



NVMe over RoCE 설정

E-Series storage systems

NetApp
January 20, 2026

목차

NVMe over RoCE 설정	1
E-Series(NVMe over RoCE)에서 Linux 구성 지원 및 검토 제한 사항 확인	1
Linux 구성이 지원되는지 확인합니다	1
NVMe over RoCE 제한 확인	1
E-Series-Linux(NVMe over RoCE)에서 DHCP를 사용하여 IP 주소 구성	2
SMcli용 SANtricity Storage Manager(11.53 이하) 설치 - Linux(NVMe over RoCE)	2
SANtricity System Manager-Linux(NVMe over RoCE)를 사용하여 스토리지 구성	4
E-Series-Linux(NVMe over RoCE)에서 스위치 구성	5
E-Series-Linux의 호스트에서 RoCE를 통해 NVMe 이니시에이터를 설정합니다	5
E-Series-Linux에서 RoCE 연결을 통해 스토리지 어레이 NVMe를 구성합니다	10
E-Series-Linux(NVMe over RoCE)에서 호스트에서 스토리지 검색 및 연결	12
SANtricity System Manager-Linux(NVMe over RoCE)를 사용하여 호스트 생성	14
SANtricity System Manager-Linux(NVMe over RoCE)를 사용하여 볼륨 할당	15
E-Series-Linux(NVMe over RoCE)에서 호스트에 표시되는 볼륨 표시	16
E-Series-Linux(NVMe over RoCE)에서 호스트에서 페일오버 설정	17
SLES 12에 대해 DMMP(Device Mapper Multipath) 활성화	18
기본 NVMe 다중 경로를 사용하여 RHEL 8을 설정합니다	18
E-Series-Linux(NVMe over RoCE)에서 가상 장치 대상을 위한 NVMe 볼륨에 액세스	19
가상 디바이스는 입출력 타겟입니다	19
예	19
E-Series-Linux(NVMe over RoCE)에서 물리적 NVMe 장치 타겟에 대한 NVMe 볼륨 액세스	21
물리적 NVMe 장치는 I/O 타겟입니다	21
E-Series에서 파일 시스템 생성 - Linux SLES 12(NVMe over RoCE)	23
E-시리즈에서 파일 시스템 생성 - Linux RHEL 8, RHEL 9, RHEL 10, SLES 15 및 SLES 16(RoCE 기반 NVMe)	24
E-Series-Linux(NVMe over RoCE)에서 호스트에서 스토리지 액세스 확인	26
E-Series-Linux에 NVMe over RoCE 구성을 기록합니다	26
직접 연결 토폴로지	26
스위치 연결 토폴로지	27
호스트 식별자입니다	28
타겟 NQN	28
대상 NQN	29
호스트 이름 매핑 중	29

NVMe over RoCE 설정

E-Series(NVMe over RoCE)에서 Linux 구성 지원 및 검토 제한 사항 확인

첫 번째 단계에서는 Linux 구성이 지원되는지 확인하고 컨트롤러, 스위치, 호스트 및 복구 제한도 검토해야 합니다.

Linux 구성이 지원되는지 확인합니다

안정적인 운영을 보장하기 위해 구축 계획을 생성한 다음 NetApp 상호 운용성 매트릭스 툴(IMT) 을 사용하여 전체 구성이 지원되는지 확인하십시오.

단계

1. 로 이동합니다 ["NetApp 상호 운용성 매트릭스 툴"](#).
2. 솔루션 검색 * 타일을 클릭합니다.
3. 메뉴: 프로토콜 [SAN 호스트] 영역에서 * E-Series SAN 호스트 * 옆에 있는 * 추가 * 버튼을 클릭합니다.
4. 검색 조건 구체화 보기 * 를 클릭합니다.

검색 조건 구체화 섹션이 표시됩니다. 이 섹션에서는 적용되는 프로토콜과 운영 체제, NetApp OS 및 호스트 다중 경로 드라이버와 같은 구성의 다른 기준을 선택할 수 있습니다.

5. 구성에 대해 알고 있는 기준을 선택한 다음 어떤 호환 구성 요소가 적용되는지 확인합니다.
6. 필요한 경우 도구에 규정된 운영 체제 및 프로토콜을 업데이트합니다.

선택한 구성에 대한 자세한 내용은 오른쪽 페이지 화살표를 클릭하여 지원되는 구성 보기 페이지에서 액세스할 수 있습니다.

NVMe over RoCE 제한 확인

NVMe over RoCE를 사용하기 전에 을(를) 참조하십시오 ["NetApp 상호 운용성 매트릭스 툴"](#) 최신 컨트롤러, 호스트 및 복구 제한 사항을 검토합니다.

스위치 제한



* 데이터 손실 위험. * NVMe over RoCE 환경에서 데이터 손실 위험을 제거하려면 스위치의 글로벌 일시 중지 컨트롤과 함께 흐름 제어를 활성화해야 합니다.

스토리지 및 재해 복구 제한 사항

- 비동기식 및 동기식 미러링은 지원되지 않습니다.
- 씬 프로비저닝(씬 볼륨 생성)은 지원되지 않습니다.

E-Series-Linux(NVMe over RoCE)에서 DHCP를 사용하여 IP 주소 구성

관리 스테이션과 스토리지 어레이 간의 통신을 구성하려면 DHCP(Dynamic Host Configuration Protocol)를 사용하여 IP 주소를 제공합니다.

시작하기 전에

다음 사항을 확인하십시오.

- 스토리지 관리 포트와 동일한 서브넷에 설치 및 구성된 DHCP 서버입니다.

이 작업에 대해

각 스토리지 어레이에는 1개의 컨트롤러(단일) 또는 2개의 컨트롤러(이중)가 있으며, 각 컨트롤러에는 2개의 스토리지 관리 포트가 있습니다. 각 관리 포트에는 IP 주소가 할당됩니다.

다음 지침은 두 개의 컨트롤러가 있는 스토리지 배열(이중 구성)을 나타냅니다.

단계

1. 아직 연결하지 않은 경우 이더넷 케이블을 관리 스테이션과 각 컨트롤러(A 및 B)의 관리 포트 1에 연결하십시오.

DHCP 서버는 각 컨트롤러의 포트 1에 IP 주소를 할당합니다.



두 컨트롤러 중 하나에서 관리 포트 2를 사용하지 마십시오. 포트 2는 NetApp 기술 담당자가 사용하도록 예약되어 있습니다.



이더넷 케이블을 분리했다가 다시 연결하거나 스토리지 배열의 전원을 껐다가 켜면 DHCP는 IP 주소를 다시 할당합니다. 이 프로세스는 고정 IP 주소가 구성될 때까지 수행됩니다. 케이블을 분리하거나 배열의 전원을 껐다가 켜는 것을 피하는 것이 좋습니다.

스토리지 배열이 30초 이내에 DHCP 할당 IP 주소를 가져올 수 없는 경우, 다음의 기본 IP 주소가 설정됩니다:

- 컨트롤러 A, 포트 1: 169.254.128.101
 - 컨트롤러 B, 포트 1: 169.254.128.102
 - 서브넷 마스크: 255.255.0.0
2. 각 컨트롤러 뒷면에서 MAC 주소 레이블을 찾은 다음 네트워크 관리자에게 각 컨트롤러의 포트 1에 대한 MAC 주소를 제공합니다.

네트워크 관리자는 각 컨트롤러의 IP 주소를 확인하기 위해 MAC 주소가 필요합니다. 브라우저를 통해 스토리지 시스템에 연결하려면 IP 주소가 필요합니다.

SMcli용 SANtricity Storage Manager(11.53 이하) 설치 - Linux(NVMe over RoCE)

SANtricity 소프트웨어 11.53 이하를 사용하는 경우, 관리 스테이션에 SANtricity 스토리지 관리자 소프트웨어를 설치하여 어레이를 관리할 수 있습니다.

SANtricity 스토리지 관리자는 추가 관리 작업을 위한 CLI(Command Line Interface)와 I/O 경로를 통해 호스트 구성 정보를 스토리지 어레이 컨트롤러로 푸시하는 Host Context Agent를 포함합니다.



