp
      Initiator Transport Address: 20.20.10.80
      Host Identifier:
ffa0c815e28b4bb18d4c7c6d5e610bfc
      Number of I/O Queues: 4
      I/O Queue Depths: 128, 128, 128, 128
      Admin Queue Depth: 32
      Max I/O Size in Bytes: 1048576
      Keep-Alive Timeout (msec): 5000
      Vserver UUID: 1d59a6b2-416b-11ef-9ed5-
d039ea50acb3
      Subsystem UUID: 9b81e3c5-5037-11ef-8a90-
d039ea50ac83
      Logical Interface UUID: 8185dcac-5035-11ef-8abb-
d039ea50acb3
      Header Digest Enabled: false
      Data Digest Enabled: false
      Authentication Hash Function: -
      Authentication Diffie-Hellman Group: -
      Authentication Mode: none
      Transport Service Identifier: 4420
      TLS Key Type: configured
      TLS PSK Identity: NVMe1R01 nqn.2014-
08.org.nvmexpress:uuid:ffa0c815-e28b-4bb1-8d4c-7c6d5e610bfc
nqn.1992-
08.com.netapp:sn.1d59a6b2416b11ef9ed5d039ea50acb3:subsystem.sles15
UoP9dEfvuCUzzpS0DYxnshKDapZYmvA0/RJJ8JAqmAo=
      TLS Cipher: TLS-AES-128-GCM-SHA256
```

알려진 문제

SUSE Linux Enterprise Server 15 SP6와 ONTAP 릴리스에 대해 알려진 문제는 없습니다.

ONTAP를 사용하는 SUSE Linux Enterprise Server 15 SP5용 NVMe-oF 호스트 구성

NVMe/FC(NVMe over Fibre Channel) 및 기타 전송을 포함한 NVMe-oF(NVMe over Fabrics)는 ANA(Asymmetric Namespace Access)가 있는 SUSE Linux Enterprise Server 15 SP5에 대해 지원됩니다. NVMe-oF 환경의 경우, ANA는 iSCSI 및 FCP 환경에서 ALUA 다중 경로와 동일하며 커널 내 NVMe 다중 경로를 통해 구현됩니다.

ONTAP를 사용하는 SUSE Linux Enterprise Server 15 SP5의 NVMe-oF 호스트 구성에 대해 다음 지원을 사용할 수 있습니다.

- NVMe와 SCSI 트래픽을 모두 동일한 호스트에서 실행할 수 있습니다. 따라서 SCSI LUN의 경우 SCSI mpath 장치에 대해 dm-multipath를 구성할 수 있지만 NVMe multipath를 사용하여 호스트에 NVMe-oF 네임스페이스 장치를 구성할 수 있습니다.
- NVMe/FC 외에 NVMe over TCP(NVMe/TCP) 지원 NetApp 플러그인은 기본 제공됩니다 `nvme-cli` 패키지에는 NVMe/FC 및 NVMe/TCP 네임스페이스에 대한 ONTAP 세부 정보가 표시됩니다.

지원되는 구성에 대한 자세한 내용은 를 참조하십시오 "[NetApp 상호 운용성 매트릭스 툴](#)".

피처

- NVMe 보안 대역 내 인증 지원
- 고유한 검색 NQN을 사용하여 영구 검색 컨트롤러(PDC)를 지원합니다

알려진 제한 사항

- 현재 NVMe-oF 프로토콜을 사용한 SAN 부팅은 지원되지 않습니다.
- ``sanlun`` NVMe-oF를 지원하지 않습니다. 따라서 SUSE Linux Enterprise Server 15 SP5 호스트에서 NVMe-oF에 대한 호스트 유틸리티 지원을 사용할 수 없습니다. 모든 NVMe-oF 전송에 대해 기본 NVMe-CLI 패키지에 포함된 NetApp 플러그인을 사용할 수 있습니다.

NVMe/FC 구성

Broadcom/Emulex FC 또는 Marvell/Qlogic FC 어댑터에 대해 NVMe/FC를 구성할 수 있습니다.

Broadcom/Emulex

단계

1. 권장 어댑터 모델을 사용하고 있는지 확인합니다.

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

◦ 출력 예 *:

```
LPe32002 M2  
LPe32002-M2
```

2. 어댑터 모델 설명을 확인합니다.

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

◦ 출력 예 *:

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

3. 권장되는 Emulex HBA(호스트 버스 어댑터) 펌웨어 버전을 사용하고 있는지 확인합니다.

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

◦ 출력 예 *:

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

4. 권장되는 lpfc 드라이버 버전을 사용하고 있는지 확인합니다.

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

◦ 출력 예 *:

```
0:14.2.0.13
```

5. 이니시에이터 포트를 볼 수 있는지 확인합니다.

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

◦ 출력 예 *:

```
0x100000109b579d5e  
0x100000109b579d5f
```

6. 이니시에이터 포트가 온라인 상태인지 확인합니다.

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

◦ 출력 예 *:

```
Online  
Online
```

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

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

◦ 출력 예 *:

다음 예에서는 이니시에이터 포트 하나가 사용하도록 설정되고 두 개의 타겟 LIF로 연결됩니다.

```

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

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

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

NVME Statistics
LS: Xmt 0000000e50 Cmpl 0000000e50 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000003c9859ca Issue 000000003c93515e OutIO
ffffffffffffaf794
abort 00000b73 noxri 00000000 nondlp 00000000 qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 0000159d Err 000135c3

```

8. 호스트를 재부팅합니다.

Marvell/QLogic

SUSE Linux Enterprise Server 15 SP5 커널에 포함된 기본 받은 편지함 qla2xxx 드라이버에는 최신 수정 사항이 포함되어 있습니다. 이러한 수정 사항은 ONTAP 지원에 필수적입니다.

단계

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

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

◦ 출력 예 *:

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

2. 를 확인합니다 ql2xnvmeenable 매개 변수는 1로 설정됩니다.

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

1MB I/O 크기 활성화(옵션)

ONTAP는 컨트롤러 식별 데이터에서 MDTS(MAX Data 전송 크기)를 8로 보고합니다. 이는 최대 I/O 요청 크기가 1MB까지 될 수 있음을 의미합니다. Broadcom NVMe/FC 호스트에 대해 1MB 크기의 I/O 요청을 발행하려면 매개 변수 값을 lpfc_sg_seg_cnt 기본값인 64에서 256으로 늘려야 lpfc 합니다.



다음 단계는 Qlogic NVMe/FC 호스트에는 적용되지 않습니다.

단계

1. `lpfc_sg_seg_cnt` 매개 변수를 256으로 설정합니다.

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

```
options lpfc lpfc_sg_seg_cnt=256
```

2. `dracut -f` 명령을 실행하고 호스트를 재부팅합니다.

3. 가 lpfc_sg_seg_cnt 256인지 확인합니다.

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

예상 값은 256입니다.

NVMe 서비스 활성화

에는 2가지 NVMe/FC 부팅 서비스가 포함되어 있습니다 nvme-cli 그러나 패키지는 _만_입니다 nvme-fc-boot-connections.service 시스템 부팅 중에 시작하도록 설정되어 있습니다. nvmmf-autoconnect.service 이(가) 활성화되지 않았습니다. 따라서 수동으로 을 사용하도록 설정해야 합니다 nvmmf-autoconnect.service 를

눌러 시스템을 부팅합니다.

단계

1. 활성화 `nvmf-autoconnect.service`:

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

2. 호스트를 재부팅합니다.
3. `nvmf-autoconnect.service` `시스템 부팅 후 및 `nvmefc-boot-connections.service`가 실행 중인지 확인합니다.

◦ 출력 예: *

```

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

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

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

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

```

NVMe/TCP를 구성합니다

다음 절차를 사용하여 NVMe/TCP를 구성할 수 있습니다.

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>
```


◦ 출력 예 *:

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

Discovery Log Number of Records 8, Generation counter 18
====Discovery Log Entry 0==== trtype: tcp
adrfam: ipv4
subtype: current discovery subsystem treq: not specified
portid: 0
trsvcid: 8009 subnqn: nqn.1992-
08.com.netapp:sn.48391d66c0a611ecaaa5d039ea165514:discovery traddr:
192.168.2.117
eflags: explicit discovery connections, duplicate discovery information
sectype: none
====Discovery Log Entry 1==== trtype: tcp
adrfam: ipv4
subtype: current discovery subsystem treq: not specified
portid: 1
trsvcid: 8009 subnqn: nqn.1992-
08.com.netapp:sn.48391d66c0a611ecaaa5d039ea165514:discovery traddr:
192.168.1.117
eflags: explicit discovery connections, duplicate discovery information
sectype: none
====Discovery Log Entry 2==== trtype: tcp
adrfam: ipv4
subtype: current discovery subsystem treq: not specified
portid: 2
trsvcid: 8009 subnqn: nqn.1992-
08.com.netapp:sn.48391d66c0a611ecaaa5d039ea165514:discovery traddr:
192.168.2.116
eflags: explicit discovery connections, duplicate discovery information
sectype: none
====Discovery Log Entry 3==== trtype: tcp
adrfam: ipv4
subtype: current discovery subsystem treq: not specified
portid: 3
trsvcid: 8009 subnqn: nqn.1992-
08.com.netapp:sn.48391d66c0a611ecaaa5d039ea165514:discovery traddr:
192.168.1.116
eflags: explicit discovery connections, duplicate discovery information
sectype: none
====Discovery Log Entry 4==== trtype: tcp
adrfam: ipv4
subtype: nvme subsystem treq: not specified portid: 0
trsvcid: 4420 subnqn: nqn.1992-
08.com.netapp:sn.48391d66c0a611ecaaa5d039ea165514:subsystem.subsys_CLIEN
```

