BeeGFS 클러스터 관리

BeeGFS on NetApp with E-Series Storage

NetApp
January 27, 2026

This PDF was generated from https://docs.netapp.com/ko-kr/beegfs/administer/clusters-overview.html on January 27, 2026. Always check docs.netapp.com for the latest.

목차

BeeGFS 클러스터 관리

개요, 주요 개념 및 용어

구축된 BeeGFS HA 클러스터를 관리하는 방법에 대해 알아보십시오.

개요

이 섹션은 구축된 BeeGFS HA 클러스터를 관리해야 하는 클러스터 관리자를 대상으로 합니다. Linux HA 클러스터에 익숙한 사람조차도 이 가이드를 완전히 읽어야 합니다. Ansible을 사용하여 재구성을 수행할 때는 특히 클러스터 관리 방법에 여러 가지 차이점이 있기 때문입니다.

주요 개념

이러한 개념 중 일부는 메인 "용어 및 개념" 페이지에 소개되었지만 BeeGFS HA 클러스터의 컨텍스트에서 다시 소개하면 도움이 됩니다.

- 클러스터 노드:** 심장박동기 및 Corosync 서비스를 실행하고 HA 클러스터에 참여하는 서버.
- 파일 노드:** 하나 이상의 BeeGFS 관리, 메타데이터 또는 스토리지 서비스를 실행하는 데 사용되는 클러스터 노드입니다.
- 블록 노드:** 파일 노드에 블록 스토리지를 제공하는 NetApp E-Series 스토리지 시스템 이러한 노드는 자체 독립형 HA 기능을 제공하므로 BeeGFS HA 클러스터에 참여하지 않습니다. 각 노드는 블록 계층에서 고가용성을 제공하는 2개의 스토리지 컨트롤러로 구성됩니다.
- BeeGFS 서비스:** BeeGFS 관리, 메타데이터 또는 스토리지 서비스 각 파일 노드는 블록 노드의 볼륨을 사용하여 데이터를 저장하는 하나 이상의 서비스를 실행합니다.
- 빌딩 블록:** BeeGFS 파일 노드, E-Series 블록 노드 및 BeeGFS 서비스를 표준화된 방식으로 구축하여 NetApp 검증 아키텍처에 따라 BeeGFS HA 클러스터/파일 시스템을 간편하게 확장할 수 있습니다. 맞춤형 HA 클러스터도 지원되지만, 확장을 단순화하기 위해 유사한 구성 요소 접근 방식을 따르는 경우가 많습니다.
- BeeGFS HA Cluster:** 블록 노드에서 지원하는 BeeGFS 서비스를 실행하는 데 사용되는 확장 가능한 수의 파일 노드를 통해 BeeGFS 데이터를 고가용성 방식으로 저장합니다. 패키징 및 배포를 위해 Ansible을 사용하여 업계에서 검증된 오픈 소스 구성 요소 페이스 메이커 및 Corosync를 기반으로 합니다.
- 클러스터 서비스: ** 클러스터에 참여하는 각 노드에서 실행 중인 심장박동기 및 Corosync 서비스를 나타냅니다. 참고: 노드가 BeeGFS 서비스를 실행하지 않고 단지 두 개의 파일 노드만 필요한 경우 "Tiebreaker" 노드로 클러스터에 참여할 수 있습니다.
- 클러스터 리소스:** 클러스터에서 실행 중인 각 BeeGFS 서비스에 대해 BeeGFS 모니터링 리소스와 BeeGFS 타겟, IP 주소(부동 IP) 및 BeeGFS 서비스 자체에 대한 리소스가 포함된 리소스 그룹이 표시됩니다.

Ansible:** 소프트웨어 프로비저닝, 구성 관리 및 애플리케이션 배포를 위한 도구로 인프라를 코드로 구현할 수 있습니다. BeeGFS HA 클러스터를 패키지화하여 NetApp에서 BeeGFS 구축, 재구성 및 업데이트 프로세스를 간소화합니다.

- PCS:** 클러스터의 파일 노드에서 사용할 수 있는 명령줄 인터페이스입니다. 이 인터페이스는 클러스터의 노드 및 리소스 상태를 쿼리하고 제어하는 데 사용됩니다.