SANtricity 소프트웨어 11.60 이상을 사용하는 경우 다음 단계를 수행할 필요가 없습니다. SANtricity 보안 CLI(SMcli)는 SANtricity OS에 포함되어 있으며 SANtricity 시스템 관리자를 통해 다운로드할 수 있습니다. SANtricity System Manager를 통해 SMcli를 다운로드하는 방법에 대한 자세한 내용은 [참조하십시오 "SANtricity System Manager 온라인 도움말에서 CLI\(Command Line Interface\) 항목을 다운로드하십시오"](#)



SANtricity 소프트웨어 버전 11.80.1부터 호스트 컨텍스트 에이전트는 더 이상 지원되지 않습니다.

시작하기 전에

다음 사항을 확인하십시오.

- SANtricity 소프트웨어 11.53 이전 버전.
- 관리자 또는 고급 사용자 권한을 수정합니다.
- 다음과 같은 최소 요구 사항이 있는 SANtricity Storage Manager 클라이언트용 시스템:
 - RAM *: Java Runtime Engine용 2GB
 - * 디스크 공간 *: 5GB
 - * OS/아키텍처 *: 지원되는 운영 체제 버전 및 아키텍처를 결정하는 지침은 [참조하십시오 "NetApp 지원"](#). 다운로드 * 탭에서 다운로드 [E-Series SANtricity 스토리지 관리자] 메뉴로 이동합니다.

이 작업에 대해

이 작업에서는 데이터 호스트에 Linux를 사용할 때 Windows와 Linux가 모두 공통 관리 스테이션 플랫폼이기 때문에 Windows 및 Linux OS 플랫폼 모두에 SANtricity 스토리지 관리자를 설치하는 방법을 설명합니다.

단계

1. 에서 SANtricity 소프트웨어 릴리스를 다운로드합니다 ["NetApp 지원"](#). 다운로드 * 탭에서 다운로드 [E-Series SANtricity 스토리지 관리자] 메뉴로 이동합니다.
2. SANtricity 설치 프로그램을 실행합니다.

Windows	리눅스
SMIA*.exe 설치 패키지를 두 번 클릭하여 설치를 시작합니다.	a. SMIA*.BIN 설치 패키지가 있는 디렉터리로 이동합니다. b. temp 마운트 지점에 실행 권한이 없는 경우 'IATEMPDIR' 변수를 설정합니다. 예: "IATEMPDIR=/root./SMIA-LINUX64-11.25.0A00.0002.BIN" c. 파일에 대한 실행 권한을 부여하려면 "chmod + x SMIA *.bin" 명령을 실행합니다. d. './SMIA *.BIN' 명령어를 실행하여 설치 프로그램을 시작한다.

3. 설치 마법사를 사용하여 관리 스테이션에 소프트웨어를 설치합니다.

SANtricity System Manager-Linux(NVMe over RoCE)를 사용하여 스토리지 구성

스토리지 배열을 구성하려면 SANtricity System Manager에서 설치 마법사를 사용할 수 있습니다.

SANtricity 시스템 관리자는 각 컨트롤러에 내장된 웹 기반 인터페이스입니다. 사용자 인터페이스에 액세스하려면 브라우저에서 컨트롤러의 IP 주소를 가리킵니다. 설치 마법사를 사용하면 시스템 구성을 시작할 수 있습니다.

시작하기 전에

다음 사항을 확인하십시오.

- 대역 외 관리.
- 다음 브라우저 중 하나가 포함된 SANtricity System Manager에 액세스하기 위한 관리 스테이션입니다.

브라우저	최소 버전
Google Chrome	89
Microsoft Edge를 참조하십시오	90
Mozilla Firefox	80
사파리	14

이 작업에 대해

System Manager를 열거나 브라우저를 새로 고치면 마법사가 자동으로 다시 시작되고, 다음 중 _ 개 이상의 조건이 충족됩니다.

- 풀 및 볼륨 그룹이 감지되지 않습니다.
- 감지된 워크로드가 없습니다.
- 알림이 구성되지 않았습니다.

단계

1. 브라우저에서 'https://<DomainNameOrIPAddress>' URL을 입력합니다

IPAddress는 스토리지 배열 컨트롤러 중 하나의 주소입니다.

구성되지 않은 어레이에서 SANtricity 시스템 관리자를 처음 열면 관리자 암호 설정 프롬프트가 나타납니다. 역할 기반 액세스 관리는 관리자, 지원, 보안 및 모니터링의 네 가지 로컬 역할을 구성합니다. 마지막 세 개의 역할에는 추측할 수 없는 임의의 암호가 있습니다. admin 역할의 암호를 설정한 후 admin 자격 증명을 사용하여 모든 암호를 변경할 수 있습니다. 4개의 로컬 사용자 역할에 대한 자세한 내용은 SANtricity System Manager 사용자 인터페이스에서 제공되는 온라인 도움말을 참조하십시오.

2. 관리자 암호 설정 및 암호 확인 필드에 관리자 역할에 대한 System Manager 암호를 입력한 다음 * 암호 설정 * 을 클릭합니다.

구성된 풀, 볼륨 그룹, 워크로드 또는 알림이 없는 경우 설정 마법사가 시작됩니다.

3. 설정 마법사를 사용하여 다음 작업을 수행합니다.

- * 하드웨어(컨트롤러 및 드라이브) 확인 * — 스토리지 배열의 컨트롤러 및 드라이브 수를 확인합니다. 어레이에 이름을 할당합니다.
- * 호스트 및 운영 체제 확인 * — 스토리지 배열이 액세스할 수 있는 호스트 및 운영 체제 유형을 확인합니다.
- * 풀 수락 * — 빠른 설치 방법에 대해 권장되는 풀 구성을 수락합니다. 풀은 드라이브의 논리적 그룹입니다.
- * 경고 구성 * — 스토리지 배열에 문제가 발생하면 System Manager가 자동 알림을 수신할 수 있도록 합니다.
- * AutoSupport 활성화 * — 스토리지 어레이의 상태를 자동으로 모니터링하고 기술 지원 부서에 디스패치를 보냅니다.

4. 볼륨을 아직 생성하지 않은 경우 Storage [Volumes > Create > Volume] 메뉴로 이동하여 생성합니다.

자세한 내용은 SANtricity 시스템 관리자의 온라인 도움말을 참조하십시오.

E-Series-Linux(NVMe over RoCE)에서 스위치 구성

NVMe over RoCE에 대한 공급업체의 권장 사항에 따라 스위치를 구성합니다. 이러한 권장 사항에는 구성 지시문과 코드 업데이트가 모두 포함될 수 있습니다.



* 데이터 손실 위험. * NVMe over RoCE 환경에서 데이터 손실 위험을 제거하려면 스위치의 글로벌 일시 중지 컨트롤과 함께 흐름 제어를 활성화해야 합니다.

단계

1. 모범 사례 구성으로 이더넷 중지 프레임 흐름 제어 * 엔드 투 엔드 * 를 활성화합니다.
2. 사용자 환경에 가장 적합한 구성을 선택하는 방법은 네트워크 관리자에게 문의하십시오.

E-Series-Linux의 호스트에서 RoCE를 통해 NVMe 이니시에이터를 설정합니다

RoCE 환경의 NVMe 이니시에이터 구성에는 RDMA 코어 및 NVMe-CLI 패키지 설치 및 구성, 이니시에이터 IP 주소 구성, 호스트에 NVMe-oF 계층 설정 등이 포함됩니다.

시작하기 전에

최신 호환 RHEL 8, RHEL 9, RHEL 10, SLES 12, SLES 15 또는 SLES 16 서비스 팩 운영 체제를 실행 중이어야 합니다. 보세요 ["NetApp 상호 운용성 매트릭스 툴"](#) 최신 요구 사항 전체 목록을 보려면 다음을 참조하십시오.

단계

1. RDMA 및 NVMe-CLI 패키지를 설치합니다.

SLES 12, SLES 15 또는 SLES 16

```
# zypper install rdma-core
# zypper install nvme-cli
```

RHEL 8, RHEL 9 또는 RHEL 10

```
# yum install rdma-core
# yum install nvme-cli
```

2. RHEL 8 및 RHEL 9의 경우 네트워크 스크립트를 설치합니다.

RHEL 8 *

```
# yum install network-scripts
```

RHEL 9 *

```
# yum install NetworkManager-initscripts-updown
```

3. 호스트를 스토리지에 구성하는 데 사용되는 호스트 NQN을 가져옵니다.