```

T116
traddr: 192.168.2.117 eflags: not specified sectype: none
=====Discovery Log Entry 5===== trtype: tcp
adrfam: ipv4
subtype: nvme subsystem treq: not specified portid: 1
trsvcid: 4420 subnqn: nqn.1992-
08.com.netapp:sn.48391d66c0a611e00000000000000000:subsystem.subsys_CLIEN
T116
traddr: 192.168.1.117 eflags: not specified sectype: none
=====Discovery Log Entry 6===== trtype: tcp
adrfam: ipv4
subtype: nvme subsystem treq: not specified portid: 2
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.48391d66c0a611e00000000000000000:subsystem.subsys_CLIEN
T116
traddr: 192.168.2.116 eflags: not specified sectype: none
=====Discovery Log Entry 7===== trtype: tcp
adrfam: ipv4
subtype: nvme subsystem treq: not specified portid: 3
trsvcid: 4420 subnqn: nqn.1992-
08.com.netapp:sn.48391d66c0a611e00000000000000000:subsystem.subsys_CLIEN
T116
traddr: 192.168.1.116 eflags: not specified sectype: none

```

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>
```

◦ 출력 예: *

```

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

```

3. 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 -l
<ctrl_loss_timeout_in_seconds>
```

◦ 출력 예: *

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

+



NetApp은 를 설정할 것을 권장합니다 ctrl-loss-tmo 옵션을 로 설정합니다 -1 따라서 경로 손실이 발생하면 NVMe/TCP 이니시에이터가 무한정 다시 연결을 시도합니다.

NVMe-oF를 검증합니다

다음 절차를 사용하여 NVMe-oF를 검증할 수 있습니다.

단계

1. 커널 내 NVMe 다중 경로가 활성화되었는지 확인:

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

2. 호스트에 ONTAP NVMe 네임스페이스에 대한 올바른 컨트롤러 모델이 있는지 확인합니다.

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

◦ 출력 예: *

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

3. 해당 ONTAP NVMe I/O 컨트롤러에 대한 NVMe I/O 정책을 확인합니다.

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

◦ 출력 예: *

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

4. ONTAP 네임스페이스가 호스트에 표시되는지 확인합니다.

```
nvme list -v
```

◦ 출력 예: *

```
Subsystem          Subsystem-NQN
Controllers
-----
-----
nvme-subsys0      nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.unidir_dhcha
p  nvme0, nvme1, nvme2, nvme3

Device   SN                      MN
FR       TxPort Adress           Subsystem      Namespaces
-----
-----
nvme0    81LGgBUqsI3EAAAAAAAAE NetApp ONTAP Controller  FFFFFFFF tcp
traddr=192.168.2.214,trsvcid=4420,host_traddr=192.168.2.14 nvme-subsys0
nvme0n1
nvme1    81LGgBUqsI3EAAAAAAAAE NetApp ONTAP Controller  FFFFFFFF tcp
traddr=192.168.2.215,trsvcid=4420,host_traddr=192.168.2.14 nvme-subsys0
nvme0n1
nvme2    81LGgBUqsI3EAAAAAAAAE NetApp ONTAP Controller  FFFFFFFF tcp
traddr=192.168.1.214,trsvcid=4420,host_traddr=192.168.1.14 nvme-subsys0
nvme0n1
nvme3    81LGgBUqsI3EAAAAAAAAE NetApp ONTAP Controller  FFFFFFFF tcp
traddr=192.168.1.215,trsvcid=4420,host_traddr=192.168.1.14 nvme-subsys0
nvme0n1

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

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

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

NVMe/FC

◦ 예제 출력 *

```
# nvme list-subsys /dev/nvme1n1
nvme-subsys1 - NQN=nqn.1992-
08.com.netapp:sn.04ba0732530911ea8e8300a098dfdd91:subsystem.nvme_145
_1
\
+- nvme2 fc traddr=nn-0x208100a098dfdd91:pn-
0x208200a098dfdd91,host_traddr=nn-0x200000109b579d5f:pn-
0x100000109b579d5f live optimized
+- nvme3 fc traddr=nn-0x208100a098dfdd91:pn-
0x208500a098dfdd91,host_traddr=nn-0x200000109b579d5e:pn-
0x100000109b579d5e live optimized
+- nvme4 fc traddr=nn-0x208100a098dfdd91:pn-
0x208400a098dfdd91,host_traddr=nn-0x200000109b579d5e:pn-
0x100000109b579d5e live non-optimized
+- nvme6 fc traddr=nn-0x208100a098dfdd91:pn-
0x208300a098dfdd91,host_traddr=nn-0x200000109b579d5f:pn-
0x100000109b579d5f live non-optimized
```

NVMe/TCP

◦ 예제 출력 *

```
# nvme list-subsys
nvme-subsys0 - NQN=nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.unidir_d
hchap
hostnqn=nqn.2014-08.org.nvmexpress:uuid:e58eca24-faff-11ea-8fee-
3a68dd3b5c5f
iopolicy=round-robin

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

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

명령

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

◦ 출력 예 *:

Device	Vserver	Namespace	Path	Size
NSID	UUID			

/dev/nvme0n1	vs_CLIENT114			
/vol/CLIENT114_vol_0_10/CLIENT114_ns10		1	c6586535-da8a-40fa-8c20-759ea0d69d33	1.07GB

JSON을 참조하십시오

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

◦ 출력 예 *:

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

영구 검색 컨트롤러를 만듭니다

ONTAP 9.11.1부터 SUSE Linux Enterprise Server 15 SP5 호스트에 대한 영구 검색 컨트롤러(PDC)를 만들 수 있습니다. PDC는 NVMe 하위 시스템의 추가 또는 제거 시나리오와 검색 로그 페이지 데이터의 변경 사항을 자동으로 감지하기 위해 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>
```

출력 예 표시:

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

====Discovery Log Entry 7=====

trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 0
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.unidir_n
one
traddr: 192.168.2.214
eflags: none
sectype: none

====Discovery Log Entry 8=====

trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 0
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.subsys_C
LIENT114
traddr: 192.168.1.214
eflags: none
sectype: none

====Discovery Log Entry 9=====

trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 0
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.subsys_C
LIENT114
traddr: 192.168.1.215
eflags: none
sectype: none

====Discovery Log Entry 10=====

trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 0
trsvcid: 4420

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

```

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

```

2. 검색 하위 시스템에 대한 PDC 생성:

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

◦ 출력 예: *

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

3. ONTAP 컨트롤러에서 PDC가 생성되었는지 확인합니다.

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

◦ 출력 예: *

```
vserver nvme show-discovery-controller -instance -vserver vs_nvme175
Vserver Name: vs_CLIENT116 Controller ID: 00C0h
Discovery Subsystem NQN: nqn.1992-08.com.netapp:sn.48391d66c0a611ecaaa5d039ea165514:discovery Logical
Interface UUID: d23cbb0a-c0a6-11ec-9731-d039ea165abc Logical Interface:
CLIENT116_lif_4a_1
Node: A400-14-124
Host NQN: nqn.2014-08.org.nvmexpress:uuid:12372496-59c4-4d1b-be09-
74362c0c1afc
Transport Protocol: nvme-tcp
Initiator Transport Address: 192.168.1.16
Host Identifier: 59de25be738348f08a79df4bce9573f3 Admin Queue Depth: 32
Header Digest Enabled: false Data Digest Enabled: false
Vserver UUID: 48391d66-c0a6-11ec-aaa5-d039ea165514
```

보안 대역내 인증을 설정합니다

ONTAP 9.12.1부터 SUSE Linux Enterprise Server 15 SP5 호스트와 ONTAP 컨트롤러 간에 NVMe/TCP 및 NVMe/FC를 통해 보안 인밴드 인증이 지원됩니다.

보안 인증을 설정하려면 각 호스트 또는 컨트롤러가 에 연결되어 있어야 합니다 DH-HMAC-CHAP 키 - NVMe 호스트 또는 컨트롤러의 NQN과 관리자가 구성한 인증 비밀의 조합입니다. 피어를 인증하려면 NVMe 호스트 또는 컨트롤러가 피어와 연결된 키를 인식해야 합니다.

CLI 또는 구성 JSON 파일을 사용하여 보안 대역 내 인증을 설정할 수 있습니다. 서로 다른 하위 시스템에 대해 다른 dhchap 키를 지정해야 하는 경우 구성 JSON 파일을 사용해야 합니다.

CLI를 참조하십시오

단계

1. 호스트 NQN 가져오기:

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

2. SUSE Linux Enterprise Server 15 SP5 호스트에 대한 dhchap 키를 생성합니다.

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

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 설정의 예제 출력: *

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

b. 컨트롤러 dhchap 키를 확인합니다.

