



HA 클러스터 구성 요소를 업데이트합니다

BeeGFS on NetApp with E-Series Storage

NetApp
January 27, 2026

목차

HA 클러스터 구성 요소를 업데이트합니다	1
BeeGFS 서비스 업그레이드	1
개요	1
테스트된 업그레이드 경로	1
BeeGFS 업그레이드 단계	2
버전 업그레이드 참고 사항	3
BeeGFS v8로 업그레이드	4
개요	4
BeeGFS v8의 주요 변경 사항	4
업그레이드를 위해 BeeGFS 클러스터를 준비하세요	4
BeeGFS 패키지 업그레이드	5
관리 데이터베이스 업그레이드	6
라이선싱 구성	7
TLS 암호화 구성	8
업데이트 관리 서비스 구성	8
BeeGFS 모니터 스크립트 업데이트	9
클러스터를 다시 온라인 상태로 전환	12
BeeGFS 클라이언트 업그레이드	13
업그레이드 확인	14
HA 클러스터의 페이스 메이커 및 Corosync 패키지를 업그레이드합니다	14
개요	14
업그레이드 접근 방식	14
파일 노드 어댑터 펌웨어를 업데이트합니다	17
개요	17
업그레이드 고려 사항	17
펌웨어 업데이트 준비	18
롤링 업데이트 접근 방식	18
2노드 클러스터 업데이트 접근 방식	20
E-Series 스토리지 시스템을 업그레이드합니다	21
개요	22
블록 노드 업그레이드 단계	22

HA 클러스터 구성 요소를 업데이트합니다

BeeGFS 서비스 업그레이드

Ansible을 사용하여 HA 클러스터에서 실행 중인 BeeGFS 버전을 업데이트하세요.

개요

BeeGFS는 major.minor.patch 버전 관리 체계를 따릅니다. BeeGFS HA Ansible 역할은 지원되는 각 major.minor 버전(예: beegfs_ha_7_2 및 beegfs_ha_7_3)에 대해 제공됩니다. 각 HA 역할은 Ansible 컬렉션 릴리즈 시점에 제공되는 최신 BeeGFS 패치 버전에 고정되어 있습니다.

BeeGFS의 주요 버전, 마이너 버전 및 패치 버전 간 이동을 포함한 모든 BeeGFS 업그레이드에는 Ansible을 사용해야 합니다. BeeGFS를 업데이트하려면 먼저 BeeGFS Ansible 컬렉션을 업데이트해야 하며, 이 과정에서 배포/관리 자동화 및 기본 HA 클러스터에 대한 최신 수정 사항과 개선 사항도 함께 가져와집니다. 컬렉션을 최신 버전으로 업데이트한 후에도 `-e "beegfs_ha_force_upgrade=true"`가 설정된 상태에서 `ansible-playbook`을 실행하기 전까지는 BeeGFS가 업그레이드되지 않습니다. 각 업그레이드에 대한 자세한 내용은 현재 버전의 ["BeeGFS 업그레이드 설명서"](#)를 참조하십시오.



BeeGFS v8로 업그레이드하는 경우 ["BeeGFS v8로 업그레이드"](#) 절차를 참조하십시오.

테스트된 업그레이드 경로

다음 업그레이드 경로는 테스트 및 검증을 완료했습니다.

원본 버전	버전 업그레이드	멀티 레일	세부 정보
7.2.6	7.3.2	예	v3.0.1에서 v3.1.0으로 Beegfs 컬렉션을 업그레이드하는 중, 멀티레일이 추가되었습니다
7.2.6	7.2.8	아니요	v3.0.1에서 v3.1.0으로 Beegfs 컬렉션 업그레이드
7.2.8	7.3.1에서 포함	예	beegfs 컬렉션 v3.1.0을 사용하여 업그레이드하십시오. 멀티레일이 추가되었습니다
7.3.1에서 포함	7.3.2	예	Beegfs 컬렉션 v3.1.0을 사용하여 업그레이드합니다
7.3.2	7.4.1	예	Beegfs 컬렉션 v3.2.0을 사용하여 업그레이드합니다
7.4.1	7.4.2	예	Beegfs 컬렉션 v3.2.0을 사용하여 업그레이드합니다
7.4.2	7.4.6	예	Beegfs 컬렉션 v3.2.0을 사용하여 업그레이드합니다
7.4.6	8.0	예	"BeeGFS v8로 업그레이드" 절차의 지침을 사용하여 업그레이드하십시오.
7.4.6	8.1	예	"BeeGFS v8로 업그레이드" 절차의 지침을 사용하여 업그레이드하십시오.
7.4.6	8.2	예	"BeeGFS v8로 업그레이드" 절차의 지침을 사용하여 업그레이드하십시오.

BeeGFS 업그레이드 단계

다음 섹션에서는 BeeGFS Ansible 컬렉션 및 BeeGFS 자체를 업데이트하는 단계를 제공합니다. BeeGFS 주 버전 또는 부 버전 업데이트에 대한 추가 단계에 특히 주의하십시오.

1단계: BeeGFS 컬렉션 업그레이드

에 액세스하여 컬렉션 업그레이드용 ["Ansible 갤러리"](#)에서 다음 명령을 실행합니다.

```
ansible-galaxy collection install netapp_eseries.beegfs --upgrade
```

오프라인 컬렉션 업그레이드의 경우 에서 컬렉션을 다운로드하십시오 ["Ansible 갤러리"](#) 원하는 을 클릭합니다 Install Version` 그리고 나서 Download tarball. 타볼을 Ansible 제어 노드로 전송하고 다음 명령을 실행합니다.

```
ansible-galaxy collection install netapp_eseries-beegfs-<VERSION>.tar.gz  
--upgrade
```

을 참조하십시오 ["컬렉션 설치 중"](#) 를 참조하십시오.

2단계: Ansible 인벤토리 업데이트

클러스터의 Ansible 인벤토리 파일에 필요한 또는 원하는 업데이트를 수행하십시오. 특정 업그레이드 요구 사항에 대한 자세한 내용은 아래 [버전 업그레이드 참고 사항](#) 섹션을 참조하십시오. BeeGFS HA 인벤토리 구성에 대한 일반 정보는 ["Ansible 인벤토리 개요"](#) 섹션을 참조하십시오.