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

4. NVMe over RoCE를 연결하는 데 사용되는 이더넷 포트에서 IPv4 IP 주소를 설정합니다. 각 네트워크 인터페이스에 대해 해당 인터페이스에 대한 다양한 변수가 포함된 구성 스크립트를 생성합니다.

이 단계에서 사용되는 변수는 서버 하드웨어와 네트워크 환경을 기반으로 합니다. 이 변수에는 IPADDR, Gateway 등이 있다. 다음은 SLES 및 RHEL의 예제 지침입니다.

◦ SLES 12 및 SLES 15 *

다음 내용으로 예제 파일 '/etc/sysconfig/network/ifcfg-eth4'를 생성합니다.


```
BOOTPROTO='static'
BROADCAST=
ETHTOOL_OPTIONS=
IPADDR='192.168.1.87/24'
GATEWAY='192.168.1.1'
MTU=
NAME='MT27800 Family [ConnectX-5]'
NETWORK=
REMOTE_IPADDR=
STARTMODE='auto'
```

그런 다음 예제 파일 '/etc/sysconfig/network/ifcfg-eth5'를 생성합니다.

```
BOOTPROTO='static'
BROADCAST=
ETHTOOL_OPTIONS=
IPADDR='192.168.2.87/24'
GATEWAY='192.168.2.1'
MTU=
NAME='MT27800 Family [ConnectX-5]'
NETWORK=
REMOTE_IPADDR=
STARTMODE='auto'
```

RHEL 8 *

다음 내용으로 예제 파일 '/etc/sysconfig/network-scripts/ifcfg-eth4'를 생성합니다.

```
BOOTPROTO='static'
BROADCAST=
ETHTOOL_OPTIONS=
IPADDR='192.168.1.87/24'
GATEWAY='192.168.1.1'
MTU=
NAME='MT27800 Family [ConnectX-5]'
NETWORK=
REMOTE_IPADDR=
STARTMODE='auto'
```

그런 다음 예제 파일 '/etc/sysconfig/network-scripts/ifcfg-eth5'를 생성합니다.

```
BOOTPROTO='static'
BROADCAST=
ETHTOOL_OPTIONS=
IPADDR='192.168.2.87/24'
GATEWAY='192.168.2.1'
MTU=
NAME='MT27800 Family [ConnectX-5]'
NETWORK=
REMOTE_IPADDR=
STARTMODE='auto'
```

RHEL 9, RHEL 10 또는 SLES 16

를 사용합니다 nmtui 도구를 사용하여 연결을 활성화하고 편집합니다. 다음은 예제 파일입니다 /etc/NetworkManager/system-connections/eth4.nmconnection 이 도구는 다음을 생성합니다.

```
[connection]
id=eth4
uuid=<unique uuid>
type=ethernet
interface-name=eth4

[ethernet]
mtu=4200

[ipv4]
address1=192.168.1.87/24
method=manual

[ipv6]
addr-gen-mode=default
method=auto

[proxy]
```

다음은 예제 파일입니다 /etc/NetworkManager/system-connections/eth5.nmconnection 이 도구는 다음을 생성합니다.

```
[connection]
id=eth5
uuid=<unique uuid>
type=ethernet
interface-name=eth5

[ethernet]
mtu=4200

[ipv4]
address1=192.168.2.87/24
method=manual

[ipv6]
addr-gen-mode=default
method=auto

[proxy]
```

5. 네트워크 인터페이스를 활성화합니다.

```
# ifup eth4
# ifup eth5
```

6. 호스트에서 NVMe-oF 계층을 설정합니다. 에서 다음 파일을 작성합니다 /etc/modules-load.d/ 를 로드합니다 nvme_rdma 커널 모듈을 사용하고 재부팅 후에도 커널 모듈이 항상 켜져 있는지 확인합니다.

```
# cat /etc/modules-load.d/nvme_rdma.conf
nvme_rdma
```

7. 호스트를 재부팅합니다.

를 확인합니다 nvme_rdma 커널 모듈이 로드되었습니다. 다음 명령을 실행합니다.

```
# lsmod | grep nvme
nvme_rdma          36864  0
nvme_fabrics       24576  1 nvme_rdma
nvme_core          114688  5 nvme_rdma,nvme_fabrics
rdma_cm            114688  7
rpcrdma,ib_srpt,ib_srp,nvme_rdma,ib_iser,ib_isert,rdma_ucm
ib_core            393216  15
rdma_cm,ib_ipoib,rpcrdma,ib_srpt,ib_srp,nvme_rdma,iw_cm,ib_iser,ib_umad,
ib_isert,rdma_ucm,ib_uverbs,mlx5_ib,qedr,ib_cm
t10_pi             16384  2 sd_mod,nvme_core
```

E-Series-Linux에서 RoCE 연결을 통해 스토리지 어레이 NVMe를 구성합니다

컨트롤러에 NVMe over RoCE(RDMA over Converged Ethernet)에 대한 연결이 포함되어 있는 경우 하드웨어 페이지 또는 SANtricity System Manager의 시스템 페이지에서 NVMe 포트 설정을 구성할 수 있습니다.

시작하기 전에

다음 사항을 확인하십시오.

- 컨트롤러의 NVMe over RoCE 호스트 포트. 그렇지 않으면 System Manager에서 NVMe over RoCE 설정을 사용할 수 없습니다.
- 호스트 연결의 IP 주소입니다.

이 작업에 대해

NVMe over RoCE 구성은 * 하드웨어 * 페이지 또는 메뉴: 설정 [시스템] 에서 액세스할 수 있습니다. 이 작업에서는 하드웨어 페이지에서 포트를 구성하는 방법을 설명합니다.



NVMe over RoCE 설정 및 기능은 스토리지 어레이의 컨트롤러에 NVMe over RoCE 포트가 포함된 경우에만 표시됩니다.

단계

1. System Manager 인터페이스에서 * Hardware * 를 선택합니다.
2. 구성할 NVMe over RoCE 포트가 있는 컨트롤러를 클릭합니다.



컨트롤러의 상황에 맞는 메뉴가 나타납니다.

3. RoCE 포트를 통한 NVMe 구성 * 을 선택합니다.

RoCE 포트를 통한 NVMe 구성 * 대화 상자가 열립니다.

4. 드롭다운 목록에서 구성할 포트를 선택한 후 * 다음 * 을 클릭합니다.
5. 사용할 포트 구성 설정을 선택한 후 * 다음 * 을 클릭합니다.




모든 포트 설정을 보려면 대화 상자 오른쪽에 있는 * 추가 포트 설정 표시 * 링크를 클릭합니다.

포트 설정	설명
이더넷 포트 속도를 구성했습니다	<p>원하는 속도를 선택합니다. 드롭다운 목록에 표시되는 옵션은 네트워크에서 지원할 수 있는 최대 속도(예: 10Gbps)에 따라 달라집니다. 가능한 값은 다음과 같습니다.</p> <ul style="list-style-type: none"> • 자동 협상 • 10Gbps • 25Gbps • 40Gbps • 50Gbps • 100Gbps • 200Gbps <div> <div></div> <div>200GB 가능 HIC를 QSFP 56 케이블로 연결한 경우 Mellanox 스위치 및/또는 어댑터에 연결하는 경우에만 자동 협상을 사용할 수 있습니다.</div> </div> <div> <div></div> <div>구성된 NVMe over RoCE 포트 속도는 선택한 포트에서 SFP의 속도 기능과 일치해야 합니다. 모든 포트는 동일한 속도로 설정되어야 합니다.</div> </div>
IPv4 및/또는 IPv6 사용 을 설정합니다	IPv4 및 IPv6 네트워크에 대한 지원을 활성화하려면 하나 또는 두 옵션을 모두 선택하십시오.
MTU 크기(* 추가 포트 설정 표시 * 를 클릭하여 사용 가능)	필요한 경우 MTU(Maximum Transmission Unit)에 대한 새 크기를 바이트 단위로 입력합니다. 기본 MTU 크기는 프레임당 1500바이트입니다. 1500 ~ 9000 사이의 값을 입력해야 합니다.

IPv4 사용 * 을 선택한 경우 * 다음 * 을 클릭하면 IPv4 설정을 선택할 수 있는 대화 상자가 열립니다. IPv6 사용 * 을 선택한 경우 * 다음 * 을 클릭하면 IPv6 설정을 선택할 수 있는 대화 상자가 열립니다. 두 옵션을 모두 선택한 경우 IPv4 설정에 대한 대화 상자가 먼저 열리고 * 다음 * 을 클릭하면 IPv6 설정에 대한 대화 상자가 열립니다.