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

▪ 양방향 구성의 출력 예: *

```
# cat /sys/class/nvme-subsystem/nvme-
subsys6/nvme*/dhchap_ctrl_secret
DHHC-
1:03:WorVEV83eY053kV4Iel5OpphbX5LAph03F8fgH3913t1rkSGDBJTt3crXeTUB8f
CwGbPsEyz6CXxdQJi6kbn4IzmkFU=:
DHHC-
1:03:WorVEV83eY053kV4Iel5OpphbX5LAph03F8fgH3913t1rkSGDBJTt3crXeTUB8f
CwGbPsEyz6CXxdQJi6kbn4IzmkFU=:
DHHC-
1:03:WorVEV83eY053kV4Iel5OpphbX5LAph03F8fgH3913t1rkSGDBJTt3crXeTUB8f
CwGbPsEyz6CXxdQJi6kbn4IzmkFU=:
DHHC-
1:03:WorVEV83eY053kV4Iel5OpphbX5LAph03F8fgH3913t1rkSGDBJTt3crXeTUB8f
CwGbPsEyz6CXxdQJi6kbn4IzmkFU=:
```

JSON 파일

를 사용할 수 있습니다 /etc/nvme/config.json 파일이 있는 파일 nvme connect-all ONTAP 컨트롤러 구성에서 여러 NVMe 서브시스템을 사용할 수 있는 경우에 명령을 사용합니다.

을 사용하여 JSON 파일을 생성할 수 있습니다 -o 옵션을 선택합니다. 자세한 구문 옵션은 NVMe connect-all man 페이지를 참조하십시오.

단계

1. JSON 파일 구성:

```
# cat /etc/nvme/config.json
[
  {
    "hostnqn": "nqn.2014-08.org.nvmexpress:uuid:12372496-59c4-4d1b-
be09-74362c0c1afc",
    "hostid": "3ae10b42-21af-48ce-a40b-cfb5bad81839",
    "dhchap_key": "DHHC-
1:03:Cu3ZzfIz1Wm1qZFncMqpAgn/T6EVOcIFHez215U+Pow8jTgBF2UbNk3DK4wfk2E
ptWpna1rpg5CndpOgxpRxxh9m41w=: "
  },
  {
    "hostnqn": "nqn.2014-08.org.nvmexpress:uuid:12372496-59c4-4d1b-
be09-74362c0c1afc",
    "subsystems": [
      {
        "nqn": "nqn.1992-
08.com.netapp:sn.48391d66c0a611ecaaa5d039ea165514:subsystem.subsys_C
LIENT116",
        "ports": [
```



```

    {
        "transport": "tcp",
        "traddr": "192.168.1.117",
        "host_traddr": "192.168.1.16",
        "trsvcid": "4420",
        "dhchap_ctrl_key": "DHHC-
1:01:0h58bcT/uu0rCpGsDYU6ZHZvRuVqsYKuBRS0Nu0VPx5HEwaZ:"
    },
    {
        "transport": "tcp",
        "traddr": "192.168.1.116",
        "host_traddr": "192.168.1.16",
        "trsvcid": "4420",
        "dhchap_ctrl_key": "DHHC-
1:01:0h58bcT/uu0rCpGsDYU6ZHZvRuVqsYKuBRS0Nu0VPx5HEwaZ:"
    },
    {
        "transport": "tcp",
        "traddr": "192.168.2.117",
        "host_traddr": "192.168.2.16",
        "trsvcid": "4420",
        "dhchap_ctrl_key": "DHHC-
1:01:0h58bcT/uu0rCpGsDYU6ZHZvRuVqsYKuBRS0Nu0VPx5HEwaZ:"
    },
    {
        "transport": "tcp",
        "traddr": "192.168.2.116",
        "host_traddr": "192.168.2.16",
        "trsvcid": "4420",
        "dhchap_ctrl_key": "DHHC-
1:01:0h58bcT/uu0rCpGsDYU6ZHZvRuVqsYKuBRS0Nu0VPx5HEwaZ:"
    }
]
}
]

```

[NOTE]

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

2. config JSON 파일을 사용하여 ONTAP 컨트롤러에 연결합니다.

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

◦ 출력 예 *:

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

3. 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
```

▪ 출력 예: *

```
DHHC-1:01:NunEWY7AZlXqxITGheByarwZdQvU4ebZg9H0jIr6nOHEkxJg:
```

b. 컨트롤러 dhchap 키를 확인합니다.

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

▪ 출력 예: *

```
DHHC-
1:03:2YJinsxa2v3+m8qqCiTnmGBZoH6mIT6G/6f0aGO8viVZB4VLNLH4z8CvK7pV
YxN6S5fOAtaU3DNi12rierMfdbg3704=:
```

알려진 문제

SUSE Linux Enterprise Server 15 SP5와 ONTAP 릴리스에 대해 알려진 문제는 없습니다.

ONTAP를 사용하는 SUSE Linux Enterprise Server 15 SP4용 NVMe-oF 호스트 구성

NVMe/FC(NVMe over Fibre Channel) 및 기타 전송을 포함한 NVMe-oF(NVMe over Fabrics)는 SUSE Linux Enterprise Server(SLES) 15 SP4와 ANA(Asymmetric Namespace Access)가 지원됩니다. NVMe-oF 환경에서 ANA는 iSCSI 및 FCP 환경에서 ALUA 다중 경로와 이와 같으며 커널 내 NVMe 다중 경로를 통해 구현됩니다.

ONTAP를 사용하는 SUSE Linux Enterprise Server 15 SP4의 NVMe-oF 호스트 구성에 대해 다음 지원을 사용할 수 있습니다.

- NVMe와 SCSI 트래픽을 모두 동일한 동일한 호스트에서 실행할 수 있습니다. 따라서 SCSI LUN의 경우 SCSI mpath 장치에 대해 dm-multipath를 구성할 수 있지만 NVMe multipath를 사용하여 호스트에 NVMe-oF 네임스페이스 장치를 구성할 수 있습니다.
- NVMe/FC 외에 NVMe over TCP(NVMe/TCP) 지원 네이티브 NVMe-CLI 패키지의 NetApp 플러그인은 NVMe/FC 및 NVMe/TCP 네임스페이스 모두에 대한 ONTAP 세부 정보를 표시합니다.

지원되는 구성에 대한 자세한 내용은 를 참조하십시오 "[NetApp 상호 운용성 매트릭스 툴](#)".

피처

- NVMe 보안 대역 내 인증 지원
- 고유한 검색 NQN을 사용하여 영구 검색 컨트롤러(PDC)를 지원합니다

알려진 제한 사항

- 현재 NVMe-oF 프로토콜을 사용한 SAN 부팅은 지원되지 않습니다.
- NVMe-oF에 대한 sanlun 지원은 없습니다. 따라서 SUSE Linux Enterprise Server 15 SP5 호스트에서 NVMe-oF에 대한 호스트 유틸리티 지원을 사용할 수 없습니다. 모든 NVMe-oF 전송에 대해 기본 NVMe-CLI 패키지에 포함된 NetApp 플러그인을 사용할 수 있습니다.

NVMe/FC 구성

Broadcom/Emulex FC 어댑터 또는 Marvell/Qlogic FC 어댑터에 대해 NVMe/FC를 구성할 수 있습니다.

Broadcom/Emulex

단계

1. 권장 어댑터 모델을 사용하고 있는지 확인합니다.

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

◦ 출력 예 *:

```
LPe32002 M2  
LPe32002-M2
```

2. 어댑터 모델 설명을 확인합니다.

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

◦ 출력 예 *:

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

3. 권장되는 Emulex HBA(호스트 버스 어댑터) 펌웨어 버전을 사용하고 있는지 확인합니다.

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

◦ 출력 예 *:

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

4. 권장되는 lpfc 드라이버 버전을 사용하고 있는지 확인합니다.

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

◦ 출력 예 *:

```
0:14.2.0.6
```

5. 이니시에이터 포트를 볼 수 있는지 확인합니다.

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

◦ 출력 예 *:

```
0x100000109b579d5e  
0x100000109b579d5f
```

6. 이니시에이터 포트가 온라인 상태인지 확인합니다.

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

◦ 출력 예 *:

```
Online  
Online
```

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

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

◦ 출력 예 *:

다음 예에서는 이니시에이터 포트 하나가 사용하도록 설정되고 두 개의 타겟 LIF로 연결됩니다.

```

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

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

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

NVME Statistics
LS: Xmt 0000000e50 Cmpl 0000000e50 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000003c9859ca Issue 000000003c93515e OutIO
ffffffffffffaf794
abort 00000b73 noxri 00000000 nondlp 00000000 qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 0000159d Err 000135c3

```

8. 호스트를 재부팅합니다.

Marvell/QLogic

SUSE Linux Enterprise Server 15 SP4 커널에 포함된 기본 받은 편지함 qla2xxx 드라이버에는 최신 수정 사항이 있습니다. 이러한 수정 사항은 ONTAP 지원에 필수적입니다.

단계

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

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

◦ 출력 예 *:

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

2. 를 확인합니다 ql2xnvmeenable 매개 변수는 1로 설정됩니다.

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

1MB I/O 크기 활성화(옵션)

ONTAP는 컨트롤러 식별 데이터에서 MDTS(MAX Data 전송 크기)를 8로 보고합니다. 이는 최대 I/O 요청 크기가 1MB까지 될 수 있음을 의미합니다. Broadcom NVMe/FC 호스트에 대해 1MB 크기의 I/O 요청을 발행하려면 매개 변수 값을 lpfc_sg_seg_cnt 기본값인 64에서 256으로 늘려야 lpfc 합니다.



다음 단계는 Qlogic NVMe/FC 호스트에는 적용되지 않습니다.

단계

1. `lpfc_sg_seg_cnt` 매개 변수를 256으로 설정합니다.

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