일반 용어

- 장애 조치:** 각 BeeGFS 서비스에는 해당 노드에 장애가 발생하지 않는 한 실행되는 기본 파일 노드가 있습니다. 비기본/보조 파일 노드에서 BeeGFS 서비스를 실행하는 경우 페일오버 중인 것으로 표시됩니다.
- 페일백:** 비기본 파일 노드에서 기본 설정 노드로 BeeGFS 서비스를 이동하는 동작
- HA 쌍:** 동일한 블록 노드 세트에 액세스할 수 있는 두 개의 파일 노드를 HA 쌍이라고도 합니다. 이는 NetApp 전체에서 사용되는 일반적인 용어로 서로 "이어갈" 수 있는 2개의 스토리지 컨트롤러 또는 노드를 참조하는 데 사용됩니다.
- 유지 관리 모드: ** 모든 리소스 모니터링을 비활성화하고 심장박동기 장치가 클러스터의 리소스를 이동하거나 관리하지 못하게 합니다(의 섹션 참조"유지보수 모드").
- HA 클러스터:** 클러스터의 여러 노드 간에 페일오버하여 가용성이 높은 BeeGFS 파일 시스템을 생성할 수 있는 BeeGFS 서비스를 실행하는 하나 이상의 파일 노드. 파일 노드는 클러스터에서 BeeGFS 서비스의 하위 집합을 실행할 수 있는 HA 쌍으로 구성되는 경우가 많습니다.

Ansible과 PCS 도구를 사용해야 하는 경우

HA 클러스터를 관리하기 위해 Ansible과 PCS 명령줄 도구를 사용해야 하는 경우는 언제입니까?

모든 클러스터 배포 및 재구성 작업은 외부 Ansible 제어 노드의 Ansible을 사용하여 완료해야 합니다. 클러스터 상태의 일시적인 변경(예: 대기 모드 내외부로 노드 배치)은 일반적으로 클러스터의 한 노드(성능이 저하되지 않거나 유지보수를 수행하려고 하지 않는 노드)에 로그인하고 PCS 명령줄 도구를 사용하여 수행됩니다.

리소스, 제약, 속성 및 BeeGFS 서비스 자체를 비롯한 클러스터 구성을 수정하려면 항상 Ansible을 사용해야 합니다. Ansible 인벤토리 및 플레이북의 최신 복사본(소스 제어 기능을 통해 변경 사항을 추적하는 것이 이상적)을 유지하는 것은 클러스터를 유지하는 데 필요한 부분입니다. 구성을 변경해야 하는 경우 인벤토리를 업데이트하고 BeeGFS HA 역할을 가져오는 Ansible 플레이북을 다시 실행합니다.

HA 역할은 클러스터를 유지 관리 모드로 설정한 다음 필요한 변경을 수행한 후 BeeGFS 또는 클러스터 서비스를 다시 시작하여 새 구성을 적용하는 작업을 처리합니다. 전체 노드 재부팅은 일반적으로 초기 구현 이후에 필요하지 않기 때문에 Ansible을 다시 실행하는 것은 일반적으로 "안전한" 절차로 간주되지만, BeeGFS 서비스를 다시 시작해야 할 경우 유지보수 시간이나 근무 시간 외에 실행하는 것이 좋습니다. 이러한 재시작으로 인해 일반적으로 응용 프로그램 오류가 발생하지는 않지만 성능이 저하될 수 있습니다(일부 응용 프로그램이 다른 응용 프로그램보다 더 잘 처리할 수 있음).

또한 Ansible을 다시 실행하는 것은 전체 클러스터를 최적의 상태로 되돌리는 옵션이며, 경우에 따라 PC를 사용할 때보다 클러스터의 상태를 더 쉽게 복구할 수 있습니다. 특히 어떤 이유로 클러스터가 중단된 비상 상황에서는 모든 노드가 다시 실행 중인 Ansible을 사용하면 PC를 사용하는 것보다 신속하고 안정적으로 클러스터를 복구할 수 있습니다.

클러스터의 상태를 검사합니다

PC를 사용하여 클러스터의 상태를 봅니다.

개요

실행 중입니다 `pcs status` 모든 클러스터 노드에서 클러스터의 전체 상태와 각 리소스의 상태(예: BeeGFS 서비스 및 해당 종속성)를 확인하는 가장 쉬운 방법입니다. 이 섹션에서는 의 출력에서 확인할 수 있는 내용을 설명합니다 `pcs`

`status` 명령.