- IPv4 및/또는 IPv6 설정을 자동 또는 수동으로 구성합니다. 모든 포트 설정을 보려면 대화 상자 오른쪽에 있는 * 추가 설정 표시 * 링크를 클릭합니다.

포트 설정	설명
DHCP 서버에서 자동으로 구성을 가져옵니다	구성을 자동으로 가져오려면 이 옵션을 선택합니다.

포트 설정	설명
수동으로 정적 설정을 지정합니다	<p>이 옵션을 선택한 다음 필드에 정적 주소를 입력합니다. IPv4의 경우 네트워크 서브넷 마스크 및 게이트웨이를 포함합니다. IPv6의 경우 라우팅 가능한 IP 주소와 라우터 IP 주소를 포함합니다.</p> <p> 라우팅 가능한 IP 주소가 하나만 있는 경우 나머지 주소를 0:0:0:0:0:0:0:0으로 설정합니다.</p>
VLAN 지원을 활성화합니다(* 추가 설정 표시 * 를 클릭하여 사용 가능).	<p> 이 옵션은 iSCSI 환경에서만 사용할 수 있습니다. NVMe over RoCE 환경에서는 사용할 수 없습니다.</p>
이더넷 우선 순위 활성화(* 추가 설정 표시 * 를 클릭하여 사용 가능)	<p> 이 옵션은 iSCSI 환경에서만 사용할 수 있습니다. NVMe over RoCE 환경에서는 사용할 수 없습니다.</p>

7. 마침 * 을 클릭합니다.

E-Series-Linux(NVMe over RoCE)에서 호스트에서 스토리지 검색 및 연결

SANtricity System Manager에서 각 호스트를 정의하기 전에 호스트에서 타겟 컨트롤러 포트를 검색한 다음 NVMe 연결을 설정해야 합니다.

단계

1. 다음 명령을 사용하여 모든 경로에 대해 NVMe-oF 타겟의 사용 가능한 서브시스템을 검색합니다.

```
nvme discover -t rdma -a target_ip_address
```

이 명령에서 target_ip_address는 target 포트의 IP 주소입니다.



"NVMe 검색" 명령은 호스트 액세스에 관계없이 하위 시스템의 모든 컨트롤러 포트를 검색합니다.

```
# nvme discover -t rdma -a 192.168.1.77
Discovery Log Number of Records 2, Generation counter 0
=====Discovery Log Entry 0=====
trtype:  rdma
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  0
trsvcid: 4420
subnqn:  nqn.1992-08.com.netapp:5700.600a098000a527a7000000005ab3af94
traddr:  192.168.1.77
rdma_prtype: roce
rdma_qptype: connected
rdma_cms:   rdma-cm
rdma_pkey: 0x0000
=====Discovery Log Entry 1=====
trtype:  rdma
adrfam:  ipv4
subtype: nvme subsystem
treq:    not specified
portid:  1
trsvcid: 4420
subnqn:  nqn.1992-08.com.netapp:5700.600a098000a527a7000000005ab3af94
traddr:  192.168.2.77
rdma_prtype: roce
rdma_qptype: connected
rdma_cms:   rdma-cm
rdma_pkey: 0x0000
```

2. 다른 모든 연결에도 1단계를 반복합니다.

3. "NVMe connect -t RDMA -n discovered_sub_nqn -a target_ip_address -Q queue_depth_setting -l controller_loss_timeout_period" 명령을 사용하여 첫 번째 경로에서 검색된 하위 시스템에 연결합니다



위에 나열된 명령은 재부팅 후에도 유지되지 않습니다. NVMe 연결을 재설정하기 위해서는 재부팅 후 NVMe connect 명령을 실행해야 한다.



호스트가 액세스할 수 없는 검색된 포트에 대한 연결이 설정되지 않았습니다.



이 명령을 사용하여 포트 번호를 지정하면 연결이 실패합니다. 기본 포트는 연결에 대해 설정된 유일한 포트입니다.



권장 대기열 길이 설정은 1024입니다. 다음 예제와 같이 "-Q 1024" 명령줄 옵션을 사용하여 128의 기본 설정을 1024로 재정의합니다.



권장 컨트롤러 손실 시간 제한 시간(초)은 60분(3600초)입니다. 다음 예제와 같이 '-l 3600' 명령줄 옵션을 사용하여 기본 설정인 600초를 3600초로 재정의합니다.

```
# nvme connect -t rdma -a 192.168.1.77 -n nqn.1992-08.com.netapp:5700.600a098000a527a7000000005ab3af94 -Q 1024 -l 3600
# nvme connect -t rdma -a 192.168.2.77 -n nqn.1992-08.com.netapp:5700.600a098000a527a7000000005ab3af94 -Q 1024 -l 3600
```

4. 3단계를 반복하여 검색된 하위 시스템을 두 번째 경로에 연결합니다.

SANtricity System Manager-Linux(NVMe over RoCE)를 사용하여 호스트 생성

SANtricity 시스템 관리자를 사용하여 스토리지 배열에 데이터를 전송하는 호스트를 정의합니다. 호스트 정의는 스토리지 배열에 연결된 호스트를 알려 주는 단계 중 하나이며 볼륨에 대한 I/O 액세스를 허용하는 데 필요한 단계 중 하나입니다.

이 작업에 대해

호스트를 정의할 때 다음 지침을 염두에 두십시오.

- 호스트와 연결된 호스트 식별자 포트를 정의해야 합니다.
- 호스트에 할당된 시스템 이름과 동일한 이름을 제공해야 합니다.
- 선택한 이름이 이미 사용 중인 경우에는 이 작업이 성공하지 않습니다.
- 이름의 길이는 30자를 초과할 수 없습니다.

단계

1. 스토리지 [호스트] 메뉴를 선택합니다.
2. MENU: Create [Host] 를 클릭합니다.

Create Host 대화 상자가 나타납니다.

3. 필요에 따라 호스트 설정을 선택합니다.

설정	설명
이름	새 호스트의 이름을 입력합니다.
호스트 운영 체제 유형입니다	드롭다운 목록에서 다음 옵션 중 하나를 선택합니다. <ul style="list-style-type: none"> • SANtricity 11.60 이상용 * Linux * • * Linux DM-MP(커널 3.10 이상) * - SANtricity 11.60 이전 버전

설정	설명
호스트 인터페이스 유형입니다	사용할 호스트 인터페이스 유형을 선택합니다. 구성된 어레이에 사용 가능한 호스트 인터페이스 유형이 하나만 있는 경우 이 설정을 선택할 수 없습니다.
호스트 포트	<p>다음 중 하나를 수행합니다.</p> <ul style="list-style-type: none"> * I/O 인터페이스 * 를 선택합니다 <p>호스트 포트가 로그인한 경우 목록에서 호스트 포트 식별자를 선택할 수 있습니다. 이 방법이 권장되는 방법입니다.</p> <ul style="list-style-type: none"> * 수동 추가 * <p>호스트 포트가 로그인하지 않은 경우 호스트에서 /etc/NVMe/hostnqn을 보고 hostnqn 식별자를 찾아 호스트 정의에 연결합니다.</p> <p>호스트 포트 식별자를 수동으로 입력하거나 /etc/NVMe/hostnqn 파일(한 번에 하나씩)에서 * 호스트 포트 * 필드로 복사/붙여넣기할 수 있습니다.</p> <p>호스트와 연결하려면 한 번에 하나의 호스트 포트 식별자를 추가해야 하지만 호스트와 연결된 식별자를 계속 선택할 수 있습니다. 각 식별자는 * 호스트 포트 * 필드에 표시됩니다. 필요한 경우 옆에 있는 * X * 를 선택하여 식별자를 제거할 수도 있습니다.</p>

4. Create * 를 클릭합니다.

결과

호스트가 성공적으로 생성되면 SANtricity System Manager는 호스트에 대해 구성된 각 호스트 포트에 대한 기본 이름을 생성합니다.

기본 별칭은 "<Hostname_Port Number>"입니다. 예를 들어, 호스트 IPT에 대해 생성된 첫 번째 포트의 기본 별칭은 IPT_1입니다.

SANtricity System Manager-Linux(NVMe over RoCE)를 사용하여 볼륨 할당

I/O 작업에 사용할 수 있도록 호스트 또는 호스트 클러스터에 볼륨(네임스페이스)을 할당해야 합니다. 이 할당은 호스트 또는 호스트 클러스터에 스토리지 배열의 하나 이상의 네임스페이스에 대한 액세스 권한을 부여합니다.