```
options lpfc lpfc_sg_seg_cnt=256
```

2. `dracut -f` 명령을 실행하고 호스트를 재부팅합니다.

3. 가 lpfc_sg_seg_cnt 256인지 확인합니다.

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

예상 값은 256입니다.

NVMe 서비스 활성화

에는 2가지 NVMe/FC 부팅 서비스가 포함되어 있습니다 nvme-cli 그러나 패키지는 _만_입니다 nvme-fc-
connections.service 시스템 부팅 중에 시작하도록 설정되어 있습니다. nvmf-autoconnect.service 이(
가) 활성화되지 않았습니다. 따라서 수동으로 을 사용하도록 설정해야 합니다 nvmf-autoconnect.service 를

눌러 시스템을 부팅합니다.

단계

1. 활성화 `nvmf-autoconnect.service`:

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

2. 호스트를 재부팅합니다.
3. `nvmf-autoconnect.service` `시스템 부팅 후 및 `nvmefc-boot-connections.service`가 실행 중인지 확인합니다.
 - 출력 예: *


```

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

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

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

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

```

NVMe/TCP를 구성합니다

다음 절차를 사용하여 NVMe/TCP를 구성할 수 있습니다.

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>
```

◦ 출력 예 *:

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

Discovery Log Number of Records 8, Generation counter 18
====Discovery Log Entry 0==== trtype: tcp
adrfam: ipv4
subtype: current discovery subsystem treq: not specified
portid: 0
trsvcid: 8009 subnqn: nqn.1992-
08.com.netapp:sn.48391d66c0a611eaaa5d039ea165514:discovery traddr:
192.168.2.117
eflags: explicit discovery connections, duplicate discovery information
sectype: none
====Discovery Log Entry 1==== trtype: tcp
adrfam: ipv4
subtype: current discovery subsystem treq: not specified
portid: 1
trsvcid: 8009 subnqn: nqn.1992-
08.com.netapp:sn.48391d66c0a611eaaa5d039ea165514:discovery traddr:
192.168.1.117
eflags: explicit discovery connections, duplicate discovery information
sectype: none
====Discovery Log Entry 2==== trtype: tcp
adrfam: ipv4
subtype: current discovery subsystem treq: not specified
portid: 2
trsvcid: 8009 subnqn: nqn.1992-
08.com.netapp:sn.48391d66c0a611eaaa5d039ea165514:discovery traddr:
192.168.2.116
eflags: explicit discovery connections, duplicate discovery information
sectype: none
====Discovery Log Entry 3==== trtype: tcp
adrfam: ipv4
subtype: current discovery subsystem treq: not specified
portid: 3
trsvcid: 8009 subnqn: nqn.1992-
08.com.netapp:sn.48391d66c0a611eaaa5d039ea165514:discovery traddr:
192.168.1.116
eflags: explicit discovery connections, duplicate discovery information
sectype: none
====Discovery Log Entry 4==== trtype: tcp
adrfam: ipv4
subtype: nvme subsystem treq: not specified portid: 0
trsvcid: 4420 subnqn: nqn.1992-
08.com.netapp:sn.48391d66c0a611eaaa5d039ea165514:subsystem.subsys_CLIEN
```

```

T116
traddr: 192.168.2.117 eflags: not specified sectype: none
=====Discovery Log Entry 5===== trtype: tcp
adrfam: ipv4
subtype: nvme subsystem treq: not specified portid: 1
trsvcid: 4420 subnqn: nqn.1992-
08.com.netapp:sn.48391d66c0a611e00000000000000000:subsystem.subsys_CLIEN
T116
traddr: 192.168.1.117 eflags: not specified sectype: none
=====Discovery Log Entry 6===== trtype: tcp
adrfam: ipv4
subtype: nvme subsystem treq: not specified portid: 2
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.48391d66c0a611e00000000000000000:subsystem.subsys_CLIEN
T116
traddr: 192.168.2.116 eflags: not specified sectype: none
=====Discovery Log Entry 7===== trtype: tcp
adrfam: ipv4
subtype: nvme subsystem treq: not specified portid: 3
trsvcid: 4420 subnqn: nqn.1992-
08.com.netapp:sn.48391d66c0a611e00000000000000000:subsystem.subsys_CLIEN
T116
traddr: 192.168.1.116 eflags: not specified sectype: none

```

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>
```

◦ 출력 예: *

```

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

```

3. 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 -l
<ctrl_loss_timeout_in_seconds>
```

◦ 출력 예: *

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

+



NetApp은 를 설정할 것을 권장합니다 `ctrl-loss-tmo` 옵션을 로 설정합니다 -1 따라서 경로 손실이 발생하면 NVMe/TCP 이니시에이터가 무한정 다시 연결을 시도합니다.

NVMe-oF를 검증합니다

다음 절차를 사용하여 NVMe-oF를 검증할 수 있습니다.

단계

1. 커널 내 NVMe 다중 경로가 활성화되었는지 확인:

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

2. 호스트에 ONTAP NVMe 네임스페이스에 대한 올바른 컨트롤러 모델이 있는지 확인합니다.

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

◦ 출력 예: *

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

3. 해당 ONTAP NVMe I/O 컨트롤러에 대한 NVMe I/O 정책을 확인합니다.

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

◦ 출력 예: *

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

4. ONTAP 네임스페이스가 호스트에 표시되는지 확인합니다.

```
nvme list -v
```

◦ 출력 예: *

```
Subsystem          Subsystem-NQN
Controllers
-----
-----
nvme-subsys0      nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.unidir_dhcha
p      nvme0, nvme1, nvme2, nvme3

Device   SN                      MN
FR       TxPort Address          Subsystem      Namespaces
-----
-----
nvme0    81LGgBUqsI3EAAAAAAAAE NetApp ONTAP Controller  FFFFFFFF tcp
traddr=192.168.2.214,trsvcid=4420,host_traddr=192.168.2.14 nvme-subsys0
nvme0n1
nvme1    81LGgBUqsI3EAAAAAAAAE NetApp ONTAP Controller  FFFFFFFF tcp
traddr=192.168.2.215,trsvcid=4420,host_traddr=192.168.2.14 nvme-subsys0
nvme0n1
nvme2    81LGgBUqsI3EAAAAAAAAE NetApp ONTAP Controller  FFFFFFFF tcp
traddr=192.168.1.214,trsvcid=4420,host_traddr=192.168.1.14 nvme-subsys0
nvme0n1
nvme3    81LGgBUqsI3EAAAAAAAAE NetApp ONTAP Controller  FFFFFFFF tcp
traddr=192.168.1.215,trsvcid=4420,host_traddr=192.168.1.14 nvme-subsys0
nvme0n1

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

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

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

NVMe/FC

```
# nvme list-subsys /dev/nvme1n1
nvme-subsys1 - NQN=nqn.1992-
08.com.netapp:sn.04ba0732530911ea8e8300a098dfdd91:subsystem.nvme_145
_1
\
+- nvme2 fc traddr=nn-0x208100a098dfdd91:pn-
0x208200a098dfdd91,host_traddr=nn-0x200000109b579d5f:pn-
0x100000109b579d5f live optimized
+- nvme3 fc traddr=nn-0x208100a098dfdd91:pn-
0x208500a098dfdd91,host_traddr=nn-0x200000109b579d5e:pn-
0x100000109b579d5e live optimized
+- nvme4 fc traddr=nn-0x208100a098dfdd91:pn-
0x208400a098dfdd91,host_traddr=nn-0x200000109b579d5e:pn-
0x100000109b579d5e live non-optimized
+- nvme6 fc traddr=nn-0x208100a098dfdd91:pn-
0x208300a098dfdd91,host_traddr=nn-0x200000109b579d5f:pn-
0x100000109b579d5f live non-optimized
```

NVMe/TCP

```
# nvme list-subsys
nvme-subsys0 - NQN=nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.unidir_d
hchap
hostnqn=nqn.2014-08.org.nvmexpress:uuid:e58eca24-faff-11ea-8fee-
3a68dd3b5c5f
iopolicy=round-robin

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

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

명령

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

◦ 출력 예 *:

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

JSON을 참조하십시오

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

◦ 출력 예 *:

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

영구 검색 컨트롤러를 만듭니다

ONTAP 9.11.1부터 SUSE Linux Enterprise Server 15 SP4 호스트에 대한 영구 검색 컨트롤러(PDC)를 만들 수 있습니다. PDC는 NVMe 하위 시스템의 추가 또는 제거 시나리오와 검색 로그 페이지 데이터의 변경 사항을 자동으로 감지하기 위해 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>
```


출력 예 표시:

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

====Discovery Log Entry 7=====

trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 0
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.unidir_n
one
traddr: 192.168.2.214
eflags: none
sectype: none

====Discovery Log Entry 8=====

trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 0
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.subsys_C
LIENT114
traddr: 192.168.1.214
eflags: none
sectype: none

====Discovery Log Entry 9=====

trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 0
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.0501daf15dda11eeab68d039eaa7a232:subsystem.subsys_C
LIENT114
traddr: 192.168.1.215
eflags: none
sectype: none

====Discovery Log Entry 10=====

trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 0
trsvcid: 4420

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

```

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

```

2. 검색 하위 시스템에 대한 PDC 생성:

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

◦ 출력 예: *

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

3. ONTAP 컨트롤러에서 PDC가 생성되었는지 확인합니다.

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

◦ 출력 예: *

```
vserver nvme show-discovery-controller -instance -vserver vs_nvme175
Vserver Name: vs_CLIENT116 Controller ID: 00C0h
Discovery Subsystem NQN: nqn.1992-08.com.netapp:sn.48391d66c0a611ecaaa5d039ea165514:discovery Logical
Interface UUID: d23cbb0a-c0a6-11ec-9731-d039ea165abc Logical Interface:
CLIENT116_lif_4a_1
Node: A400-14-124
Host NQN: nqn.2014-08.org.nvmexpress:uuid:12372496-59c4-4d1b-be09-
74362c0c1afc
Transport Protocol: nvme-tcp
Initiator Transport Address: 192.168.1.16
Host Identifier: 59de25be738348f08a79df4bce9573f3 Admin Queue Depth: 32
Header Digest Enabled: false Data Digest Enabled: false
Vserver UUID: 48391d66-c0a6-11ec-aaa5-d039ea165514
```

보안 대역내 인증을 설정합니다

ONTAP 9.12.1부터 SUSE Linux Enterprise Server 15 SP4 호스트와 ONTAP 컨트롤러 간에 NVMe/TCP 및 NVMe/FC를 통해 보안 인밴드 인증이 지원됩니다.