3단계: Ansible 플레이북 업데이트(주 버전 또는 부 버전을 업데이트하는 경우에만)

주 버전 또는 부 버전 간에 이동하는 경우 playbook.yml 클러스터를 배포하고 유지 관리하는 데 사용되는 파일에서 원하는 버전을 반영하도록 역할의 이름을 beegfs_ha_<VERSION> 업데이트합니다. 예를 들어 BeeGFS 7.4를 배포하려는 경우 다음과 같은 이점이 `beegfs_ha_7_4` 있습니다.

```
- hosts: all  
  gather_facts: false  
  any_errors_fatal: true  
  collections:  
    - netapp_eseries.beegfs  
  tasks:  
    - name: Ensure BeeGFS HA cluster is setup.  
      ansible.builtin.import_role: # import_role is required for tag  
        availability.  
        name: beegfs_ha_7_4
```

이 플레이북 파일의 내용에 대한 자세한 ["BeeGFS HA 클러스터를 구축합니다"](#)내용은 섹션을 참조하십시오.

4단계: BeeGFS 업그레이드를 실행합니다

BeeGFS 업데이트 적용하기:

```
ansible-playbook -i inventory.yml beegfs_ha_playbook.yml -e  
"beegfs_ha_force_upgrade=true" --tags beegfs_ha
```

BeeGFS HA 역할에서 다루는 비하인드 스토리:

- 각 BeeGFS 서비스가 기본 노드에 위치하도록 클러스터가 최적의 상태인지 확인합니다.
- 클러스터를 유지보수 모드로 전환합니다.
- HA 클러스터 구성 요소를 업데이트합니다(필요한 경우).
- 다음과 같이 각 파일 노드를 한 번에 하나씩 업그레이드합니다.
 - 대기 노드에 배치하고 서비스를 보조 노드로 페일오버합니다.
 - BeeGFS 패키지를 업그레이드합니다.
 - 서비스 대체.
- 클러스터를 유지보수 모드 외부로 이동합니다.

버전 업그레이드 참고 사항

BeeGFS 버전 7.2.6 또는 7.3.0에서 업그레이드

연결 기반 인증에 대한 변경 사항

BeeGFS 버전 7.3.2 이상에서는 연결 기반 인증이 구성되어 있어야 합니다. 다음 중 하나라도 구성되지 않으면 서비스가 시작되지 않습니다.

- `connAuthFile` 또는 `connAuthFile` 를 지정합니다.
- 서비스의 구성 파일에서 `connDisableAuthentication=true` 설정.

보안을 위해 연결 기반 인증을 활성화하는 것을 적극 권장합니다. 자세한 내용은 "[BeeGFS 연결 기반 인증](#)"을 참조하십시오.

``beegfs_ha*`` 역할은 인증 파일을 자동으로 생성하여 다음 위치로 배포합니다.

- 클러스터의 모든 파일 노드
- ``<playbook_directory>/files/beegfs/<beegfs_mgmt_ip_address>_connAuthFile``의 Ansible 제어 노드

``beegfs_client`` 역할은 파일이 존재할 경우 자동으로 감지하여 클라이언트에 적용합니다.



beegfs_client 역할을 사용하여 클라이언트를 구성하지 않은 경우 각 클라이언트에 인증 파일을 수동으로 배포하고 beegfs-client.conf 파일에서 connAuthFile 설정을 구성해야 합니다. 연결 기반 인증이 없는 BeeGFS 버전에서 업그레이드하는 경우 `group_vars/ha_cluster.yml`에서 `beegfs_ha_conn_auth_enabled: false`를 설정하여 업그레이드 중에 연결 기반 인증을 비활성화하지 않으면 클라이언트가 액세스 권한을 잃게 됩니다(권장하지 않음).

자세한 내용 및 다른 구성 옵션은 "[일반 파일 노드 구성을 지정합니다](#)" 섹션의 연결 인증 구성 단계를 참조하십시오.

BeeGFS v8로 업그레이드

다음 단계를 따라 BeeGFS HA 클러스터를 버전 7.4.6에서 BeeGFS v8로 업그레이드하십시오.

개요

BeeGFS v8은 BeeGFS v7에서 업그레이드하기 전에 추가 설정이 필요한 몇 가지 중요한 변경 사항을 도입했습니다. 이 문서에서는 BeeGFS v8의 새로운 요구 사항에 맞게 클러스터를 준비하고 BeeGFS v8로 업그레이드하는 과정을 안내합니다.



BeeGFS v8로 업그레이드하기 전에 시스템에 BeeGFS 7.4.6 이상이 설치되어 있는지 확인하십시오. BeeGFS 7.4.6 이전 버전을 실행 중인 클러스터는 이 BeeGFS v8 업그레이드 절차를 진행하기 전에 먼저 "[버전 7.4.6으로 업그레이드](#)"해야 합니다.

BeeGFS v8의 주요 변경 사항

BeeGFS v8에서는 다음과 같은 주요 변경 사항이 도입되었습니다.

- **라이선스 시행:** BeeGFS v8은 스토리지 풀, 원격 스토리지 대상, BeeOND 등과 같은 프리미엄 기능을 사용하기 위해 라이선스가 필요합니다. 업그레이드하기 전에 BeeGFS 클러스터에 대한 유효한 라이선스를 확보하십시오. 필요한 경우 "[BeeGFS 라이선스 포털](#)"에서 임시 BeeGFS v8 평가판 라이선스를 받을 수 있습니다.
- **관리 서비스 데이터베이스 마이그레이션:** BeeGFS v8의 새로운 TOML 기반 형식으로 구성을 활성화하려면 BeeGFS v7 관리 서비스 데이터베이스를 업데이트된 BeeGFS v8 형식으로 수동으로 마이그레이션해야 합니다.
- **TLS 암호화:** BeeGFS v8은 서비스 간의 안전한 통신을 위해 TLS를 도입했습니다. 업그레이드 과정에서 BeeGFS 관리 서비스와 beegfs 명령줄 유틸리티에 사용할 TLS 인증서를 생성하고 배포해야 합니다.

BeeGFS 8의 자세한 내용 및 추가 변경 사항은 "[BeeGFS v8.0.0 업그레이드 가이드](#)"를 참조하십시오.



BeeGFS v8로 업그레이드하려면 클러스터 가동 중지 시간이 필요합니다. 또한 BeeGFS v7 클라이언트는 BeeGFS v8 클러스터에 연결할 수 없습니다. 운영에 미치는 영향을 최소화하려면 클러스터와 클라이언트 간의 업그레이드 시기를 신중하게 조율하십시오.

업그레이드를 위해 BeeGFS 클러스터를 준비하세요

업그레이드를 시작하기 전에 원활한 전환과 다운타임 최소화를 위해 환경을 신중하게 준비하십시오.