이 작업에 대해

볼륨을 할당할 때 다음 지침을 엄두에 두십시오.

- 한 번에 하나의 호스트 또는 호스트 클러스터에만 볼륨을 할당할 수 있습니다.
- 할당된 볼륨은 스토리지 배열의 컨트롤러 간에 공유됩니다.
- 호스트 또는 호스트 클러스터에서 볼륨에 액세스하는 데 동일한 네임스페이스 ID(NSID)를 두 번 사용할 수 없습니다. 고유한 NSID를 사용해야 합니다.

다음과 같은 조건에서는 볼륨을 할당할 수 없습니다.

- 모든 볼륨이 할당됩니다.
- 볼륨이 이미 다른 호스트 또는 호스트 클러스터에 할당되어 있습니다.

다음과 같은 조건에서는 볼륨을 할당할 수 없습니다.

- 유효한 호스트 또는 호스트 클러스터가 없습니다.
- 모든 볼륨 할당이 정의되었습니다.

할당되지 않은 모든 볼륨이 표시되지만 DA(Data Assurance)를 사용하거나 사용하지 않는 호스트의 기능은 다음과 같이 적용됩니다.

- DA 지원 호스트의 경우 DA 사용 또는 DA 사용 안 함 볼륨을 선택할 수 있습니다.
- DA를 사용할 수 없는 호스트의 경우 DA를 사용할 수 있는 볼륨을 선택하면 호스트에 볼륨을 할당하기 전에 시스템에서 자동으로 볼륨의 DA를 해제해야 한다는 경고가 표시됩니다.

단계

1. 스토리지 [호스트] 메뉴를 선택합니다.
2. 볼륨을 할당할 호스트 또는 호스트 클러스터를 선택한 다음 * 볼륨 할당 * 을 클릭합니다.

할당할 수 있는 모든 볼륨이 나열된 대화 상자가 나타납니다. 특정 볼륨을 쉽게 찾을 수 있도록 모든 열을 정렬하거나 * Filter * (필터 *) 상자에 원하는 항목을 입력할 수 있습니다.

3. 할당할 각 볼륨 옆의 확인란을 선택하거나 표 머리글에서 확인란을 선택하여 모든 볼륨을 선택합니다.
4. 작업을 완료하려면 * 배정 * 을 클릭하십시오.

결과

볼륨이나 볼륨을 호스트 또는 호스트 클러스터에 성공적으로 할당한 후 시스템은 다음 작업을 수행합니다.

- 할당된 볼륨은 다음으로 사용 가능한 NSID를 받습니다. 호스트는 NSID를 사용하여 볼륨에 액세스합니다.
- 사용자 제공 볼륨 이름이 호스트에 연결된 볼륨 목록에 나타납니다.

E-Series-Linux(NVMe over RoCE)에서 호스트에 표시되는 볼륨 표시

SMdevices 툴을 사용하여 호스트에 현재 표시된 볼륨을 볼 수 있습니다. 이 도구는 NVMe-CLI 패키지의 일부이며 'NVMe list' 명령 대신 사용할 수 있습니다.

E-Series 볼륨에 대한 각 NVMe 경로에 대한 정보를 보려면 'NVMe NetApp SMdevices[-o<format>]' 명령을 사용하십시오. format> 출력은 정상(-o가 사용되지 않는 경우 기본값), 열 또는 json이 될 수 있습니다.

```
# nvme netapp smdevices
/dev/nvme1n1, Array Name ICTM0706SYS04, Volume Name NVMe2, NSID 1, Volume
ID 000015bd5903df4a00a0980000af4462, Controller A, Access State unknown,
2.15GB
/dev/nvme1n2, Array Name ICTM0706SYS04, Volume Name NVMe3, NSID 2, Volume
ID 000015c05903e24000a0980000af4462, Controller A, Access State unknown,
2.15GB
/dev/nvme1n3, Array Name ICTM0706SYS04, Volume Name NVMe4, NSID 4, Volume
ID 00001bb0593a46f400a0980000af4462, Controller A, Access State unknown,
2.15GB
/dev/nvme1n4, Array Name ICTM0706SYS04, Volume Name NVMe6, NSID 6, Volume
ID 00001696593b424b00a0980000af4112, Controller A, Access State unknown,
2.15GB
/dev/nvme2n1, Array Name ICTM0706SYS04, Volume Name NVMe2, NSID 1, Volume
ID 000015bd5903df4a00a0980000af4462, Controller B, Access State unknown,
2.15GB
/dev/nvme2n2, Array Name ICTM0706SYS04, Volume Name NVMe3, NSID 2, Volume
ID 000015c05903e24000a0980000af4462, Controller B, Access State unknown,
2.15GB
/dev/nvme2n3, Array Name ICTM0706SYS04, Volume Name NVMe4, NSID 4, Volume
ID 00001bb0593a46f400a0980000af4462, Controller B, Access State unknown,
2.15GB
/dev/nvme2n4, Array Name ICTM0706SYS04, Volume Name NVMe6, NSID 6, Volume
ID 00001696593b424b00a0980000af4112, Controller B, Access State unknown,
2.15GB
```

E-Series-Linux(NVMe over RoCE)에서 호스트에서 페일오버 설정

스토리지 배열에 대한 중복 경로를 제공하기 위해 호스트에서 장애 조치를 실행하도록 구성할 수 있습니다.

시작하기 전에

시스템에 필요한 패키지를 설치해야 합니다.

- RHEL(Red Hat) 호스트의 경우 "rpm -q device-mapper-multipath"를 실행하여 패키지가 설치되어 있는지 확인합니다
- SLES 호스트의 경우 'rpm-q multipath-tools'를 실행하여 패키지가 설치되어 있는지 확인합니다



을 참조하십시오 ["NetApp 상호 운용성 매트릭스 툴"](#) 필요한 업데이트가 설치되었는지 확인하기 위해, 다중 경로가 SLES 또는 RHEL의 GA 버전과 제대로 작동하지 않을 수 있습니다.

이 작업에 대해

SLES 12는 RoCE를 통한 NVMe 멀티패싱을 위해 DMMP(Device Mapper Multipath)를 사용합니다. RHEL 8, RHEL 9, RHEL 10, SLES 15 및 SLES 16은 내장된 네이티브 NVMe 페일오버 기능을 사용합니다. 사용 중인 운영체제에 따라 멀티패스를 제대로 작동시키려면 추가적인 설정이 필요할 수 있습니다.

SLES 12에 대해 DMMP(Device Mapper Multipath) 활성화

기본적으로 DM-MP는 SLES에서 비활성화되어 있습니다. 호스트에서 DM-MP 구성 요소를 활성화하려면 다음 단계를 완료하세요.

단계

1. 다음 예에 표시된 대로 NVMe E-Series 장치 항목을 '/etc/multipath.conf' 파일의 devices 섹션에 추가합니다.

```
devices {  
    device {  
        vendor "NVME"  
        product "NetApp E-Series*"  
        path_grouping_policy group_by_prio  
        failback immediate  
        no_path_retry 30  
    }  
}
```

2. 시스템 부팅 시 시작되도록 'multipid'를 구성합니다.

```
# systemctl enable multipathd
```

3. 현재 실행되고 있지 않으면 '멀티페이드'를 시작합니다.

```
# systemctl start multipathd
```

4. '멀티페이드' 상태를 확인하여 활성 상태이고 실행 중인지 확인합니다.

```
# systemctl status multipathd
```

기본 NVMe 다중 경로를 사용하여 RHEL 8을 설정합니다

기본 NVMe Multipathing은 RHEL 8에서 기본적으로 비활성화되어 있으며 다음 절차를 사용하여 활성화해야 합니다.

1. 기본 NVMe 다중 경로를 사용하도록 'modprobe' 규칙을 설정합니다.

```
# echo "options nvme_core multipath=y" >> /etc/modprobe.d/50-nvme_core.conf
```

2. 새로운 '모데프로브' 파라미터로 'initramfs'를 다시 만듭니다.

```
# dracut -f
```

3. 기본 NVMe 다중 경로를 사용하도록 설정한 상태에서 서버를 재부팅합니다.

```
# reboot
```

4. 호스트를 부팅한 후 기본 NVMe Multipathing이 활성화되어 있는지 확인합니다.

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

- a. 명령 출력이 N인 경우 기본 NVMe Multipathing은 여전히 비활성화되어 있습니다.
- b. 명령 출력이 Y인 경우 기본 NVMe 다중경로 기능이 활성화되고 발견한 모든 NVMe 장치에서 이 기능을 사용할 것입니다.