보안 인증을 설정하려면 각 호스트 또는 컨트롤러가 에 연결되어 있어야 합니다 DH-HMAC-CHAP 키 - NVMe 호스트 또는 컨트롤러의 NQN과 관리자가 구성한 인증 비밀의 조합입니다. 피어를 인증하려면 NVMe 호스트 또는 컨트롤러가 피어와 연결된 키를 인식해야 합니다.

CLI 또는 구성 JSON 파일을 사용하여 보안 대역 내 인증을 설정할 수 있습니다. 서로 다른 하위 시스템에 대해 다른 dhchap 키를 지정해야 하는 경우 구성 JSON 파일을 사용해야 합니다.

CLI를 참조하십시오

단계

1. 호스트 NQN 가져오기:

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

2. SUSE Linux Enterprise Server 15 SP4 호스트에 대한 dhchap 키를 생성합니다.

```
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

를 누릅니다

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:d3ca725a-  
ac8d-4d88-b46a-174ac235139b  
DHHC-  
1:03:J2UJQfj9f0pLnpF/ASDJRTyILKJr5CougGpGdQSysPrLu6RW1fG15VSjbeDF1n1DE  
h3nVBe19nQ/LxreSBeH/bx/pU=:
```

1. 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}
```

2. 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>
```

3. 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 설정의 예제 출력: *

```
SR650-14-114:~ # cat /sys/class/nvme-subsystem/nvme-
subsys1/nvme*/dhchap_secret
DHHC-
1:03:je1nQCmjJLUKD62mpYbz1puw00Iws86NB96uNO/t3jbvhp7fjyR9bIRjOHg8
wQtye1JCFSMkBQH3pTKGdYR1OV9gx00=:
DHHC-
1:03:je1nQCmjJLUKD62mpYbz1puw00Iws86NB96uNO/t3jbvhp7fjyR9bIRjOHg8
wQtye1JCFSMkBQH3pTKGdYR1OV9gx00=:
DHHC-
1:03:je1nQCmjJLUKD62mpYbz1puw00Iws86NB96uNO/t3jbvhp7fjyR9bIRjOHg8
wQtye1JCFSMkBQH3pTKGdYR1OV9gx00=:
DHHC-
1:03:je1nQCmjJLUKD62mpYbz1puw00Iws86NB96uNO/t3jbvhp7fjyR9bIRjOHg8
wQtye1JCFSMkBQH3pTKGdYR1OV9gx00=:
```

b. 컨트롤러 dhchap 키를 확인합니다.

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

▪ 양방향 구성의 출력 예: *


```

SR650-14-114:~ # cat /sys/class/nvme-subsystem/nvme-
subsys6/nvme*/dhchap_ctrl_secret
DHHC-
1:03:WorVEV83eY053kV4Iel50pphbX5LAph03F8fgH3913t1rkSGDBJTt3crXeTUB8f
CwGbPsEyz6CXxdQJi6kbn4IzmkFU=:
DHHC-
1:03:WorVEV83eY053kV4Iel50pphbX5LAph03F8fgH3913t1rkSGDBJTt3crXeTUB8f
CwGbPsEyz6CXxdQJi6kbn4IzmkFU=:
DHHC-
1:03:WorVEV83eY053kV4Iel50pphbX5LAph03F8fgH3913t1rkSGDBJTt3crXeTUB8f
CwGbPsEyz6CXxdQJi6kbn4IzmkFU=:
DHHC-
1:03:WorVEV83eY053kV4Iel50pphbX5LAph03F8fgH3913t1rkSGDBJTt3crXeTUB8f
CwGbPsEyz6CXxdQJi6kbn4IzmkFU=:

```

JSON 파일

를 사용할 수 있습니다 /etc/nvme/config.json 파일이 있는 파일 nvme connect-all ONTAP 컨트롤러 구성에서 여러 NVMe 서브시스템을 사용할 수 있는 경우에 명령을 사용합니다.

을 사용하여 JSON 파일을 생성할 수 있습니다 -o 옵션을 선택합니다. 자세한 구문 옵션은 NVMe connect-all man 페이지를 참조하십시오.

단계

1. JSON 파일 구성:

```

# cat /etc/nvme/config.json
[
  {
    "hostnqn": "nqn.2014-08.org.nvmexpress:uuid:12372496-59c4-4d1b-
be09-74362c0c1afc",
    "hostid": "3ae10b42-21af-48ce-a40b-cfb5bad81839",
    "dhchap_key": "DHHC-
1:03:Cu3ZzfIz1WmlqZFncMqpAgn/T6EVOcIFHez215U+Pow8jTgBF2UbNk3DK4wfk2E
ptWpna1rpwG5CndpOgxpRxx9m41w=: "
  },
  {
    "hostnqn": "nqn.2014-08.org.nvmexpress:uuid:12372496-59c4-4d1b-
be09-74362c0c1afc",
    "subsystems": [
      {
        "nqn": "nqn.1992-
08.com.netapp:sn.48391d66c0a611ecaaa5d039ea165514:subsystem.subsys_C
LIENT116",
        "ports": [

```

```

    {
        "transport": "tcp",
        "traddr": "192.168.1.117",
        "host_traddr": "192.168.1.16",
        "trsvcid": "4420",
        "dhchap_ctrl_key": "DHHC-
1:01:0h58bcT/uu0rCpGsDYU6ZHZvRuVqsYKuBRS0Nu0VPx5HEwaZ:"
    },
    {
        "transport": "tcp",
        "traddr": "192.168.1.116",
        "host_traddr": "192.168.1.16",
        "trsvcid": "4420",
        "dhchap_ctrl_key": "DHHC-
1:01:0h58bcT/uu0rCpGsDYU6ZHZvRuVqsYKuBRS0Nu0VPx5HEwaZ:"
    },
    {
        "transport": "tcp",
        "traddr": "192.168.2.117",
        "host_traddr": "192.168.2.16",
        "trsvcid": "4420",
        "dhchap_ctrl_key": "DHHC-
1:01:0h58bcT/uu0rCpGsDYU6ZHZvRuVqsYKuBRS0Nu0VPx5HEwaZ:"
    },
    {
        "transport": "tcp",
        "traddr": "192.168.2.116",
        "host_traddr": "192.168.2.16",
        "trsvcid": "4420",
        "dhchap_ctrl_key": "DHHC-
1:01:0h58bcT/uu0rCpGsDYU6ZHZvRuVqsYKuBRS0Nu0VPx5HEwaZ:"
    }
]
}
]

```

[NOTE]

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

2. config JSON 파일을 사용하여 ONTAP 컨트롤러에 연결합니다.

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

◦ 출력 예 *:

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

3. 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
```

▪ 출력 예: *

```
DHHC-1:01:NunEWY7AZlXqxITGheByarwZdQvU4ebZg9H0jIr6nOHEkxJg:
```

b. 컨트롤러 dhchap 키를 확인합니다.

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

▪ 출력 예: *