1. 클러스터가 정상 상태인지, 모든 BeeGFS 서비스가 기본 노드에서 실행 중인지 확인하십시오. BeeGFS 서비스가 실행 중인 파일 노드에서 모든 Pacemaker 리소스가 기본 노드에서 실행 중인지 확인하십시오.

```
pcs status
```

2. 클러스터 구성을 기록하고 백업합니다.

- 클러스터 구성 백업에 대한 지침은 "[BeeGFS 백업 설명서](#)"을 참조하십시오.
- 기존 관리 데이터 디렉토리를 백업합니다.

```
cp -r /mnt/mgmt_tgt_mgmt01/data  
/mnt/mgmt_tgt_mgmt01/data_beegfs_v7_backup_$(date +%Y%m%d)
```

- beegfs 클라이언트에서 다음 명령을 실행하고 출력을 참조용으로 저장합니다.

```
beegfs-ctl --getentryinfo --verbose /path/to/beegfs/mountpoint
```

- 미러링을 사용하는 경우 자세한 상태 정보를 수집하십시오.

```
beegfs-ctl --listtargets --longnodes --state --spaceinfo  
--mirrorgroups --nodetype=meta  
beegfs-ctl --listtargets --longnodes --state --spaceinfo  
--mirrorgroups --nodetype=storage
```

3. 클라이언트의 다운타임을 준비하고 'beegfs-client' 서비스를 중지하십시오. 각 클라이언트에 대해 다음을 실행하십시오.

```
systemctl stop beegfs-client
```

4. 각 Pacemaker 클러스터에 대해 STONITH를 비활성화하십시오. 이렇게 하면 불필요한 노드 재부팅을 트리거하지 않고 업그레이드 후 클러스터의 무결성을 검증할 수 있습니다.

```
pcs property set stonith-enabled=false
```

5. BeeGFS 네임스페이스의 모든 Pacemaker 클러스터에 대해 PCS를 사용하여 클러스터를 중지합니다.

```
pcs cluster stop --all
```

BeeGFS 패키지 업그레이드

클러스터의 모든 파일 노드에 사용 중인 Linux 배포판에 맞는 BeeGFS v8 패키지 저장소를 추가하십시오. 공식 BeeGFS 저장소 사용 방법은 "[BeeGFS 다운로드 페이지](#)"에서 확인할 수 있습니다. 그렇지 않은 경우 로컬 beegfs 미러

저장소를 적절히 구성하십시오.

다음 단계는 RHEL 9 파일 노드에서 공식 BeeGFS 8.2 리포지토리를 사용하는 방법을 안내합니다. 클러스터의 모든 파일 노드에서 다음 단계를 수행하십시오.

1. BeeGFS GPG 키를 가져옵니다:

```
rpm --import https://www.beegfs.io/release/beegfs_8.2/gpg/GPG-KEY-beegfs
```

2. BeeGFS 리포지토리를 가져옵니다.

```
curl -L -o /etc/yum.repos.d/beegfs-rhel9.repo  
https://www.beegfs.io/release/beegfs_8.2/dists/beegfs-rhel9.repo
```



새로운 BeeGFS v8 리포지토리와의 충돌을 방지하려면 이전에 구성된 BeeGFS 리포지토리를 모두 제거하십시오.

3. 패키지 관리자 캐시를 정리하세요:

```
dnf clean all
```

4. 모든 파일 노드에서 BeeGFS 패키지를 BeeGFS 8.2로 업데이트하십시오.

```
dnf update beegfs-mgmt beegfs-storage beegfs-meta libbeegfs-ib
```



표준 클러스터에서 `beegfs-mgmt` 패키지는 처음 두 개의 파일 노드에서만 업데이트됩니다.

관리 데이터베이스 업그레이드

BeeGFS 관리 서비스가 실행 중인 파일 노드 중 하나에서 다음 단계를 수행하여 관리 데이터베이스를 BeeGFS v7에서 v8로 마이그레이션합니다.

1. 모든 NVMe 디바이스를 나열하고 관리 타겟을 기준으로 필터링합니다.

```
nvme netapp smdevices | grep mgmt_tgt
```

a. 출력에서 장치 경로를 확인하십시오.

b. 기존 관리 대상 마운트 지점에 관리 대상 장치를 마운트합니다(``/dev/nvmeXnY``를 장치 경로로 대체):

```
mount /dev/nvmeXnY /mnt/mgmt_tgt_mgmt01/
```

2. 다음을 실행하여 BeeGFS 7 관리 데이터를 새 데이터베이스 형식으로 가져옵니다.

```
/opt/beegfs/sbin/beegfs-mgmt --import-from  
-v7=/mnt/mgmt_tgt_mgmt01/data/
```

예상 출력:

```
Created new database version 3 at "/var/lib/beegfs/mgmt.sqlite".  
Successfully imported v7 management data from  
"/mnt/mgmt_tgt_mgmt01/data/".
```



BeeGFS v8의 강화된 유효성 검사 요구 사항으로 인해 자동 가져오기가 모든 경우에 성공하지 못할 수 있습니다. 예를 들어, 대상이 존재하지 않는 스토리지 풀에 할당된 경우 가져오기가 실패합니다. 마이그레이션이 실패하면 업그레이드를 진행하지 마십시오. 데이터베이스 마이그레이션 문제 해결을 위해 NetApp 지원팀에 문의하십시오. 임시 해결책으로, 문제가 해결될 때까지 BeeGFS v8 패키지를 다운그레이드하고 BeeGFS v7을 계속 실행할 수 있습니다.

3. 생성된 SQLite 파일을 관리 서비스 마운트로 이동합니다.

```
mv /var/lib/beegfs/mgmt.sqlite /mnt/mgmt_tgt_mgmt01/data/
```

4. 생성된 `beegfs-mgmt.toml`을 관리 서비스 마운트로 이동합니다.

```
mv /etc/beegfs/beegfs-mgmt.toml /mnt/mgmt_tgt_mgmt01/mgmt_config/
```

`beegfs-mgmt.toml` 구성 파일 준비는 다음 섹션의 라이선싱 및 TLS 구성 단계를 완료한 후에 수행됩니다.