의 출력 이해 `pcs status`

실행 `pcs status` 클러스터 서비스(박동조율기 및 Corosync)가 시작되는 모든 클러스터 노드에서. 출력 맨 위에는 클러스터의 요약이 표시됩니다.

```
[root@beegfs_01 ~]# pcs status
Cluster name: hacluster
Cluster Summary:
  * Stack: corosync
  * Current DC: beegfs_01 (version 2.0.5-9.el8_4.3-ba59be7122) - partition
with quorum
  * Last updated: Fri Jul  1 13:37:18 2022
  * Last change:  Fri Jul  1 13:23:34 2022 by root via cibadmin on
beegfs_01
  * 6 nodes configured
  * 235 resource instances configured
```

아래 섹션에는 클러스터의 노드가 나열됩니다.

```
Node List:
  * Node beegfs_06: standby
  * Online: [ beegfs_01 beegfs_02 beegfs_04 beegfs_05 ]
  * OFFLINE: [ beegfs_03 ]
```

이는 대기 또는 오프라인 상태인 노드를 나타냅니다. 대기 상태의 노드는 여전히 클러스터에 참여하고 있지만 리소스 실행 부적격으로 표시되어 있습니다. 오프라인 상태인 노드는 노드가 수동으로 중지되었거나 노드가 재부팅/종료되었기 때문에 해당 노드에서 클러스터 서비스가 실행되고 있지 않음을 나타냅니다.

노드가 처음 시작될 때 클러스터 서비스가 중지되며, 비정상적인 노드로 리소스가 실수로 페일백되지 않도록 수동으로 시작해야 합니다.

노드가 비관리적 이유(예: 실패)로 인해 대기 또는 오프라인 상태인 경우 노드의 상태 옆에 괄호 안에 추가 텍스트가 표시됩니다. 예를 들어 펜싱을 사용하지 않도록 설정하고 리소스에 장애가 발생하면 표시됩니다 `Node <HOSTNAME>: standby (on-fail)`. 다른 가능한 상태는 입니다 `Node <HOSTNAME>: UNCLEAN (offline)`펜싱이 실패하여 클러스터가 노드 상태를 확인할 수 없음을 나타내는 경우(다른 노드에서 리소스가 시작되는 것을 차단할 수 있음) 잠시 노드가 펜싱되는 것으로 나타나지만 지속합니다.

다음 섹션에는 클러스터의 모든 리소스 목록과 해당 상태가 표시됩니다.

```
Full List of Resources:
  * mgmt-monitor    (ocf::eseries:beegfs-monitor):   Started beegfs_01
  * Resource Group: mgmt-group:
    * mgmt-FS1  (ocf::eseries:beegfs-target):    Started beegfs_01
    * mgmt-IP1  (ocf::eseries:beegfs-ipaddr2):   Started beegfs_01
    * mgmt-IP2  (ocf::eseries:beegfs-ipaddr2):   Started beegfs_01
    * mgmt-service  (systemd:beegfs-mgmtd):  Started beegfs_01
[...]
```

노드와 마찬가지로 리소스에 문제가 있는 경우 괄호 안의 리소스 상태 옆에 추가 텍스트가 표시됩니다. 예를 들어 페이스메이커의 리소스 중지를 요청했으나 할당된 시간 내에 완료되지 못한 경우 심장박동기는 노드를 울타리로 만들려고 시도합니다. 펜싱이 비활성화되거나 펜싱 작업이 실패하는 경우 리소스 상태는 입니다 `FAILED <HOSTNAME> (blocked)` 다른 노드에서 심장박동조율기를 시작할 수 없습니다.

BeeGFS HA 클러스터가 여러 BeeGFS에 최적화된 맞춤형 OCF 리소스 에이전트를 사용하는 것을 주목할 필요가 있습니다. 특히 BeeGFS 모니터는 특정 노드의 BeeGFS 리소스를 사용할 수 없을 때 페일오버를 트리거합니다.

HA 클러스터와 BeeGFS를 재구성합니다

Ansible을 사용하여 클러스터를 재구성합니다.

개요

일반적으로 BeeGFS HA 클러스터의 모든 측면을 재구성하려면 Ansible 인벤토리를 업데이트하고 `ansible-playbook` 명령을 다시 실행해야 합니다. 여기에는 알림 업데이트, 영구 펜싱 구성 변경 또는 BeeGFS 서비스 구성 조정 등이 포함됩니다. 이러한 옵션은 `group_vars/ha_cluster.yml` 파일을 사용하여 조정되며 전체 옵션 목록은 "일반 파일 노드 구성을 지정합니다"섹션에서 확인할 수 있습니다.

유지 관리를 수행하거나 클러스터를 서비스할 때 관리자가 알아야 하는 구성 옵션에 대한 자세한 내용은 아래를 참조하십시오.