```
DHHC-
1:03:2YJinsxa2v3+m8qqCiTnmGBZoH6mIT6G/6f0aGO8viVZB4VLNLH4z8CvK7pVYxN
6S5fOAtaU3DNI12rieRMfdbg3704=:
```

알려진 문제

SUSE Linux Enterprise Server 15 SP4(ONTAP 릴리스 포함)에는 알려진 문제가 없습니다.

ONTAP가 포함된 SUSE Linux Enterprise Server 15 SP3의 NVMe-oF 호스트 구성

ANA(비대칭 네임스페이스 액세스)가 있는 SUSE Linux Enterprise Server 15 SP3에 대해 NVMe over Fabrics 또는 NVMe-oF(NVMe/FC 및 기타 전송 포함)가 지원됩니다. ANA는 NVMe-oF 환경에서 ALUA에 상응하는 제품으로 현재 커널 내 NVMe 다중 경로를 통해 구현됩니다. 이 절차를 사용하면 SUSE Linux Enterprise Server 15 SP3의 ANA 및 ONTAP를 대상으로 사용하여 커널 내 NVMe 다중 경로와 함께 NVMe-oF를 활성화할 수 있습니다.

을 참조하십시오 ["NetApp 상호 운용성 매트릭스"](#) 지원되는 구성에 대한 정확한 세부 정보

피처

- SUSE Linux Enterprise Server 15 SP3은 NVMe/FC 및 기타 전송을 지원합니다.
- NVMe-oF에는 sandlun 지원이 없습니다. 따라서 SUSE Linux Enterprise Server 15 SP3의 NVMe-OF에 대한 LUHU 지원은 없습니다. NVMe-oF용 기본 NVMe-CLI 패키지에 포함된 NetApp 플러그인을 사용할 수 있습니다. 모든 NVMe-oF 전송을 지원해야 합니다.
- NVMe와 SCSI 트래픽을 모두 동일한 동일한 호스트에서 실행할 수 있습니다. 실제로 이는 일반적으로 고객에게 배포된 호스트 구성일 것으로 예상됩니다. 따라서 SCSI의 경우 SCSI LUN에 대해 평소처럼 dm-multipath를 구성하여 mpath 디바이스를 생성할 수 있지만 NVMe multipath를 사용하여 호스트에 NVMe-of multipath 장치를 구성할 수 있습니다.

알려진 제한 사항

현재 NVMe-oF 프로토콜을 사용한 SAN 부팅은 지원되지 않습니다.

In-kernel NVMe Multipath를 활성화합니다

커널 내 NVMe 다중 경로는 SUSE Linux Enterprise Server 15 SP3과 같은 SUSE Linux Enterprise Server 호스트에서 이미 기본적으로 활성화되어 있습니다. 따라서 여기서는 추가 설정이 필요하지 않습니다. 지원되는 구성에 대한 정확한 세부 정보는 ["NetApp 상호 운용성 매트릭스"](#)참조하십시오.

NVMe - 이니시에이터 패키지

을 참조하십시오 ["NetApp 상호 운용성 매트릭스"](#) 지원되는 구성에 대한 정확한 세부 정보

1. SUSE Linux Enterprise Server 15 SP3 MU 호스트에 필수 커널 및 NVMe-CLI MU 패키지가 설치되어 있는지 확인합니다.

예:

```
# uname -r
5.3.18-59.5-default

# rpm -qa|grep nvme-cli
nvme-cli-1.13-3.3.1.x86_64
```

이제 위의 NVMe-CLI MU 패키지에 다음이 포함됩니다.

- * NVMe/FC 자동 연결 스크립트 * - 네임스페이스에 대한 기본 경로가 복구되거나 호스트를 재부팅하는 동안 NVMe/FC 자동 연결에 필요함:

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

- * ONTAP udev 규칙 * - NVMe 다중 경로 라운드 로빈 로드 밸런서가 기본적으로 모든 ONTAP 네임스페이스에 적용되도록 보장하는 새로운 udev 규칙:

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

- * ONTAP 장치용 NetApp 플러그인 * - 기존 NetApp 플러그인은 ONTAP 네임스페이스도 처리하도록 수정되었습니다.

2. 호스트의 '/etc/NVMe/hostnqn'에서 hostnqn 문자열을 확인하고 ONTAP 배열의 해당 하위 시스템에 대한 hostnqn 문자열과 올바르게 일치하는지 확인합니다. 예를 들면, 다음과 같습니다.

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

호스트에서 사용 중인 FC 어댑터에 따라 다음 단계를 수행하십시오.

NVMe/FC 구성

Broadcom/Emulex

1. 권장 어댑터 및 펌웨어 버전이 있는지 확인합니다. 예를 들면, 다음과 같습니다.

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

- 최신 lpfc 드라이버(수신함 및 아웃박스 모두)에는 lpfc_enable_fc4_type 기본값이 3으로 설정되어 있으므로 더 이상 '/etc/modprobe.d/lpfc.conf'에서 명시적으로 설정하고 'initrd'를 다시 생성할 필요가 없습니다. lpfc NVMe 지원은 기본적으로 이미 활성화되어 있습니다.

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

- 기존의 기본 받은 편지함 lpfc 드라이버는 이미 최신 버전이며 NVMe/FC와 호환됩니다. 따라서 lpfc OOB 드라이버를 설치할 필요가 없습니다.

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

2. 이니시에이터 포트가 실행 중인지 확인합니다.

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

3. NVMe/FC 이니시에이터 포트가 활성화되고 타겟 포트를 볼 수 있으며 모든 포트가 실행 중인지 확인합니다. + 다음 예에서는 이니시에이터 포트 하나만 사용하도록 설정되고 두 개의 타겟 LIF로 연결됩니다.

```

# cat /sys/class/scsi_host/host*/nvme_info
NVME Initiator Enabled
XRI Dist lpfc0 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc0 WWPN x100000109b579d5e WWNN x200000109b579d5e DID x011c00 ONLINE
NVME RPORT WWPN x208400a098dfdd91 WWNN x208100a098dfdd91 DID x011503 TARGET DISCSRVC ONLINE
NVME RPORT WWPN x208500a098dfdd91 WWNN x208100a098dfdd91 DID x010003 TARGET DISCSRVC ONLINE
NVME Statistics
LS: Xmt 0000000e49 Cmpl 0000000e49 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000003ceb594f Issue 000000003ce65dbe OutIO
ffffffffffffb046f
abort 00000bd2 noxri 00000000 nondlp 00000000 qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 000014f4 Err 00012abd
NVME Initiator Enabled
XRI Dist lpfc1 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc1 WWPN x100000109b579d5f WWNN x200000109b579d5f DID x011b00 ONLINE
NVME RPORT WWPN x208300a098dfdd91 WWNN x208100a098dfdd91 DID x010c03 TARGET DISCSRVC ONLINE
NVME RPORT WWPN x208200a098dfdd91 WWNN x208100a098dfdd91 DID x012a03 TARGET DISCSRVC ONLINE
NVME Statistics
LS: Xmt 0000000e50 Cmpl 0000000e50 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000003c9859ca Issue 000000003c93515e OutIO
ffffffffffffaf794
abort 00000b73 noxri 00000000 nondlp 00000000 qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 0000159d Err 000135c3

```

4. 호스트를 재부팅합니다.

1MB I/O 크기 활성화(옵션)

ONTAP는 컨트롤러 식별 데이터에 8의 MDTS(MAX Data 전송 크기)를 보고합니다. 즉, 최대 I/O 요청 크기는 최대 1MB여야 합니다. 그러나 Broadcom NVMe/FC 호스트에 대해 크기 1MB의 입출력 요청을 발급하려면 lpfc 매개 변수 "lpfc_sg_seg_cnt"를 기본값 64에서 최대 256까지 범프해야 합니다. 다음 지침에 따라 수행합니다.

1. 해당 'modprobe lpfc.conf' 파일에 256을 추가합니다.


```
# cat /etc/modprobe.d/lpfc.conf
options lpfc lpfc_sg_seg_cnt=256
```

2. `dracut -f` 명령을 실행하고 호스트를 재부팅합니다.
3. 재부팅 후 해당 sysfs 값을 확인하여 위의 설정이 적용되었는지 확인합니다.

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

이제 Broadcom NVMe/FC 호스트는 ONTAP 네임스페이스 장치에서 1MB I/O 요청을 보낼 수 있습니다.

Marvell/QLogic

최신 SUSE Linux Enterprise Server 15 SP3 MU 커널에 포함된 기본 받은 편지함 qla2xxx 드라이버에는 최신 업스트림 수정 사항이 있습니다. 이러한 수정 사항은 ONTAP 지원에 필수적입니다.

1. 지원되는 어댑터 드라이버 및 펌웨어 버전을 실행하고 있는지 확인합니다(예:

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

2. Marvell 어댑터가 NVMe/FC Initiator로 작동하도록 하는 "ql2xnvmeenable"이 설정되어 있는지 확인합니다.

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

NVMe/TCP를 구성합니다

NVMe/FC와 달리 NVMe/TCP에는 자동 연결 기능이 없습니다. Linux NVMe/TCP 호스트에는 다음과 같은 두 가지 주요 제한 사항이 있습니다.

- * 경로 복구 후 자동 재연결 불가 * NVMe/TCP는 경로 다운 후 10분 동안 기본 'Ctrl-Loss-TMO' 타이머 이후에 복구된 경로에 자동으로 다시 연결할 수 없습니다.
- * 호스트 부팅 중 자동 연결 없음 * 호스트 부팅 중에 NVMe/TCP도 자동으로 연결할 수 없습니다.

시간 초과를 방지하려면 페일오버 이벤트에 대한 재시도 기간을 최소 30분으로 설정해야 합니다. Ctrl_Loss_TMO 타이머 값을 증가시켜 재시도 기간을 늘릴 수 있습니다. 다음은 세부 정보입니다.

단계

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

```

# nvme discover -t tcp -w 192.168.1.8 -a 192.168.1.51
Discovery Log Number of Records 10, Generation counter 119
=====Discovery Log Entry 0=====
trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 0
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.56e362e9bb4f11ebbade039ea165abc:subsystem.nvme_118_tcp
_1
traddr: 192.168.2.56
sectype: none
=====Discovery Log Entry 1=====
trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 1
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.56e362e9bb4f11ebbade039ea165abc:subsystem.nvme_118_tcp
_1
traddr: 192.168.1.51
sectype: none
=====Discovery Log Entry 2=====
trtype: tcp
adrfam: ipv4
subtype: nvme subsystem
treq: not specified
portid: 0
trsvcid: 4420
subnqn: nqn.1992-
08.com.netapp:sn.56e362e9bb4f11ebbade039ea165abc:subsystem.nvme_118_tcp
_2
traddr: 192.168.2.56
sectype: none
...

```

2. 다른 NVMe/TCP 이니시에이터-타겟 LIF combos가 검색 로그 페이지 데이터를 성공적으로 가져올 수 있는지 확인합니다. 예를 들면, 다음과 같습니다.

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

3. 실행 `nvme connect-all` 노드를 통해 지원되는 모든 NVMe/TCP 이니시에이터-타겟 LIF에 대해 명령을 실행합니다. 를 더 길게 설정하십시오 `ctrl_loss_tmo` 타이머 재시도 기간(예: 에서 설정할 수 있는 30분 -1 1800) 연결 중 - 경로 손실이 발생할 경우 더 오랜 시간 동안 다시 시도하도록 합니다. 예를 들면, 다음과 같습니다.

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

NVMe-oF를 검증합니다

1. 다음을 확인하여 In-kernel NVMe multipath가 실제로 활성화되어 있는지 확인합니다.

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

2. 각 ONTAP 네임스페이스에 대한 적절한 NVMe-oF 설정(예: "NetApp ONTAP Controller"로 설정된 모델 및 "라운드 로빈"으로 설정된 로드 밸런싱 지정값이 호스트에 올바르게 반영되는지 확인하십시오.

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

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

3. ONTAP 네임스페이스가 호스트에 제대로 반영되는지 확인합니다. 예를 들면, 다음과 같습니다.