SLES 15, SLES 16, RHEL 9 및 RHEL 10의 경우 네이티브 NVMe 멀티패싱이 기본적으로 활성화되어 있으므로 추가 구성이 필요하지 않습니다.

E-Series-Linux(NVMe over RoCE)에서 가상 장치 대상을 위한 NVMe 볼륨에 액세스

사용하는 OS(및 확장 경로 다중화 방법)를 기반으로 디바이스 타겟으로 연결되는 입출력을 구성할 수 있습니다.

SLES 12의 경우 I/O는 Linux 호스트에 의해 가상 장치 대상으로 전달됩니다. DM-MP는 이러한 가상 대상의 기본이 되는 물리적 경로를 관리합니다.

가상 디바이스는 입출력 타겟입니다

물리적 디바이스 경로가 아닌 DM-MP에서 생성한 가상 디바이스에만 입출력을 실행해야 합니다. 물리적 경로에 대한 입출력을 실행하는 경우 DM-MP가 페일오버 이벤트를 관리할 수 없으며 입출력이 실패합니다.

/dev/mapper의 dm 장치 또는 'symlink'를 통해 이러한 블록 장치에 액세스할 수 있습니다. 예를 들면 다음과 같습니다.

```
/dev/dm-1  
/dev/mapper/eui.00001bc7593b7f5f00a0980000af4462
```

예

다음 "NVMe list" 명령의 출력 예에서는 호스트 노드 이름과 네임스페이스 ID의 상관 관계를 보여 줍니다.

NODE	SN	MODEL	NAMESPACE
/dev/nvme1n1	021648023072	NetApp E-Series	10
/dev/nvme1n2	021648023072	NetApp E-Series	11
/dev/nvme1n3	021648023072	NetApp E-Series	12
/dev/nvme1n4	021648023072	NetApp E-Series	13
/dev/nvme2n1	021648023151	NetApp E-Series	10
/dev/nvme2n2	021648023151	NetApp E-Series	11
/dev/nvme2n3	021648023151	NetApp E-Series	12
/dev/nvme2n4	021648023151	NetApp E-Series	13

열	설명
'노드'	<p>노드 이름은 다음 두 부분으로 구성됩니다.</p> <ul style="list-style-type: none"> • nvme1은 컨트롤러 A를 나타내고 nvme2는 컨트롤러 B를 나타냅니다 • n1, n2 등의 표기법은 호스트 관점에서 네임스페이스 식별자를 나타냅니다. 이러한 식별자는 표에서 반복되고 컨트롤러 A는 한 번, 컨트롤러 B는 한 번 반복됩니다
'네임스페이스'	네임스페이스 열에는 스토리지 배열 관점에서 사용되는 식별자인 네임스페이스 ID(NSID)가 나열됩니다.

다음 '멀티패스-II' 출력에서는 최적화 경로가 프리오 값 50으로 표시되고 최적화되지 않은 경로는 프리오 값 10으로 표시됩니다.

Linux 운영체제는 'Status=active'로 표시된 경로 그룹에 대한 I/O를 라우팅하고, 'Status=enabled'로 나열된 경로 그룹은 파일오버에 사용할 수 있습니다.

```
eui.00001bc7593b7f500a0980000af4462 dm-0 NVME,NetApp E-Series
size=15G features='1 queue_if_no_path' hwhandler='0' wp=rw
|+- policy='service-time 0' prio=50 status=active
|  `- #:#:#:# nvme1n1 259:5 active ready running
`+- policy='service-time 0' prio=10 status=enabled
   `- #:#:#:# nvme2n1 259:9 active ready running

eui.00001bc7593b7f5f00a0980000af4462 dm-0 NVME,NetApp E-Series
size=15G features='1 queue_if_no_path' hwhandler='0' wp=rw
|+- policy='service-time 0' prio=0 status=enabled
|  `- #:#:#:# nvme1n1 259:5 failed faulty running
`+- policy='service-time 0' prio=10 status=active
   `- #:#:#:# nvme2n1 259:9 active ready running
```

라인 항목	설명
' policy='service-time 0' prio=50 status=active	이 선과 다음 줄은 NSID가 10인 namespace인 nvme1n1이 prio 값 50과 active의 '상태' 값으로 경로에 최적화되는 것을 보여 줍니다. 이 네임스페이스는 컨트롤러 A에서 소유합니다
' policy='service-time 0' prio=10 status=enabled'	이 선은 네임스페이스 10의 페일오버 경로를 보여 주며, 프리오 값이 10이고 '상태' 값이 '활성화됨'인 것을 나타냅니다. 현재 I/O가 이 경로의 네임스페이스로 보내지지 않습니다. 이 네임스페이스는 컨트롤러 B에서 소유합니다
' policy='service-time 0' prio=0 status=enabled'	이 예제는 컨트롤러 A가 재부팅되는 동안 다른 시점의 '멀티패스-I' 출력을 보여줍니다. 네임스페이스 10의 경로는 프리오 값이 0이고 '상태' 값이 '활성화됨'인 실패한 실행 경로로 표시됩니다.
' policy='service-time 0' prio=10 status=active	활성 경로는 nvme2를 가리키므로 I/O는 이 경로를 통해 컨트롤러 B로 리디렉션됩니다

E-Series-Linux(NVMe over RoCE)에서 물리적 NVMe 장치 타겟에 대한 NVMe 볼륨 액세스

사용 중인 OS 및 확장 경로 다중화 방법에 따라 디바이스 타겟에 대한 입출력을 구성할 수 있습니다.

RHEL 8, RHEL 9 및 SLES 15의 경우, I/O는 Linux 호스트가 물리적 NVMe 장치 타겟으로 지정합니다. 기본 NVMe 다중 경로 솔루션은 호스트에서 표시되는 분명한 단일 물리적 장치의 물리적 경로를 관리합니다.

물리적 NVMe 장치는 I/O 타겟입니다

의 링크에 대한 I/O를 실행하는 것이 가장 좋습니다 /dev/disk/by-id/nvme-eui.[uuid#] 물리적 NVMe 장치 경로로 직접 이동하는 것이 아니라 /dev/nvme[sys#]n[id#]. 이러한 두 위치 간의 링크는 다음 명령을 사용하여 찾을 수 있습니다.

```
# ls /dev/disk/by-id/ -l
lrwxrwxrwx 1 root root 13 Oct 18 15:14 nvme-
eui.0000320f5cad32cf00a0980000af4112 -> ../../nvme0n1
```

I/O가 에 실행됩니다 /dev/disk/by-id/nvme-eui.[uuid#] 을 통해 직접 전달됩니다 /dev/nvme[sys#]n[id#] 기본 NVMe 다중 경로 솔루션을 사용하여 모든 경로를 이 경로 아래에서 가상화했습니다.

다음을 실행하여 경로를 볼 수 있습니다.

```
# nvme list-subsys
```

예제 출력:

```
nvme-subsys0 - NQN=nqn.1992-
08.com.netapp:5700.600a098000a522500000000589aa8a6
\
+- nvme0 rdma traddr=192.4.21.131 trsvcid=4420 live
+- nvme1 rdma traddr=192.4.22.141 trsvcid=4420 live
```

"NVMe list -Subsys" 명령을 사용할 때 네임스페이스 장치를 지정하면 해당 네임스페이스에 대한 경로에 대한 추가 정보가 제공됩니다.

```
# nvme list-subsys /dev/nvme0n1
nvme-subsys0 - NQN=nqn.1992-
08.com.netapp:5700.600a098000af44620000000058d5dd96
\
+- nvme0 rdma traddr=192.168.130.101 trsvcid=4420 live non-optimized
+- nvme1 rdma traddr=192.168.131.101 trsvcid=4420 live non-optimized
+- nvme2 rdma traddr=192.168.130.102 trsvcid=4420 live optimized
+- nvme3 rdma traddr=192.168.131.102 trsvcid=4420 live optimized
```

또한 다중 경로 명령에 후크를 사용하여 기본 파일오버를 위한 경로 정보를 볼 수도 있습니다.

```
#multipath -ll
```



경로 정보를 보려면 `/etc/multipath.conf`에서 다음을 설정해야 합니다.

```
defaults {
    enable_foreign nvme
}
```



이 방법은 RHEL 10에서는 더 이상 작동하지 않습니다. RHEL 9 이하 버전과 SLES 16 이하 버전에서 작동합니다.