펜싱 비활성화 및 활성화 방법

클러스터를 설정할 때 펜싱은 기본적으로 활성화/필요합니다. 경우에 따라 펜싱을 일시적으로 비활성화하여 운영 체제 업그레이드와 같은 특정 유지보수 작업을 수행할 때 노드가 실수로 종료되지 않도록 하는 것이 좋습니다. 이 기능은 수동으로 비활성화할 수 있지만 관리자가 주의해야 할 단점이 있습니다.

옵션 **1: Ansible**을 사용하여 펜싱 비활성화(권장)

Ansible을 사용하여 펜싱을 비활성화하면 BeeGFS 모니터의 장애 발생 시 동작이 "fence"에서 "standby"로 변경됩니다. 즉, BeeGFS 모니터가 장애를 감지하면 노드를 대기 상태로 두고 모든 BeeGFS 서비스를 페일오버하려고 합니다. 활성 상태의 문제 해결/테스트 이외의 경우 일반적으로 옵션 2보다 더 권장됩니다. 단점은 리소스가 원래 노드에서 중지되지 않을 경우 다른 위치에서 시작하는 것이 차단된다는 것입니다(즉, 펜싱은 일반적으로 운영 클러스터에 필요합니다).

1. 에 있는 Ansible 인벤토리에 들어 있습니다 `groups_vars/ha_cluster.yml` 다음 구성을 추가합니다.

```
beegfs_ha_cluster_crm_config_options:
    stonith-enabled: False
```

2. Ansible 플레이북을 다시 실행하여 클러스터의 변경 사항을 적용합니다.

옵션 **2:** 수동으로 펜싱을 비활성화합니다**.**

경우에 따라 Ansible을 다시 실행하지 않고 펜싱을 일시적으로 비활성화할 수 있으며, 클러스터에 대한 문제 해결 또는 테스트를 용이하게 할 수 있습니다.

> 이 구성에서 BeeGFS 모니터가 장애를 감지하면 클러스터가 해당 리소스 그룹을 중지하려고 시도합니다. 전체 페일오버를 트리거하거나 영향을 받는 리소스 그룹을 다시 시작하거나 다른 호스트로 이동하려고 시도하지 않습니다. 복구하려면 문제를 해결한 다음 를 실행하십시오 `pcs resource cleanup` 또는 노드를 수동으로 대기 상태로 전환합니다.

단계:

1. 펜싱(STONITH)이 전역적으로 활성화 또는 비활성화되었는지 확인하려면 다음을 실행하십시오. `pcs property show stonith-enabled`
2. 펜싱 실행을 비활성화하려면: `pcs property set stonith-enabled=false`
3. 펜싱 실행을 활성화하려면: `pcs property set stonith-enabled=true`

> 다음에 Ansible 플레이북을 실행하면 이 설정이 재정의됩니다.

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"을 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	begfs 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 갤럭시"에서 다음 명령을 실행합니다.

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

오프라인 컬렉션 업그레이드의 경우 에서 컬렉션을 다운로드하십시오 "Ansible 갤럭시" 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 **7.3.0**에서 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 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.
   a. 클러스터 구성 백업에 대한 지침은 "BeeGFS 백업 설명서"을 참조하십시오.
   b. 기존 관리 데이터 디렉터리를 백업합니다.

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

   c. beegfs 클라이언트에서 다음 명령을 실행하고 출력을 참조용으로 저장합니다.

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

   d. 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-mgmtd beegfs-storage beegfs-meta libbeegfs-ib
```

표준 클러스터에서 `beegfs-mgmtd` 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-mgmtd --import-from
-v7=/mnt/mgmt_tgt_mgmt01/data/
```

   예상 출력:

```
Created new database version 3 at "/var/lib/beegfs/mgmtd.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/mgmtd.sqlite /mnt/mgmt_tgt_mgmt01/data/
```

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

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

`beegfs-mgmtd.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 `.conf`에서 `.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-mgmtd.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-
mgmtd.toml"
Successfully initialized certificate verification library.
Successfully loaded license certificate: TMP-113489268
Opened database at "/mnt/mgmt_tgt_mgmt01/data/mgmtd.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-mgmtd.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 을 "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 를 참조하십시오"문제 해결 설명서".

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 버전'이 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 을 "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 에서 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 을 "클러스터의 상태를 검사합니다" 참조하십시오.
2. 업데이트할 파일 노드를 선택하고 대기 모드로 전환하면 해당 노드에서 모든 BeeGFS 서비스를 드레이닝(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 를 참조하십시오"문제 해결 설명서".

4. 을 사용하여 어댑터 펌웨어를 `mlxfwmanager` 업데이트합니다.