```
# nvme list
```

Node	SN	Model	Namespace
/dev/nvme0n1	81CZ5BQuUNfGAAAAAAB	NetApp ONTAP Controller	1
Usage	Format	FW Rev	
85.90 GB / 85.90 GB	4 KiB + 0 B	FFFFFFFF	

다른 예:

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

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

4. 각 경로의 컨트롤러 상태가 라이브이고 적절한 ANA 상태인지 확인합니다. 예를 들면, 다음과 같습니다.

```
# nvme list-subsys /dev/nvme1n1
nvme-subsys1 - NQN=nqn.1992-
08.com.netapp:sn.04ba0732530911ea8e8300a098dfdd91:subsystem.nvme_145_1
\
+- nvme2 fc traddr=nn-0x208100a098dfdd91:pn-0x208200a098dfdd91
host_traddr=nn-0x200000109b579d5f:pn-0x100000109b579d5f live non-
optimized
+- nvme3 fc traddr=nn-0x208100a098dfdd91:pn-0x208500a098dfdd91
host_traddr=nn-0x200000109b579d5e:pn-0x100000109b579d5e live non-
optimized
+- nvme4 fc traddr=nn-0x208100a098dfdd91:pn-0x208400a098dfdd91
host_traddr=nn-0x200000109b579d5e:pn-0x100000109b579d5e live optimized
+- nvme6 fc traddr=nn-0x208100a098dfdd91:pn-0x208300a098dfdd91
host_traddr=nn-0x200000109b579d5f:pn-0x100000109b579d5f live optimized
```

다른 예:

```

#nvme list-subsys /dev/nvme0n1
nvme-subsys0 - NQN=nqn.1992-
08.com.netapp:sn.37ba7d9cbfba11eba35dd039ea165514:subsystem.nvme_114_tcp
_1
\
+- nvme0 tcp traddr=192.168.2.36 trsvcid=4420 host_traddr=192.168.1.4
live optimized
+- nvme1 tcp traddr=192.168.1.31 trsvcid=4420 host_traddr=192.168.1.4
live optimized
+- nvme10 tcp traddr=192.168.2.37 trsvcid=4420 host_traddr=192.168.1.4
live non-optimized
+- nvme11 tcp traddr=192.168.1.32 trsvcid=4420 host_traddr=192.168.1.4
live non-optimized
+- nvme20 tcp traddr=192.168.2.36 trsvcid=4420 host_traddr=192.168.2.5
live optimized
+- nvme21 tcp traddr=192.168.1.31 trsvcid=4420 host_traddr=192.168.2.5
live optimized
+- nvme30 tcp traddr=192.168.2.37 trsvcid=4420 host_traddr=192.168.2.5
live non-optimized
+- nvme31 tcp traddr=192.168.1.32 trsvcid=4420 host_traddr=192.168.2.5
live non-optimized

```

5. NetApp 플러그인에 각 ONTAP 네임스페이스 장치에 대한 올바른 값이 표시되는지 확인합니다. 예를 들면, 다음과 같습니다.

```

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

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

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

```

다른 예:

```

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

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

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

```

알려진 문제

알려진 문제가 없습니다.

ONTAP가 포함된 SUSE Linux Enterprise Server 15 SP2용 NVMe/FC 호스트 구성

NVMe/FC는 SUSE Linux Enterprise Server 15 SP2를 사용하는 ONTAP 9.6 이상에서 지원됩니다. SUSE Linux Enterprise Server 15 SP2 호스트는 동일한 파이버 채널 이니시에이터 어댑터 포트를 통해 NVMe/FC 및 FCP 트래픽을 모두 실행할 수 있습니다. 지원되는 FC 어댑터 및 컨트롤러 목록은 ["Hardware Universe"](#) 참조하십시오.

지원되는 구성 및 버전의 최신 목록은 ["NetApp 상호 운용성 매트릭스"](#)를 참조하십시오.



이 문서에 제공된 구성 설정을 사용하여 에 연결된 클라우드 클라이언트를 구성할 수 있습니다 ["Cloud Volumes ONTAP"](#) 및 ["ONTAP용 Amazon FSx"](#).

알려진 제한 사항

현재 NVMe-oF 프로토콜을 사용한 SAN 부팅은 지원되지 않습니다.

SUSE Linux Enterprise Server 15 SP2에서 NVMe/FC를 활성화합니다

1. 권장 SUSE Linux Enterprise Server 15 SP2 MU 커널 버전으로 업그레이드하십시오.
2. 기본 NVMe-CLI 패키지를 업그레이드합니다.

이 기본 NVMe-CLI 패키지에는 NVMe/FC 자동 연결 스크립트, NVMe 다중 경로에 대한 라운드 로빈 로드 밸런싱을 지원하는 ONTAP udev 규칙 및 ONTAP 네임스페이스를 위한 NetApp 플러그인이 포함되어 있습니다.

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

3. SUSE Linux Enterprise Server 15 SP2 호스트에서 호스트 NQN 문자열을 확인하고 ONTAP 스토리지의 해당 하위 시스템에 대한 호스트 NQN /etc/nvme/hostnqn 문자열과 일치하는지 확인합니다. 예를 들면 다음과 같습니다.

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

```
::> vsserver nvme subsystem host show -vsserver vs_fc_nvme_145
Vserver Subsystem Host NQN
-----
vs_fc_nvme_145
nvme_145_1
nqn.2014-08.org.nvmexpress:uuid:c7b07b16-a22e-41a6-a1fd-cf8262c8713f
nvme_145_2
nqn.2014-08.org.nvmexpress:uuid:c7b07b16-a22e-41a6-a1fd-cf8262c8713f
nvme_145_3
nqn.2014-08.org.nvmexpress:uuid:c7b07b16-a22e-41a6-a1fd-cf8262c8713f
nvme_145_4
nqn.2014-08.org.nvmexpress:uuid:c7b07b16-a22e-41a6-a1fd-cf8262c8713f
nvme_145_5
nqn.2014-08.org.nvmexpress:uuid:c7b07b16-a22e-41a6-a1fd-cf8262c8713f
5 entries were displayed.
```

4. 호스트를 재부팅합니다.

NVMe/FC용 Broadcom FC 어댑터를 구성합니다

1. 지원되는 어댑터를 사용하고 있는지 확인합니다. 지원되는 어댑터의 최신 목록은 ["NetApp 상호 운용성 매트릭스"](#)를 참조하십시오.


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

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

2. 권장되는 Broadcom lpfc 펌웨어 및 기본 받은 편지함 드라이버 버전을 사용하고 있는지 확인합니다.

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

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

3. lpfc_enable_fc4_type이 3으로 설정되어 있는지 확인합니다.

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

4. 이니시에이터 포트가 실행 중인지 확인합니다.

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

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

5. NVMe/FC 이니시에이터 포트가 활성화되어 실행 중이며 타겟 LIF를 볼 수 있는지 확인하십시오.

```

# cat /sys/class/scsi_host/host*/nvme_info
NVME Initiator Enabled
XRI Dist lpfc0 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc0 WWPN x100000109b579d5e WWNN x200000109b579d5e DID
x011c00 ONLINE
NVME RPORT WWPN x208400a098dfdd91 WWNN x208100a098dfdd91 DID x011503
TARGET DISCSRVC ONLINE
NVME RPORT WWPN x208500a098dfdd91 WWNN x208100a098dfdd91 DID x010003
TARGET DISCSRVC ONLINE
NVME Statistics
LS: Xmt 0000000e49 Cmpl 0000000e49 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000003ceb594f Issue 000000003ce65dbe OutIO
ffffffffffffb046f
abort 00000bd2 noxri 00000000 nondlp 00000000 qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 000014f4 Err 00012abd
NVME Initiator Enabled
XRI Dist lpfc1 Total 6144 IO 5894 ELS 250
NVME LPORT lpfc1 WWPN x100000109b579d5f WWNN x200000109b579d5f DID
x011b00 ONLINE
NVME RPORT WWPN x208300a098dfdd91 WWNN x208100a098dfdd91 DID x010c03
TARGET DISCSRVC ONLINE
NVME RPORT WWPN x208200a098dfdd91 WWNN x208100a098dfdd91 DID x012a03
TARGET DISCSRVC ONLINE
NVME Statistics
LS: Xmt 0000000e50 Cmpl 0000000e50 Abort 00000000
LS XMIT: Err 00000000 CMPL: xb 00000000 Err 00000000
Total FCP Cmpl 000000003c9859ca Issue 000000003c93515e OutIO
ffffffffffffaf794
abort 00000b73 noxri 00000000 nondlp 00000000 qdepth 00000000 wqerr
00000000 err 00000000
FCP CMPL: xb 0000159d Err 000135c3

```

NVMe/FC를 검증합니다

1. 다음 NVMe/FC 설정을 확인하십시오.