예제 출력:


```
eui.0000a0335c05d57a00a0980000a5229d [nvme]:nvme0n9 NVMe,Netapp E-
Series,08520001
size=4194304 features='n/a' hwhandler='ANA' wp=rw
|+- policy='n/a' prio=50 status=optimized
|  `-- 0:0:1 nvme0c0n1 0:0 n/a optimized    live
`+- policy='n/a' prio=10 status=non-optimized
  `-- 0:1:1 nvme0c1n1 0:0 n/a non-optimized    live
```

E-Series에서 파일 시스템 생성 - Linux SLES 12(NVMe over RoCE)

SLES 12의 경우 네임스페이스에 파일 시스템을 만들고 파일 시스템을 마운트합니다.

단계

1. '/dev/mapper/dm' 디바이스 목록을 가져오려면 'multipath -ll' 명령을 실행합니다.

```
# multipath -ll
```

이 명령의 결과는 dm-19와 dm-16이라는 두 가지 장치를 보여 줍니다.

```
eui.00001ffe5a94ff8500a0980000af4444 dm-19 NVME,NetApp E-Series
size=10G features='1 queue_if_no_path' hwhandler='0' wp=rw
|+- policy='service-time 0' prio=50 status=active
|  |-- #:#:#:# nvme0n19 259:19  active ready running
|  `-- #:#:#:# nvme1n19 259:115 active ready running
`+- policy='service-time 0' prio=10 status=enabled
  |-- #:#:#:# nvme2n19 259:51  active ready running
  `-- #:#:#:# nvme3n19 259:83  active ready running
eui.00001fd25a94fef000a0980000af4444 dm-16 NVME,NetApp E-Series
size=16G features='1 queue_if_no_path' hwhandler='0' wp=rw
|+- policy='service-time 0' prio=50 status=active
|  |-- #:#:#:# nvme0n16 259:16  active ready running
|  `-- #:#:#:# nvme1n16 259:112 active ready running
`+- policy='service-time 0' prio=10 status=enabled
  |-- #:#:#:# nvme2n16 259:48  active ready running
  `-- #:#:#:# nvme3n16 259:80  active ready running
```

2. 각 '/dev/mapper/eui -' 디바이스의 파티션에 파일 시스템을 생성합니다.

파일 시스템을 생성하는 방법은 선택한 파일 시스템에 따라 다릅니다. 이 예에서는 ext4 파일 시스템을 만드는 방법을 보여 줍니다.

```
# mkfs.ext4 /dev/mapper/dm-19
mke2fs 1.42.11 (09-Jul-2014)
Creating filesystem with 2620928 4k blocks and 655360 inodes
Filesystem UUID: 97f987e9-47b8-47f7-b434-bf3ebbbe826d0
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
```

3. 새 디바이스를 마운트할 폴더를 생성합니다.

```
# mkdir /mnt/ext4
```

4. 디바이스를 마운트합니다.

```
# mount /dev/mapper/eui.00001ffe5a94ff8500a0980000af4444 /mnt/ext4
```

E-시리즈에서 파일 시스템 생성 - Linux RHEL 8, RHEL 9, RHEL 10, SLES 15 및 SLES 16(RoCE 기반 NVMe)

RHEL 8, RHEL 9, RHEL 10, SLES 15 및 SLES 16의 경우 네이티브 NVMe 장치에 파일 시스템을 생성하고 해당 파일 시스템을 마운트합니다.

단계

1. 를 실행합니다 `multipath -ll` NVMe 장치 목록을 가져오는 명령입니다.

```
# multipath -ll
```

이 명령의 결과는 연결된 장치를 찾는 데 사용할 수 있습니다 `/dev/disk/by-id/nvme-eui.[uuid#]` 위치. 아래 예는입니다 `/dev/disc/by-id/nvme-eui.000082dd5c05d39300a0980000a52225`.

```
eui.000082dd5c05d39300a0980000a52225 [nvme]:nvme0n6 NVMe,NetApp E-
Series,08520000
size=4194304 features='n/a' hwhandler='ANA' wp=rw
|+- policy='n/a' prio=50 status=optimized
|  `-- 0:0:1 nvme0c0n1 0:0 n/a optimized      live
|+- policy='n/a' prio=50 status=optimized
|  `-- 0:1:1 nvme0c1n1 0:0 n/a optimized      live
|+- policy='n/a' prio=10 status=non-optimized
|  `-- 0:2:1 nvme0c2n1 0:0 n/a non-optimized live
`+- policy='n/a' prio=10 status=non-optimized
   `-- 0:3:1 nvme0c3n1 0:0 n/a non-optimized live
```

2. 위치를 사용하여 원하는 NVMe 장치의 파티션에 파일 시스템을 생성합니다 /dev/disk/by-id/nvme-eui.[id#].

파일 시스템을 생성하는 방법은 선택한 파일 시스템에 따라 다릅니다. 이 예에서는 ext4 파일 시스템을 만드는 방법을 보여 줍니다.

```
# mkfs.ext4 /dev/disk/by-id/nvme-eui.000082dd5c05d39300a0980000a52225
mke2fs 1.42.11 (22-Oct-2019)
Creating filesystem with 2620928 4k blocks and 655360 inodes
Filesystem UUID: 97f987e9-47b8-47f7-b434-bf3ebbe826d0
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
```

3. 새 디바이스를 마운트할 폴더를 생성합니다.

```
# mkdir /mnt/ext4
```

4. 디바이스를 마운트합니다.

```
# mount /dev/disk/by-id/nvme-eui.000082dd5c05d39300a0980000a52225
/mnt/ext4
```

E-Series-Linux(NVMe over RoCE)에서 호스트에서 스토리지 액세스 확인

네임스페이스를 사용하기 전에 호스트에서 네임스페이스에 데이터를 쓰고 다시 읽을 수 있는지 확인합니다.

시작하기 전에

다음 사항을 확인하십시오.

- 파일 시스템으로 포맷된 초기화된 네임스페이스입니다.

단계

1. 호스트에서 하나 이상의 파일을 디스크의 마운트 지점으로 복사합니다.
2. 파일을 원래 디스크의 다른 폴더로 다시 복사합니다.
3. "IFF" 명령을 실행하여 복사된 파일을 원본과 비교합니다.

작업을 마친 후

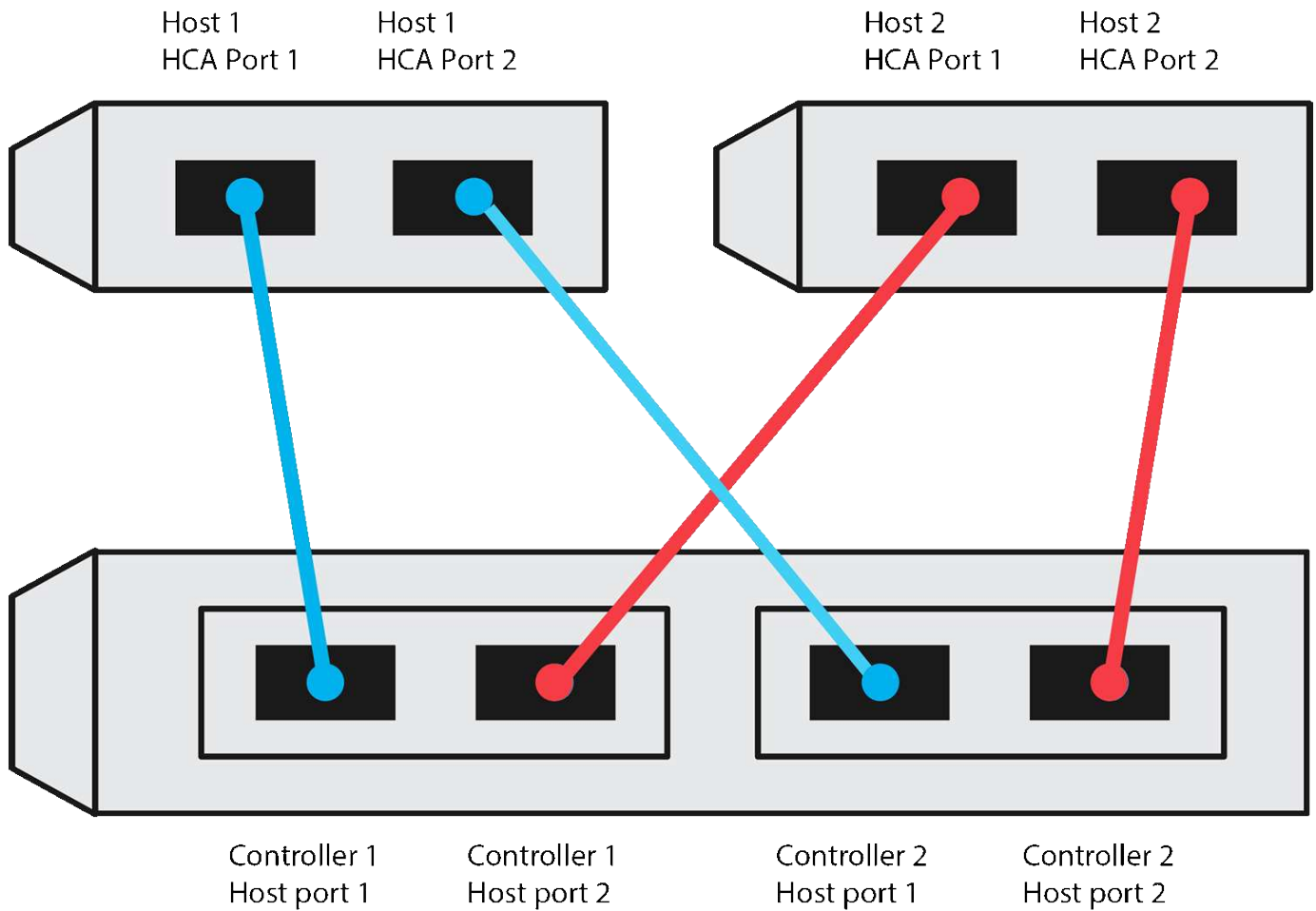
복사한 파일 및 폴더를 제거합니다.