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

펌웨어 업데이트를 수신하는 각 어댑터에 대해 를 `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 을 "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 재배치:

```
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 을 "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 `Started` 있습니다. BeeGFS 서비스를 시작하지 못하는 경우 를 참조하십시오"문제 해결 설명서".

4. 클러스터를 유지보수 모드로 전환합니다.

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

5. 을 사용하여 어댑터 펌웨어를 `mlxfwmanager` 업데이트합니다.

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

펌웨어 업데이트를 수신하는 각 어댑터에 대해 를 `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 을 "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 갤럭시"에서 다음 명령을 실행합니다.

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

   - 오프라인 업그레이드의 경우 에서 컬렉션 타르볼을 다운로드하여 "Ansible 갤럭시"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_nvsram: "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.

서비스 및 유지 관리

장애 조치 및 장애 복구 서비스

클러스터 노드 간에 BeeGFS 서비스 이동

개요

BeeGFS 서비스는 클러스터에서 노드 간에 페일오버를 수행하여 노드에 장애가 발생하거나 계획된 유지 관리를 수행해야 하는 경우 클라이언트가 파일 시스템에 계속 액세스할 수 있도록 합니다. 이 섹션에서는 장애를 복구한 후 관리자가 클러스터를 복구하거나 노드 간에 서비스를 수동으로 이동할 수 있는 다양한 방법에 대해 설명합니다.

단계

페일오버 및 페일백

페일오버(계획됨)

일반적으로 유지 관리를 위해 단일 파일 노드를 오프라인으로 전환해야 하는 경우 해당 노드에서 모든 BeeGFS 서비스를 이동(또는 드레이닝)해야 합니다. 먼저 노드를 대기 상태로 전환하여 이 작업을 수행할 수 있습니다.

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

를 사용하여 확인한 후 `pcs status` 모든 리소스가 대체 파일 노드에서 다시 시작되었습니다. 노드를 종료하거나 필요에 따라 변경할 수 있습니다.

페일백(계획된 페일오버 후)

BeeGFS 서비스를 기본 노드 첫 번째 실행으로 복구할 준비가 되면 `pcs status` "노드 목록"에서 상태가 대기 상태인지 확인합니다. 노드가 재부팅된 경우 클러스터 서비스를 온라인 상태로 전환할 때까지 오프라인 상태로 표시됩니다.

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

노드가 온라인 상태가 되면 다음을 사용하여 대기 모드에서 나오게 합니다.

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

마지막으로 다음을 통해 모든 BeeGFS 서비스를 기본 노드에 다시 재배치하십시오.

```
pcs resource relocate run
```

페일백(계획되지 않은 페일오버 후)

노드에 하드웨어 또는 기타 장애가 발생할 경우 HA 클러스터가 자동으로 대응하고 서비스를 정상 노드로 이동하여 관리자에게 수정 조치를 취할 수 있는 시간을 제공해야 합니다. 계속하기 전에 "문제 해결"섹션을 참조하여 장애 조치의 원인을 확인하고 미해결 문제를 해결하십시오. 노드 전원이 다시 켜지고 정상 상태가 되면 페일백을 진행할 수 있습니다.

예정되지 않은(또는 계획된) 재부팅 후 노드가 부팅될 때 클러스터 서비스가 자동으로 시작되도록 설정되지 않으므로 먼저 를 사용하여 노드를 온라인 상태로 가져와야 합니다.

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

그런 다음 리소스 장애를 정리하고 노드의 펜싱 기록을 재설정합니다.

```
pcs resource cleanup node=<HOSTNAME>
pcs stonith history cleanup <HOSTNAME>
```

에서 확인하십시오 `pcs status` 노드가 온라인 상태이고 정상 상태입니다. 기본적으로 BeeGFS 서비스는 실수로 리소스가 정상 상태가 아닌 노드로 다시 이동하는 것을 방지하기 위해 자동으로 페일백하지 않습니다. 준비되면 클러스터의 모든 리소스를 원하는 노드로 다시 돌려볼 수 있는 방법은 다음과 같습니다.

```
pcs resource relocate run
```

BeeGFS 서비스를 새 파일 노드로 영구적으로 이동합니다

개별 BeeGFS 서비스에 대해 선호하는 파일 노드를 영구적으로 변경하려면 선호하는 노드가 먼저 나열되도록 Ansible 인벤토리를 조정하고 Ansible 플레이북을 다시 실행하십시오.

예를 들어, 이 샘플 `inventory.yml` 파일에서 beegfs_01은 BeeGFS 관리 서비스를 실행하는 데 사용되는 파일 노드입니다.

```
        mgmt:
          hosts:
            beegfs_01:
            beegfs_02:
```

순서를 반대로 하면 beegfs_02에서 관리 서비스가 선호됩니다.