```

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

```

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

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

2. 네임스페이스가 만들어졌는지 확인합니다.

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

3. ANA 경로 상태를 확인한다.

```
# nvme list-subsys /dev/nvme1n1
nvme-subsys1 - NQN=nqn.1992-
08.com.netapp:sn.04ba0732530911ea8e8300a098dfdd91:subsystem.nvme_145_1
\
+- nvme2 fc traddr=nn-0x208100a098dfdd91:pn-0x208200a098dfdd91
host_traddr=nn-0x200000109b579d5f:pn-0x100000109b579d5f live
inaccessible
+- nvme3 fc traddr=nn-0x208100a098dfdd91:pn-0x208500a098dfdd91
host_traddr=nn-0x200000109b579d5e:pn-0x100000109b579d5e live
inaccessible
+- nvme4 fc traddr=nn-0x208100a098dfdd91:pn-0x208400a098dfdd91
host_traddr=nn-0x200000109b579d5e:pn-0x100000109b579d5e live optimized
+- nvme6 fc traddr=nn-0x208100a098dfdd91:pn-0x208300a098dfdd91
host_traddr=nn-0x200000109b579d5f:pn-0x100000109b579d5f live optimized
```

4. ONTAP 장치용 NetApp 플러그인을 확인합니다.

```

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

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

```

알려진 문제

알려진 문제가 없습니다.

Broadcom NVMe/FC에 대해 1MB I/O 크기를 활성화합니다

ONTAP는 컨트롤러 식별 데이터에서 MDTS(MAX Data 전송 크기)를 8로 보고합니다. 이는 최대 I/O 요청 크기가 1MB까지 될 수 있음을 의미합니다. Broadcom NVMe/FC 호스트에 대해 1MB 크기의 I/O 요청을 발행하려면 매개 변수 값을 `lpfc_sg_seg_cnt` 기본값인 64에서 256으로 늘려야 `lpfc` 합니다.



다음 단계는 Qlogic NVMe/FC 호스트에는 적용되지 않습니다.

단계

1. `lpfc_sg_seg_cnt` 매개변수를 256으로 설정합니다.

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

```
options lpfc lpfc_sg_seg_cnt=256
```

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

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

예상 값은 256입니다.

lpfc Verbose 로깅

NVMe/FC용 lpfc 드라이버를 설정합니다.

단계

1. 를 설정합니다 `lpfc_log_verbose` 다음 값 중 하나에 대한 드라이버 설정을 사용하여 NVMe/FC 이벤트를 기록합니다.

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

2. 값을 설정한 후 를 실행합니다 `dracut-f` 명령을 실행하여 호스트를 재부팅합니다.
3. 설정을 확인합니다.

```
# cat /etc/modprobe.d/lpfc.conf options lpfc lpfc_log_verbose=0xf00083

# cat /sys/module/lpfc/parameters/lpfc_log_verbose 15728771
```

SUSE Linux Enterprise Server 15 SP1 및 ONTAP을 위한 NVMe/FC 호스트 구성

SUSE Linux Enterprise Server 15 SP1 및 ONTAP를 타겟으로 실행하는 호스트에서 NVMe over Fibre Channel(NVMe/FC)을 구성할 수 있습니다.

NVMe/FC는 다음 SUSE Linux Enterprise Server 버전에 대해 ONTAP 9.6 이상에서 지원됩니다.

- SUSE Linux Enterprise Server 15 SP1

SUSE Linux Enterprise Server 15 SP1 호스트는 동일한 파이버 채널 이니시에이터 어댑터 포트를 통해 NVMe/FC 및 FCP 트래픽을 모두 실행할 수 있습니다. 지원되는 FC 어댑터 및 컨트롤러 목록은 를 "[Hardware Universe](#)" 참조하십시오.

지원되는 구성 및 버전의 최신 목록은 를 참조하십시오 "[NetApp 상호 운용성 매트릭스](#)".

- 기본 NVMe/FC 자동 연결 스크립트가 NVMe-CLI 패키지에 포함되어 있습니다. SUSE Linux Enterprise Server 15 SP1에서 기본 받은 편지함 lpfc 드라이버를 사용할 수 있습니다.

알려진 제한 사항

현재 NVMe-oF 프로토콜을 사용한 SAN 부팅은 지원되지 않습니다.

SUSE Linux Enterprise Server 15 SP1에서 NVMe/FC를 활성화합니다

1. 권장 SUSE Linux Enterprise Server 15 SP2 MU 커널로 업그레이드하십시오
2. 권장되는 NVMe-CLI MU 버전으로 업그레이드하십시오.

이 NVMe-CLI 패키지에는 기본 NVMe/FC 자동 연결 스크립트가 포함되어 있으므로 SUSE Linux Enterprise Server 15 SP1 호스트에 Broadcom에서 제공하는 외부 NVMe/FC 자동 연결 스크립트를 설치할 필요가 없습니다. 또한 이 패키지에는 NVMe 다중 경로에 라운드 로빈 로드 밸런싱을 지원하는 ONTAP udev 규칙 및 ONTAP 장치용 NetApp 플러그인이 포함되어 있습니다.

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

3. SUSE Linux Enterprise Server 15 SP1 호스트에서 에서 호스트 NQN 문자열을 확인하고 ONTAP 스토리지의 해당 하위 시스템에 대한 호스트 NQN /etc/nvme/hostnqn 문자열과 일치하는지 확인합니다. 예를 들면 다음과 같습니다.

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

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

4. 호스트를 재부팅합니다.

NVMe/FC용 Broadcom FC 어댑터를 구성합니다

1. 지원되는 어댑터를 사용하고 있는지 확인합니다. 지원되는 어댑터의 최신 목록은 를 참조하십시오 "[NetApp 상호 운용성 매트릭스](#)".

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

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

2. 권장되는 Broadcom lpfc 펌웨어 및 기본 받은 편지함 드라이버 버전을 사용하고 있는지 확인합니다.

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

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

3. lpfc_enable_fc4_type이 3으로 설정되어 있는지 확인합니다.

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

4. 이니시에이터 포트가 실행 중인지 확인합니다.

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

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

5. NVMe/FC 이니시에이터 포트가 활성화되어 실행 중이며 타겟 LIF를 볼 수 있는지 확인하십시오.

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

NVMe/FC를 검증합니다

1. 다음 NVMe/FC 설정을 확인하십시오.

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

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

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

2. 네임스페이스가 만들어졌는지 확인합니다.

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

3. ANA 경로 상태를 확인한다.


```
# nvme list-subsys/dev/nvme0n1
Nvme-subsysf0 - NQN=nqn.1992-
08.com.netapp:sn.341541339b9511e8a9b500a098c80f09:subsystem.sles_117_nvme_ss_10_0
\
+- nvme0 fc traddr=nn-0x202c00a098c80f09:pn-0x202d00a098c80f09
host_traddr=nn-0x20000090fae0ec61:pn-0x10000090fae0ec61 live optimized
+- nvme1 fc traddr=nn-0x207300a098dfdd91:pn-0x207600a098dfdd91
host_traddr=nn-0x200000109b1c1204:pn-0x100000109b1c1204 live
inaccessible
+- nvme2 fc traddr=nn-0x207300a098dfdd91:pn-0x207500a098dfdd91
host_traddr=nn-0x200000109b1c1205:pn-0x100000109b1c1205 live optimized
+- nvme3 fc traddr=nn-0x207300a098dfdd91:pn-0x207700a098dfdd91 host
traddr=nn-0x200000109b1c1205:pn-0x100000109b1c1205 live inaccessible
```

4. ONTAP 장치용 NetApp 플러그인을 확인합니다.

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

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

알려진 문제

알려진 문제가 없습니다.

Broadcom NVMe/FC에 대해 1MB I/O 크기를 활성화합니다

ONTAP는 컨트롤러 식별 데이터에서 MDTs(MAX Data 전송 크기)를 8로 보고합니다. 이는 최대 I/O 요청 크기가 1MB까지 될 수 있음을 의미합니다. Broadcom NVMe/FC 호스트에 대해 1MB 크기의 I/O 요청을 발행하려면 매개 변수 값을 `lpfc_sg_seg_cnt` 기본값인 64에서 256으로 늘려야 합니다.



다음 단계는 Qlogic NVMe/FC 호스트에는 적용되지 않습니다.

단계

1. `lpfc_sg_seg_cnt` 매개변수를 256으로 설정합니다.

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

```
options lpfc lpfc_sg_seg_cnt=256
```

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

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

예상 값은 256입니다.

lpfc Verbose 로깅

NVMe/FC용 lpfc 드라이버를 설정합니다.

단계

1. 를 설정합니다 `lpfc_log_verbose` 다음 값 중 하나에 대한 드라이버 설정을 사용하여 NVMe/FC 이벤트를 기록합니다.

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

2. 값을 설정한 후 를 실행합니다 `dracut -f` 명령을 실행하여 호스트를 재부팅합니다.
3. 설정을 확인합니다.

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

저작권 정보

Copyright © 2024 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.