라이선싱 구성

1. beegfs 관리 서비스를 실행하는 모든 노드에 beegfs 라이선스 패키지를 설치하십시오. 일반적으로 클러스터의 첫 번째 두 노드입니다.

```
dnf install libbeegfs-license
```

2. BeeGFS v8 라이선스 파일을 관리 노드에 다운로드하여 다음 위치에 배치하십시오.

```
/etc/beegfs/license.pem
```

TLS 암호화 구성

BeeGFS v8은 관리 서비스와 클라이언트 간의 안전한 통신을 위해 TLS 암호화를 필요로 합니다. 관리 서비스와 클라이언트 서비스 간의 네트워크 통신에 TLS 암호화를 구성하는 방법은 세 가지가 있습니다. 권장되는 가장 안전한 방법은 신뢰할 수 있는 인증 기관(CA)에서 서명한 인증서를 사용하는 것입니다. 또는 자체 로컬 CA를 생성하여 BeeGFS 클러스터용 인증서를 서명할 수도 있습니다. 암호화가 필요하지 않은 환경이나 문제 해결을 위한 경우에는 TLS를 완전히 비활성화할 수 있지만, 이 경우 민감한 정보가 네트워크에 노출되므로 권장하지 않습니다.

진행하기 전에 "[BeeGFS 8용 TLS 암호화 구성](#)" 가이드의 지침에 따라 환경에 맞는 TLS 암호화를 설정하십시오.

업데이트 관리 서비스 구성

BeeGFS v7 구성 파일의 설정을 `/mnt/mgmt_tgt_mgmt01/mgmt_config/beegfs-mgmtd.toml` 파일로 수동으로 전송하여 BeeGFS v8 관리 서비스 구성 파일을 준비합니다.

1. 관리 대상이 마운트된 관리 노드에서 BeeGFS 7용 관리 서비스 파일

`/mnt/mgmt_tgt_mgmt01/mgmt_config/beegfs-mgmtd.conf`을 참조한 다음 모든 설정을 `/mnt/mgmt_tgt_mgmt01/mgmt_config/beegfs-mgmtd.toml` 파일로 전송합니다. 기본 설정의 경우 `'beegfs-mgmtd.toml'`은 다음과 같을 수 있습니다.

```
beemsg-port = 8008
grpc-port = 8010
log-level = "info"
node-offline-timeout = "900s"
quota-enable = false
auth-disable = false
auth-file = "/etc/beegfs/<mgmt_service_ip>_connAuthFile"
db-file = "/mnt/mgmt_tgt_mgmt01/data/mgmtd.sqlite"
license-disable = false
license-cert-file = "/etc/beegfs/license.pem"
tls-disable = false
tls-cert-file = "/etc/beegfs/mgmtd_tls_cert.pem"
tls-key-file = "/etc/beegfs/mgmtd_tls_key.pem"
interfaces = ['i1b:mgmt_1', 'i2b:mgmt_2']
```

필요에 따라 모든 경로를 조정하여 사용자의 환경 및 TLS 구성과 일치시키십시오.

2. 관리 서비스를 실행하는 각 파일 노드에서 systemd 서비스 파일을 수정하여 새 구성 파일 위치를 가리키도록 합니다.

```
sudo sed -i 's|ExecStart=.*|ExecStart=nice -n -3
/opt/beegfs/sbin/beegfs-mgmtd --config-file
/mnt/mgmt_tgt_mgmt01/mgmt_config/beegfs-mgmtd.toml|'
/etc/systemd/system/beegfs-mgmtd.service
```

a. systemd를 다시 로드합니다.

```
systemctl daemon-reload
```

3. 관리 서비스를 실행하는 각 파일 노드에 대해 관리 서비스의 gRPC 통신을 위해 포트 8010을 엽니다.
 - a. beegfs 영역에 포트 8010/tcp를 추가합니다.

```
sudo firewall-cmd --zone=beegfs --permanent --add-port=8010/tcp
```

- b. 변경 사항을 적용하려면 방화벽을 다시 로드하십시오.

```
sudo firewall-cmd --reload
```

BeeGFS 모니터 스크립트 업데이트

Pacemaker beegfs-monitor OCF 스크립트는 새로운 TOML 구성 형식과 systemd 서비스 관리를 지원하도록 업데이트해야 합니다. 클러스터의 한 노드에서 스크립트를 업데이트한 다음, 업데이트된 스크립트를 다른 모든 노드로 복사하십시오.

1. 현재 스크립트의 백업을 생성합니다:

```
cp /usr/lib/ocf/resource.d/eseries/beegfs-monitor  
/usr/lib/ocf/resource.d/eseries/beegfs-monitor.bak.$(date +%F)
```

2. 관리 구성 파일 경로를 .conf에서 .toml(으)로 업데이트하십시오:

```
sed -i 's|mgmt_config/beegfs-mgmtd\.conf|mgmt_config/beegfs-mgmtd.toml|'  
/usr/lib/ocf/resource.d/eseries/beegfs-monitor
```

또는 스크립트에서 다음 블록을 수동으로 찾습니다.

```
case $type in  
  management)  
    conf_path="${configuration_mount}/mgmt_config/beegfs-mgmtd.conf"  
    ;;
```

그리고 다음으로 바꾸세요:

```
case $type in
management)
    conf_path="${configuration_mount}/mgmt_config/beegfs-mgmt.d.toml"
    ;;
```

3. `get_interfaces()` 및 `get_subnet_ips()` 함수를 업데이트하여 TOML 구성을 지원합니다.
 - a. 텍스트 편집기에서 스크립트를 엽니다.

```
vi /usr/lib/ocf/resource.d/eseries/beegfs-monitor
```

- b. 다음 두 함수를 찾으세요: `get_interfaces()` 및 `get_subnet_ips()`.
- c. `get_interfaces()`에서 시작하여 `get_subnet_ips()`의 끝까지 두 함수 전체를 삭제하세요.
- d. 다음 업데이트된 함수를 복사하여 해당 위치에 붙여넣으세요.

```

# Return network communication interface name(s) from the BeeGFS
resource's connInterfaceFile
get_interfaces() {
    # Determine BeeGFS service network IP interfaces.
    if [ "$type" = "management" ]; then
        interfaces_line=$(grep "^interfaces =" "$conf_path")
        interfaces_list=$(echo "$interfaces_line" | sed "s/.*= \[\\(.*)
\\)\]/\1/")
        interfaces=$(echo "$interfaces_list" | tr -d '"' | tr -d " " | tr
', ' '\n')

        for entry in $interfaces; do
            echo "$entry" | cut -d ':' -f 1
        done
    else
        connInterfacesFile_path=$(grep "^connInterfacesFile" "$conf_path"
| tr -d "[:space:]" | cut -f 2 -d "=")

        if [ -f "$connInterfacesFile_path" ]; then
            while read -r entry; do
                echo "$entry" | cut -f 1 -d ':'
            done < "$connInterfacesFile_path"
        fi
    fi
}

# Return list containing all the BeeGFS resource's usable IP
addresses. *Note that these are filtered by the connNetFilterFile
entries.
get_subnet_ips() {
    # Determine all possible BeeGFS service network IP addresses.
    if [ "$type" != "management" ]; then
        connNetFilterFile_path=$(grep "^connNetFilterFile" "$conf_path" |
tr -d "[:space:]" | cut -f 2 -d "=")

        filter_ips=""
        if [ -n "$connNetFilterFile_path" ] && [ -e
$connNetFilterFile_path ]; then
            while read -r filter; do
                filter_ips="$filter_ips $(get_ipv4_subnet_addresses $filter)"
            done < $connNetFilterFile_path
        fi

        echo "$filter_ips"
    fi
}