```
        mgmt:
          hosts:
            beegfs_02:
            beegfs_01:
```

BeeGFS 서비스를 대체 파일 노드로 임시 이동합니다

일반적으로 노드가 유지 관리 중인 경우 [failover and failback steps](#failover-and-failback)를 사용하여 해당 노드에서 모든 서비스를 이동하려고 합니다.

어떤 이유로 개별 서비스를 다른 파일 노드로 이동해야 하는 경우 다음을 실행합니다.

```
pcs resource move <SERVICE>-monitor <HOSTNAME>
```

> 개별 리소스 또는 리소스 그룹을 지정하지 마십시오. 재배치할 BeeGFS 서비스에 대한 모니터 이름을 항상 지정합니다. 예를 들어 BeeGFS 관리 서비스를 beegfs_02 실행으로 이동하려면 다음을 실행합니다 `pcs resource move mgmt-monitor beegfs_02`. 이 프로세스를 반복하여 하나 이상의 서비스를 원하는 노드에서 이동할 수 있습니다. 서비스를 사용하여 새 노드에서 재배치 /시작되었는지 `pcs status` 확인합니다.

BeeGFS 서비스를 기본 노드로 다시 이동하려면 먼저 임시 리소스 제약 조건을 해제합니다(여러 서비스에 필요한 경우 이 단계를 반복).

```
pcs resource clear <SERVICE>-monitor
```

그런 다음 실제로 서비스를 원하는 노드로 다시 이동할 준비가 되면 다음을 실행합니다.

```
pcs resource relocate run
```

참고 이 명령은 더 이상 임시 리소스 제약 조건이 없는 서비스를 기본 노드에 배치하지 않습니다.

클러스터를 유지보수 모드로 전환합니다

HA 클러스터가 운영 환경의 의도된 변경 사항에 실수로 반응하는 것을 방지합니다.

개요

클러스터를 유지보수 모드로 전환하면 모든 리소스 모니터링이 비활성화되고 박동기가 클러스터 리소스를 이동하거나 관리하는 것을 방지할 수 있습니다. 액세스를 방해하는 일시적인 장애 조건이 있더라도 모든 리소스는 원래 노드에서 계속 실행됩니다. 권장/유용한 시나리오는 다음과 같습니다.

- 파일 노드와 BeeGFS 서비스 간의 연결이 일시적으로 중단될 수 있는 네트워크 유지 보수
- 블록 노드 업그레이드
- 파일 노드 운영 체제, 커널 또는 기타 패키지 업데이트.

일반적으로 클러스터를 유지 관리 모드로 수동으로 설정하는 유일한 이유는 해당 클러스터가 환경의 외부 변경에 반응하지 않도록 하기 위해서입니다. 클러스터의 개별 노드에 물리적 복구가 필요한 경우 유지보수 모드를 사용하지 말고 위의 절차에 따라 해당 노드를 대기 모드에 두십시오. Ansible을 다시 실행하면 업그레이드 및 구성 변경을 포함하여 대부분의 소프트웨어 유지보수를 수행할 수 있도록 클러스터가 유지보수 모드로 자동으로 전환됩니다.

단계

클러스터가 유지보수 모드인지 확인하려면 다음을 실행합니다.