E-Series-Linux에 NVMe over RoCE 구성을 기록합니다

이 페이지의 PDF를 생성하여 인쇄한 다음, 다음 워크시트를 사용하여 NVMe over RoCE 스토리지 구성 정보를 기록할 수 있습니다. 프로비저닝 작업을 수행하려면 이 정보가 필요합니다.

직접 연결 토폴로지

직접 연결 토폴로지에서는 하나 이상의 호스트가 서브시스템에 직접 연결됩니다. SANtricity OS 11.50 릴리스에서는 아래와 같이 각 호스트에서 하위 시스템 컨트롤러로의 단일 연결을 지원합니다. 이 구성에서는 각 호스트의 HCA(호스트 채널 어댑터) 포트 1개가 연결된 E-Series 컨트롤러 포트와 동일한 서브넷에 있어야 하지만 다른 HCA 포트와 다른 서브넷에 있어야 합니다.

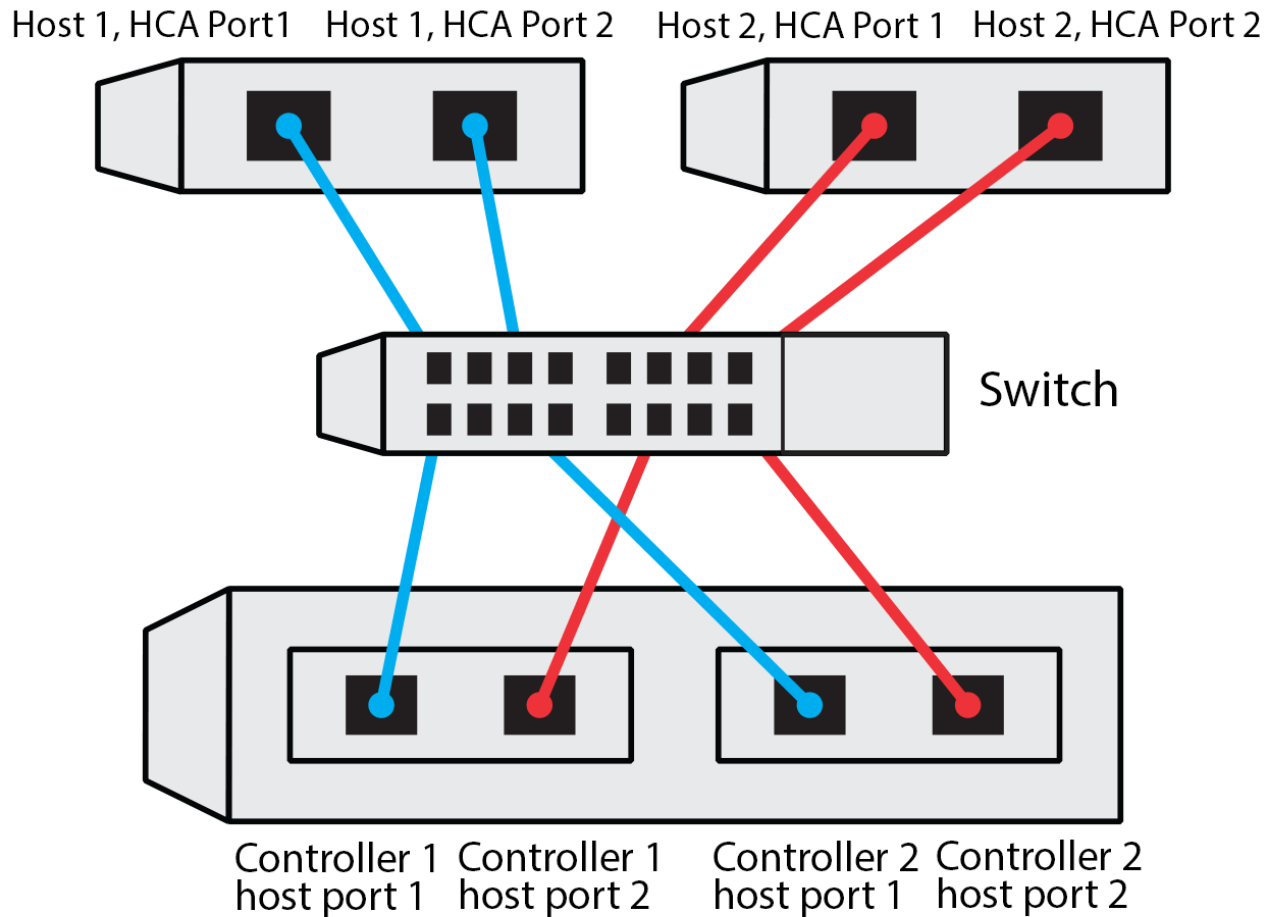


요구 사항을 충족하는 구성의 예는 다음과 같이 4개의 네트워크 서브넷으로 구성됩니다.

- 서브넷 1: 호스트 1 HCA 포트 1 및 컨트롤러 1 호스트 포트 1
- 서브넷 2: 호스트 1 HCA 포트 2 및 컨트롤러 2 호스트 포트 1
- 서브넷 3: 호스트 2 HCA 포트 1 및 컨트롤러 1 호스트 포트 2
- 서브넷 4: 호스트 2 HCA 포트 2 및 컨트롤러 2 호스트 포트 2

스위치 연결 토폴로지

Fabric 토폴로지에서는 하나 이상의 스위치가 사용됩니다. 을 참조하십시오 ["NetApp 상호 운용성 매트릭스 툴"](#) 지원되는 스위치 목록을 참조하십시오.



호스트 식별자입니다

각 호스트에서 이니시에이터 NQN을 찾아 문서화합니다.

호스트 포트 연결	소프트웨어 이니시에이터 NQN
호스트(이니시에이터) 1	
호스트(이니시에이터) 2	

타겟 **NQN**

스토리지 배열의 대상 NQN을 기록합니다.

스토리지 이름입니다	타겟 NQN
어레이 컨트롤러(타겟)	

대상 NQN

스토리지 포트에서 사용할 NQN을 기록합니다.

어레이 컨트롤러(타겟) 포트 연결	NQN
컨트롤러 A, 포트 1	
컨트롤러 B, 포트 1	
컨트롤러 A, 포트 2	
컨트롤러 B, 포트 2	

호스트 이름 매핑 중



매핑 호스트 이름은 워크플로우 중에 생성됩니다.

호스트 이름 매핑 중
호스트 OS 유형입니다

저작권 정보

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

NetApp이 저작권을 가진 자료에 있는 소프트웨어에는 아래의 라이선스와 고지사항이 적용됩니다.

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

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

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

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

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

상표 정보

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