```

- e. 텍스트 편집기를 저장하고 종료합니다.
- f. 진행하기 전에 다음 명령을 실행하여 스크립트의 구문 오류를 확인하십시오. 출력이 없으면 스크립트의 구문이 올바릅니다.

```
bash -n /usr/lib/ocf/resource.d/eseries/beegfs-monitor
```

4. 일관성을 유지하기 위해 업데이트된 beegfs-monitor OCF 스크립트를 클러스터의 다른 모든 노드에 복사하십시오.

```
scp /usr/lib/ocf/resource.d/eseries/beegfs-monitor
user@node:/usr/lib/ocf/resource.d/eseries/beegfs-monitor
```

클러스터를 다시 온라인 상태로 전환

1. 이전의 모든 업그레이드 단계를 완료한 후 모든 노드에서 BeeGFS 서비스를 시작하여 클러스터를 다시 온라인 상태로 전환하십시오.

```
pcs cluster start --all
```

2. beegfs-mgmtd 서비스가 성공적으로 시작되었는지 확인하십시오.

```
journalctl -xeu beegfs-mgmtd
```

예상 출력에는 다음과 같은 줄이 포함됩니다:

```
Started Cluster Controlled beegfs-mgmtd.
Loaded config file from "/mnt/mgmt_tgt_mgmt01/mgmt_config/beegfs-
mgmt.d.toml"
Successfully initialized certificate verification library.
Successfully loaded license certificate: TMP-113489268
Opened database at "/mnt/mgmt_tgt_mgmt01/data/mgmt.d.sqlite"
Listening for BeeGFS connections on [::]:8008
Serving gRPC requests on [::]:8010
```



저널 로그에 오류가 나타나면 관리 구성 파일 경로를 검토하고 BeeGFS 7 구성 파일에서 모든 값이 올바르게 전송되었는지 확인하십시오.

3. `pcs status`를 실행하고 클러스터가 정상 상태이며 서비스가 기본 설정 노드에서 시작되었는지 확인합니다.
4. 클러스터가 정상 상태임을 확인한 후 STONITH를 다시 활성화하십시오.

```
pcs property set stonith-enabled=true
```

5. 클러스터의 BeeGFS 클라이언트를 업그레이드하고 BeeGFS 클러스터의 상태를 확인하려면 다음 섹션으로 이동하십시오.

BeeGFS 클라이언트 업그레이드

클러스터를 BeeGFS v8로 성공적으로 업그레이드한 후에는 모든 BeeGFS 클라이언트도 업그레이드해야 합니다.

다음 단계는 Ubuntu 기반 시스템에서 BeeGFS 클라이언트를 업그레이드하는 과정을 설명합니다.

1. 아직 하지 않았다면 BeeGFS 클라이언트 서비스를 중지하십시오.

```
systemctl stop beegfs-client
```

2. 사용하는 Linux 배포판에 맞는 BeeGFS v8 패키지 저장소를 추가하세요. 공식 BeeGFS 저장소 사용 방법은 "[^BeeGFS 다운로드 페이지](#)"에서 확인할 수 있습니다. 그렇지 않은 경우 로컬 BeeGFS 미러 저장소를 적절히 구성하세요.

다음 단계에서는 Ubuntu 기반 시스템에서 공식 BeeGFS 8.2 리포지토리를 사용합니다.

3. BeeGFS GPG 키를 가져옵니다:

```
wget https://www.beegfs.io/release/beegfs_8.2/gpg/GPG-KEY-beegfs -O /etc/apt/trusted.gpg.d/beegfs.asc
```

4. 리포지토리 파일을 다운로드하세요:

```
wget https://www.beegfs.io/release/beegfs_8.2/dists/beegfs-noble.list -O /etc/apt/sources.list.d/beegfs.list
```



새로운 BeeGFS v8 리포지토리와의 충돌을 방지하려면 이전에 구성된 BeeGFS 리포지토리를 모두 제거하십시오.

5. BeeGFS 클라이언트 패키지를 업그레이드합니다.

```
apt-get update  
apt-get install --only-upgrade beegfs-client
```

6. 클라이언트에 TLS를 구성하십시오. BeeGFS CLI를 사용하려면 TLS가 필요합니다. "[BeeGFS 8용 TLS 암호화 구성](#)" 절차를 참조하여 클라이언트에서 TLS를 구성하십시오.
7. BeeGFS 클라이언트 서비스를 시작합니다.

```
systemctl start beegfs-client
```

업그레이드 확인

BeeGFS v8로 업그레이드를 완료한 후 다음 명령을 실행하여 업그레이드가 성공했는지 확인하십시오.

1. 루트 inode가 이전과 동일한 메타데이터 노드에 의해 소유되는지 확인하십시오. 관리 서비스에서 `import-from-v7` 기능을 사용한 경우 이 작업은 자동으로 수행됩니다.

```
beegfs entry info /mnt/beegfs
```

2. 모든 노드와 대상이 온라인 상태이고 양호한 상태인지 확인하십시오.

```
beegfs health check
```



"Available Capacity" 검사에서 대상의 여유 공간이 부족하다는 경고가 표시되면 `beegfs-mgmt.d.toml` 파일에 정의된 "capacity pool" 임계값을 환경에 맞게 조정할 수 있습니다.

HA 클러스터의 페이스 메이커 및 Corosync 패키지를 업그레이드합니다

HA 클러스터의 심장박동기 및 Corosync 패키지를 업그레이드하려면 다음 단계를 수행하십시오.

개요

페이스 메이커와 Corosync를 업그레이드하면 새로운 기능, 보안 패치 및 성능 향상의 이점을 누릴 수 있습니다.

업그레이드 접근 방식

클러스터를 업그레이드하는 데에는 롤링 업그레이드 또는 전체 클러스터 종료라는 두 가지 권장 방법이 있습니다. 각 접근 방식에는 고유한 장점과 단점이 있습니다. 업그레이드 절차는 심장박동기 릴리스 버전에 따라 다를 수 있습니다. ClusterLabs의 "[심장박동기 클러스터 업그레이드](#)" 설명서를 참조하여 사용할 방법을 결정합니다. 업그레이드 방법을 따르기 전에 다음 사항을 확인하십시오.