```
pcs property config
```

`maintenance-mode`클러스터가 정상적으로 작동하는 경우에는 속성이 나타나지 않습니다. 클러스터가 현재 유지 관리 모드에 있는 경우 속성은 로 보고됩니다. `true` 유지보수 모드를 활성화하려면 다음을 실행하십시오.

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

PC 상태를 실행하고 모든 리소스에 "(관리되지 않음)"이 표시되는지 확인하여 확인할 수 있습니다. 클러스터를 유지보수 모드에서 제외하려면 다음을 실행하십시오.

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

클러스터를 중지하고 시작합니다

정상적으로 HA 클러스터를 중지 및 시작합니다.

개요

이 섹션에서는 BeeGFS 클러스터를 정상적으로 종료하고 다시 시작하는 방법에 대해 설명합니다. 이러한 작업이 필요할 수 있는 시나리오에는 전기 유지보수 또는 데이터 센터 또는 랙 간 마이그레이션이 포함됩니다.

단계

어떤 이유로든 전체 BeeGFS 클러스터를 중지하고 모든 서비스를 종료해야 하는 경우 다음을 실행합니다.

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

또한 개별 노드에서 클러스터를 중지할 수도 있습니다(다른 노드로 자동으로 서비스 페일오버). 먼저 노드를 대기 상태로 두는 것이 좋지만("페일오버"섹션 참조):

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

모든 노드에서 클러스터 서비스 및 리소스를 시작하려면 다음을 실행합니다.

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

또는 다음 특정 노드에서 서비스를 시작합니다.

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

이 시점에서 를 실행합니다 `pcs status` 모든 노드에서 클러스터와 BeeGFS 서비스가 시작되는지, 그리고 원하는 노드에서 서비스가 실행되고 있는지 확인합니다.

클러스터 크기에 따라 전체 클러스터가 중지되거나 에서 시작된 것으로 표시되는 데 몇 초 또는 몇 분이 걸릴 수 `pcs status` 있습니다. 가 5분 이상 중단된 경우 `pcs cluster <COMMAND>` "Ctrl+C"를 실행하여 명령을 취소하기 전에 클러스터의 각 노드에 로그인하여 `pcs status` 클러스터 서비스(Corosync/Pacemaker)가 해당 노드에서 계속 실행되고 있는지 확인하십시오. 클러스터가 여전히 활성 상태인 모든 노드에서 클러스터를 차단하는 리소스를 확인할 수 있습니다. 문제를 수동으로 해결하고 명령을 완료하거나 다시 실행하여 나머지 서비스를 중지할 수 있습니다.

파일 노드를 바꿉니다

원래 서버에 오류가 있는 경우 파일 노드를 교체합니다.

개요

다음은 클러스터의 파일 노드를 교체하는 데 필요한 단계에 대한 개요입니다. 다음 단계에서는 하드웨어 문제로 인해 파일 노드가 실패했으며 동일한 새 파일 노드로 교체된다고 가정합니다.

단계:

1. 파일 노드를 물리적으로 교체하고 블록 노드 및 스토리지 네트워크에 대한 모든 케이블 연결을 복원합니다.
2. Red Hat 서브스크립션 추가를 포함하여 파일 노드에 운영 체제를 다시 설치합니다.
3. 파일 노드에서 관리 및 BMC 네트워킹을 구성합니다.
4. 호스트 이름, IP, PCIe-논리 인터페이스 매핑 또는 새 파일 노드에 대해 변경된 사항이 있는 경우 Ansible 인벤토리를 업데이트합니다. 일반적으로 노드가 동일한 서버 하드웨어로 교체되었고 원래 네트워크 구성을 사용하는 경우에는 필요하지 않습니다.
   a. 예를 들어 호스트 이름이 변경된 경우 노드의 인벤토리 파일을 생성하거나 이름을 변경합니다 (`host_vars/<NEW_NODE>.yml`)를 선택한 다음 Ansible 인벤토리 파일에 저장합니다 (`inventory.yml`)에서 이전 노드의 이름을 새 노드 이름으로 바꿉니다.

```
all:
    ...
    children:
    ha_cluster:
        children:
        mgmt:
            hosts:
            node_h1_new:    # Replaced "node_h1" with "node_h1_new"
            node_h2:
```

5. 클러스터의 다른 노드 중 하나에서 이전 노드를 제거합니다. `pcs cluster node remove <HOSTNAME>`.

   ⚠ 이 단계를 실행하기 전에 진행하지 마십시오.

6. Ansible 제어 노드에서:
   a. 다음을 사용하여 이전 SSH 키를 제거합니다.

```
`ssh-keygen -R <HOSTNAME_OR_IP>`
```

   b. 바꾸기 노드에 대해 암호 없는 SSH를 다음으로 구성:

```
ssh-copy-id <USER>@<HOSTNAME_OR_IP>
```

7. Ansible 플레이북을 다시 실행하여 노드를 구성하고 클러스터에 추가합니다.