- NetApp BeeGFS 솔루션 내에서 새로운 페이스 메이커 및 Corosync 패키지가 지원됩니다.
- BeeGFS 파일 시스템 및 Pacemaker 클러스터 구성에 유효한 백업이 있습니다.
- 클러스터가 정상 상태입니다.

롤링 업그레이드

이 방법은 클러스터에서 각 노드를 제거하고 업그레이드한 다음 모든 노드에서 새 버전이 실행될 때까지 클러스터에

다시 도입하는 것입니다. 이렇게 하면 클러스터가 운영 상태로 유지되므로 대규모 HA 클러스터에 이상적이지만, 프로세스 중에 혼합 버전을 실행할 위험이 있습니다. 2노드 클러스터에서 이 접근 방식은 사용하지 않아야 합니다.

1. 각 BeeGFS 서비스가 1차 노드에서 실행되고 있는 상태에서 클러스터가 최적의 상태인지 확인합니다. 자세한 내용은 ["클러스터의 상태를 검사합니다"](#) 참조하십시오.
2. 업그레이드할 노드를 대기 모드로 전환하여 모든 BeeGFS 서비스를 방전하거나 이동합니다.

```
pcs node standby <HOSTNAME>
```

3. 다음을 실행하여 노드의 서비스가 소진되었는지 확인합니다.

```
pcs status
```

대기 상태인 노드에 대해 보고된 서비스가 `started` 없는지 확인합니다.



클러스터 크기에 따라 서비스가 자매 노드로 이동하는 데 몇 초 또는 몇 분이 걸릴 수 있습니다. BeeGFS 서비스가 자매 노드에서 시작되지 않는 경우 ["문제 해결 설명서"](#)를 참조하십시오.

4. 노드에서 클러스터를 종료합니다.

```
pcs cluster stop <HOSTNAME>
```

5. 노드에서 심장박동기, Corosync 및 PCS 패키지를 업그레이드합니다.



패키지 관리자 명령은 운영 체제에 따라 다릅니다. 다음은 RHEL 8 이상을 실행하는 시스템에 대한 명령입니다.

```
dnf update pacemaker-<version>
```

```
dnf update corosync-<version>
```

```
dnf update pcs-<version>
```

6. 노드에서 심장박동기 클러스터 서비스를 시작합니다.

```
pcs cluster start <HOSTNAME>
```

7. 패키지가 업데이트된 경우 pcs 클러스터를 사용하여 노드를 다시 인증합니다.

```
pcs host auth <HOSTNAME>
```

8. 박동조율기 구성이 여전히 도구에 유효한지 `crm_verify` 확인합니다.



클러스터 업그레이드 중에 한 번만 확인하면 됩니다.

```
crm_verify -L -V
```

9. 노드를 대기 모드에서 해제합니다.

```
pcs node unstandby <HOSTNAME>
```

10. 모든 BeeGFS 서비스를 기본 노드로 다시 재배치:

```
pcs resource relocate run
```

11. 모든 노드가 원하는 페이스 메이커, Corosync 및 PCS 버전을 실행할 때까지 클러스터의 각 노드에 대해 이전 단계를 반복합니다.
12. 마지막으로, `pcs status` 클러스터를 실행하고 클러스터 상태가 양호한지 확인하고 가 Current DC 원하는 페이스 메이커 버전을 보고합니다.



'Current DC' 혼합 버전이 보고되면 클러스터의 노드가 이전 페이스 메이커 버전과 함께 실행되고 있으므로 업그레이드해야 합니다. 업그레이드된 노드가 클러스터에 다시 연결할 수 없거나 리소스가 시작되지 않는 경우, 클러스터 로그를 확인하고 알려진 업그레이드 문제에 대해서는 Pacemaker 릴리즈 노트 또는 사용자 가이드를 참조하십시오.

클러스터 종료를 완료합니다

이 접근 방식에서는 모든 클러스터 노드 및 리소스가 종료되고 노드가 업그레이드된 다음 클러스터가 다시 시작됩니다. 이 방법은 페이스 메이커 및 Corosync 버전이 혼합 버전 구성을 지원하지 않는 경우에 필요합니다.

1. 각 BeeGFS 서비스가 1차 노드에서 실행되고 있는 상태에서 클러스터가 최적의 상태인지 확인합니다. 자세한 내용은 ["클러스터의 상태를 검사합니다"](#) 참조하십시오.
2. 모든 노드에서 클러스터 소프트웨어(심장박동기 및 Corosync)를 종료합니다.



클러스터 크기에 따라 전체 클러스터를 중지하는 데 몇 초 또는 몇 분이 걸릴 수 있습니다.

```
pcs cluster stop --all
```

3. 모든 노드에서 클러스터 서비스가 종료되면 요구 사항에 따라 각 노드의 심장박동기, Corosync 및 PCS 패키지를 업그레이드합니다.



패키지 관리자 명령은 운영 체제에 따라 다릅니다. 다음은 RHEL 8 이상을 실행하는 시스템에 대한 명령입니다.

```
dnf update pacemaker-<version>
```

```
dnf update corosync-<version>
```

```
dnf update pcs-<version>
```

4. 모든 노드를 업그레이드한 후 모든 노드에서 클러스터 소프트웨어를 시작합니다.

```
pcs cluster start --all
```

5. 패키지가 업데이트된 경우 pcs 클러스터의 각 노드를 다시 인증합니다.

```
pcs host auth <HOSTNAME>
```

6. 마지막으로, pcs status 클러스터를 실행하고 클러스터의 상태가 양호한지 확인하고 가 Current DC 올바른 심장박동기 버전을 보고합니다.



`Current DC`혼합 버전'이 보고되면 클러스터의 노드가 이전 페이스 메이커 버전과 함께 실행되고 있으므로 업그레이드해야 합니다.

파일 노드 어댑터 펌웨어를 업데이트합니다

다음 단계에 따라 파일 노드의 ConnectX-7 어댑터를 최신 펌웨어로 업데이트합니다.

개요

새로운 MLNX_OFED 드라이버를 지원하거나 새로운 기능을 활성화하거나 버그를 수정하려면 ConnectX-7 어댑터 펌웨어를 업데이트해야 할 수 있습니다. 이 설명서에서는 NVIDIA의 유틸리티를 사용하여 어댑터를 업데이트할 수 있습니다. 이 mlxfwmanager 유틸리티는 사용 편의성과 효율성이 우수합니다.

업그레이드 고려 사항

이 가이드에서는 ConnectX-7 어댑터 펌웨어를 업데이트하는 두 가지 방법, 즉 롤링 업데이트와 2노드 클러스터 업데이트에 대해 설명합니다. 클러스터 크기에 따라 적절한 업데이트 방법을 선택합니다. 펌웨어 업데이트를 수행하기 전에 다음 사항을 확인하십시오.

- 지원되는 MLNX_OFED 드라이버가 설치되어 있으면 을 참조하십시오."[기술 요구 사항](#)"

- BeeGFS 파일 시스템 및 Pacemaker 클러스터 구성에 유효한 백업이 있습니다.
- 클러스터가 정상 상태입니다.

펌웨어 업데이트 준비

NVIDIA의 MLNX_OFED 드라이버와 함께 번들로 제공되는 노드의 어댑터 펌웨어를 업데이트하려면 NVIDIA의 유틸리티를 사용하는 것이 좋습니다 `mlxfwmanager`. 업데이트를 시작하기 전에 어댑터의 펌웨어 이미지를 "[NVIDIA의 지원 사이트](#)" 다운로드하여 각 파일 노드에 저장합니다.



Lenovo ConnectX-7 어댑터의 경우 `mlxfwmanager_LES` NVIDIA 페이지에서 사용할 수 있는 도구를 "[OEM 펌웨어](#)" 사용합니다.

롤링 업데이트 접근 방식

이 접근 방식은 3개 이상의 노드가 있는 HA 클러스터에 권장됩니다. 이 접근 방식에는 한 번에 하나의 파일 노드에서 어댑터 펌웨어를 업데이트하여 HA 클러스터가 서비스 요청을 처리할 수 있습니다. 하지만 이 시간 동안 I/O를 처리하지 않는 것이 좋습니다.

1. 각 BeeGFS 서비스가 1차 노드에서 실행되고 있는 상태에서 클러스터가 최적의 상태인지 확인합니다. 자세한 내용은 ["클러스터의 상태를 검사합니다"](#) 참조하십시오.
2. 업데이트할 파일 노드를 선택하고 대기 모드로 전환하면 해당 노드에서 모든 BeeGFS 서비스를 트레이닝(또는 이동)합니다.

```
pcs node standby <HOSTNAME>
```

3. 다음을 실행하여 노드의 서비스가 방전되었는지 확인합니다.

```
pcs status
```

대기 상태인 노드에 대해 보고하는 서비스가 없는지 확인합니다 `started`.



클러스터 크기에 따라 BeeGFS 서비스가 자매 노드로 이동하는 데 몇 초 또는 몇 분이 걸릴 수 있습니다. BeeGFS 서비스가 자매 노드에서 시작되지 않는 경우 ["문제 해결 설명서"](#)를 참조하십시오.

4. `mlxfwmanager` 유틸리티를 사용하여 어댑터 펌웨어를 업데이트합니다.

```
mlxfwmanager -i <path/to/firmware.bin> -u
```

펌웨어 업데이트를 수신하는 각 어댑터에 대해 PCI Device Name 확인합니다.

5. 유틸리티를 사용하여 각 어댑터를 재설정하여 `mlxfwreset` 새 펌웨어를 적용합니다.



일부 펌웨어 업데이트에서는 업데이트를 적용하기 위해 재부팅해야 할 수 있습니다. 지침은 ["NVIDIA의 mlxfwreset 제한 사항"](#) 참조하십시오. 재부팅이 필요한 경우 어댑터를 재설정하는 대신 재부팅을 수행하십시오.

- a. OpenSM 서비스를 중지합니다.

```
systemctl stop opensm
```

- b. 앞서 언급한 각 명령에 대해 다음 명령을 PCI Device Name 실행합니다.

```
mlxfwreset -d <pci_device_name> reset -y
```

- c. OpenSM 서비스를 시작합니다.

```
systemctl start opensm
```

- d. 다시 시작하세요 eseries_nvme_ib.service .

```
systemctl restart eseries_nvme_ib.service
```

- e. E-시리즈 스토리지 어레이의 볼륨이 있는지 확인하세요.

```
multipath -ll
```

1. `ibstat` 다음을 실행하여 모든 어댑터가 원하는 펌웨어 버전에서 실행되고 있는지 확인합니다.

```
ibstat
```

2. 노드에서 심장박동기 클러스터 서비스를 시작합니다.

```
pcs cluster start <HOSTNAME>
```

3. 노드를 대기 모드에서 해제합니다.

```
pcs node unstandby <HOSTNAME>
```

4. 모든 BeeGFS 서비스를 기본 노드로 다시 재배포:

```
pcs resource relocate run
```

모든 어댑터가 업데이트될 때까지 클러스터의 각 파일 노드에 대해 이 단계를 반복합니다.

2노드 클러스터 업데이트 접근 방식

이 접근 방식은 2개의 노드만 있는 HA 클러스터에 권장됩니다. 이 방법은 롤링 업데이트와 유사하지만 한 노드의 클러스터 서비스가 중지될 때 서비스 다운타임을 방지하기 위한 추가 단계가 포함되어 있습니다.

1. 각 BeeGFS 서비스가 1차 노드에서 실행되고 있는 상태에서 클러스터가 최적의 상태인지 확인합니다. 자세한 내용은 ["클러스터의 상태를 검사합니다"](#) 참조하십시오.
2. 업데이트할 파일 노드를 선택하고 노드를 대기 모드로 전환하면 해당 노드에서 모든 BeeGFS 서비스를 압축(또는 이동)합니다.

```
pcs node standby <HOSTNAME>
```

3. 다음을 실행하여 노드의 리소스가 소모되었는지 확인합니다.

```
pcs status
```

대기 상태인 노드에 대해 보고하는 서비스가 없는지 확인합니다 Started.



클러스터 크기에 따라 BeeGFS 서비스가 자매 노드로 보고되려면 몇 초 또는 몇 분이 걸릴 수 있습니다. BeeGFS 서비스를 시작하지 못하는 경우 ["문제 해결 설명서"](#)를 참조하십시오.

4. 클러스터를 유지보수 모드로 전환합니다.

```
pcs property set maintenance-mode=true
```

5. `mlxfwmanager`를 사용하여 어댑터 펌웨어를 `mlxfwmanager` 업데이트합니다.

```
mlxfwmanager -i <path/to/firmware.bin> -u
```

펌웨어 업데이트를 수신하는 각 어댑터에 대해 `PCI Device Name` 확인합니다.

6. 유틸리티를 사용하여 각 어댑터를 재설정하여 `mlxfwreset` 새 펌웨어를 적용합니다.



일부 펌웨어 업데이트에서는 업데이트를 적용하기 위해 재부팅해야 할 수 있습니다. 지침은 ["NVIDIA의 mlxfwreset 제한 사항"](#) 참조하십시오. 재부팅이 필요한 경우 어댑터를 재설정하는 대신 재부팅을 수행하십시오.

a. OpenSM 서비스를 중지합니다.

```
systemctl stop opensm
```

b. 앞서 언급한 각 명령에 대해 다음 명령을 PCI Device Name 실행합니다.

```
mlxfwreset -d <pci_device_name> reset -y
```

c. OpenSM 서비스를 시작합니다.

```
systemctl start opensm
```

7. `ibstat` 다음을 실행하여 모든 어댑터가 원하는 펌웨어 버전에서 실행되고 있는지 확인합니다.

```
ibstat
```

8. 노드에서 심장박동기 클러스터 서비스를 시작합니다.

```
pcs cluster start <HOSTNAME>
```

9. 노드를 대기 모드에서 해제합니다.

```
pcs node unstandby <HOSTNAME>
```

10. 클러스터를 유지보수 모드에서 해제합니다.

```
pcs property set maintenance-mode=false
```

11. 모든 BeeGFS 서비스를 기본 노드로 다시 재배치:

```
pcs resource relocate run
```

모든 어댑터가 업데이트될 때까지 클러스터의 각 파일 노드에 대해 이 단계를 반복합니다.

E-Series 스토리지 시스템을 업그레이드합니다

HA 클러스터의 E-Series 스토리지 어레이 구성 요소를 업그레이드하려면 다음 단계를

따르십시오.

개요

최신 펌웨어로 HA 클러스터의 NetApp E-Series 스토리지 어레이를 최신 상태로 유지하면 최적의 성능과 향상된 보안을 보장할 수 있습니다. 스토리지 어레이용 펌웨어 업데이트는 SANtricity OS, NVSRAM 및 드라이브 펌웨어 파일을 통해 적용됩니다.



스토리지 어레이는 HA 클러스터를 온라인으로 업그레이드할 수 있지만 모든 업그레이드를 위해 클러스터를 유지보수 모드로 전환하는 것이 좋습니다.

블록 노드 업그레이드 단계

다음 단계에서는 `Netapp_Eseries.Santricity Ansible` 컬렉션을 사용하여 스토리지 어레이의 펌웨어를 업데이트하는 방법을 간략히 설명합니다. 계속하기 전에 ["업그레이드 고려 사항"](#) E-Series 시스템 업데이트에 대한 를 검토하십시오.



SANtricity OS 11.80 이상 릴리즈로 업그레이드하는 것은 11.70.5P1부터 가능합니다. 추가 업그레이드를 적용하기 전에 먼저 스토리지 어레이를 11.70.5P1로 업그레이드해야 합니다.

1. Ansible 제어 노드에서 최신 SANtricity Ansible Collection을 사용하고 있는지 확인합니다.

- 에 액세스하여 컬렉션 업그레이드용 ["Ansible 갤러리"](#)에서 다음 명령을 실행합니다.

```
ansible-galaxy collection install netapp_eseries.santricity --upgrade
```

- 오프라인 업그레이드의 경우 에서 컬렉션 타르볼을 다운로드하여 ["Ansible 갤러리"](#) 컨트롤 노드로 전송한 후 다음을 실행합니다.

```
ansible-galaxy collection install netapp_eseries-santricity-  
<VERSION>.tar.gz --upgrade
```

을 참조하십시오 ["컬렉션 설치 중"](#) 를 참조하십시오.

2. 스토리지 배열 및 드라이브에 대한 최신 펌웨어를 가져옵니다.

a. 펌웨어 파일을 다운로드합니다.

- * SANtricity OS 및 NVSRAM: * ["NetApp Support 사이트"](#) 스토리지 어레이 모델에 맞는 SANtricity OS 및 NVSRAM 최신 릴리스로 이동하여 다운로드합니다.
- * 드라이브 펌웨어: * 로 ["E-Series 디스크 펌웨어 사이트입니다"](#) 이동하여 각 스토리지 배열의 드라이브 모델에 대한 최신 펌웨어를 다운로드합니다.

b. SANtricity OS, NVSRAM 및 드라이브 펌웨어 파일을 Ansible 제어 노드의 `<inventory_directory>/packages` 디렉토리에 저장합니다.

3. 필요한 경우 업데이트가 필요한 모든 스토리지 어레이(블록 노드)를 포함하도록 클러스터의 Ansible 인벤토리 파일을 업데이트합니다. 지침은 ["Ansible 인벤토리 개요"](#) 섹션을 참조하십시오.

4. 1차 노드에서 각 BeeGFS 서비스를 통해 클러스터가 최적의 상태로 유지되도록 합니다. 자세한 내용은 을

"클러스터의 상태를 검사합니다" 참조하십시오.

5. 의 지침에 따라 클러스터를 유지보수 모드로 "클러스터를 유지보수 모드로 전환합니다"전환합니다.
6. 이라는 새 Ansible 플레이북을 `update_block_node_playbook.yml` 생성합니다. SANtricity OS, NVSRAM 및 드라이브 펌웨어 버전을 원하는 업그레이드 경로로 대체하여 플레이북에 다음 콘텐츠를 채웁니다.

```
- hosts: eseries_storage_systems
gather_facts: false
any_errors_fatal: true
collections:
  - netapp_eseries.santricity
vars:
  eseries_firmware_firmware: "packages/<SantricityOS>.dlp"
  eseries_firmware_nvram: "packages/<NVSRAM>.dlp"
  eseries_drive_firmware_firmware_list:
    - "packages/<drive_firmware>.dlp"
  eseries_drive_firmware_upgrade_drives_online: true

tasks:
  - name: Configure NetApp E-Series block nodes.
    import_role:
      name: nar_santricity_management
```

7. 업데이트를 시작하려면 Ansible 컨트롤 노드에서 다음 명령을 실행합니다.

```
ansible-playbook -i inventory.yml update_block_node_playbook.yml
```

8. 플레이북이 완료된 후 각 스토리지 어레이가 최적의 상태인지 확인합니다.
9. 클러스터를 유지보수 모드에서 해제하고 각 BeeGFS 서비스가 기본 노드에 있을 때 클러스터가 최적의 상태인지 확인합니다.

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.