```
ansible-playbook -i <inventory>.yml <playbook>.yml
```

8. 이때 를 실행합니다 `pcs status` 교체된 노드가 나열되고 서비스가 실행 중인지 확인합니다.

클러스터를 확장 또는 축소합니다

클러스터에서 구성 요소를 추가하거나 제거합니다.

개요

이 섹션에서는 BeeGFS HA 클러스터의 크기를 조정하는 다양한 고려 사항 및 옵션에 대해 설명합니다. 일반적으로 클러스터 크기는 구성 요소를 추가 또는 제거하여 조정합니다. 구성 요소는 일반적으로 2개의 파일 노드를 HA 쌍으로 설정합니다. 필요한 경우 개별 파일 노드(또는 다른 유형의 클러스터 노드)를 추가하거나 제거할 수도 있습니다.

클러스터에 빌딩 블록 추가

고려 사항

추가 구성 요소를 추가하여 클러스터를 늘리는 것은 간단한 프로세스입니다. 시작하기 전에 각 개별 HA 클러스터의 최소 및 최대 클러스터 노드 수에 대한 제한을 염두에 두고, 기존 HA 클러스터에 노드를 추가하거나 새로운 HA 클러스터를 생성해야 하는지 확인합니다. 일반적으로 각 구성 요소는 2개의 파일 노드로 구성되지만, 3개의 노드는 쿼럼(quorum)을 설정하기 위해 클러스터당 최소 노드 수이고 10개는 권장(테스트)입니다. 고급 시나리오의 경우 2노드 클러스터를 구축할 때 BeeGFS 서비스를 실행하지 않는 단일 "Tiebreaker" 노드를 추가할 수 있습니다. 이러한 배포를 고려 중인 경우 NetApp 지원에 문의하십시오.

클러스터를 확장하는 방법을 결정할 때는 이러한 제한 사항과 향후 클러스터 확장에 대한 예상에 유의하십시오. 예를 들어, 6노드 클러스터가 있고 4노드를 더 추가해야 하는 경우 새 HA 클러스터를 시작하는 것이 좋습니다.

단일 BeeGFS 파일 시스템은 여러 독립 HA 클러스터로 구성될 수 있습니다. 따라서 파일 시스템은 기본 HA 클러스터 구성 요소의 권장/하드 제한보다 훨씬 높은 확장성을 유지할 수 있습니다.

단계

구성 요소를 클러스터에 추가할 때 `host_vars` 각 새 파일 노드 및 블록 노드(E-Series 스토리지)에 대한 파일을 생성해야 합니다. 이러한 호스트의 이름은 생성할 새 리소스와 함께 인벤토리에 추가해야 합니다. `group_vars`각 새 리소스에 대해 해당 파일을 생성해야 합니다. "사용자 지정 아키텍처 사용"자세한 내용은 섹션을 참조하십시오.

올바른 파일을 생성한 후 다음 명령을 사용하여 자동화를 다시 실행해야 합니다.

```
ansible-playbook -i <inventory>.yml <playbook>.yml
```

클러스터에서 빌딩 블록 제거

구성 요소를 폐기해야 할 경우에는 다음과 같은 여러 가지 사항을 고려해야 합니다.

- 이 빌딩 블록에서 실행 중인 BeeGFS 서비스는 무엇입니까?
- 파일 노드만 사용 중지되며 블록 노드를 새 파일 노드에 연결해야 합니까?

- 전체 빌딩 블록이 사용 중단된 경우 데이터를 새 빌딩 블록으로 이동하거나, 클러스터의 기존 노드로 분산하거나, 새로운 BeeGFS 파일 시스템 또는 다른 스토리지 시스템으로 이동해야 합니까?
- 이 문제는 중단 중에 발생할 수 있습니까? 아니면 중단 없이 수행해야 합니까?
- 구성 요소를 적극적으로 사용하고 있습니까, 아니면 주로 더 이상 활성 상태가 아니지 않은 데이터가 포함되어 있습니까?

가능한 다양한 시작 지점과 원하는 종료 상태 때문에 NetApp 지원에 문의하여 고객 환경과 요구 사항에 따라 최상의 전략을 파악하고 구현할 수 있습니다.

문제 해결

BeeGFS HA 클러스터 문제 해결

개요

이 섹션에서는 BeeGFS HA 클러스터를 운영할 때 발생할 수 있는 다양한 장애 및 기타 시나리오를 조사하고 해결하는 방법을 설명합니다.

문제 해결 설명서

예상치 못한 장애 조치 조사

노드가 예기치 않게 펜싱되고 해당 서비스가 다른 노드로 이동된 경우 첫 번째 단계는 클러스터가 하단에 리소스 장애를 나타내는지 확인하는 것입니다 `pcs status`. 펜싱이 성공적으로 완료되고 리소스가 다른 노드에서 다시 시작된 경우에는 일반적으로 아무 것도 표시되지 않습니다.

일반적으로 다음 단계는 를 사용하여 시스템 로그를 검색하는 것입니다 `journalctl` 나머지 파일 노드 중 하나에서(심장박동기 로그는 모든 노드에서 동기화됨) 오류가 발생한 시간을 알고 있는 경우 장애가 발생하기 바로 직전에 검색을 시작할 수 있습니다(일반적으로 10분 이상 전에 검색을 시작하는 것이 좋습니다).

```
journalctl --since "<YYYY-MM-DD HH:MM:SS>"
```

다음 섹션에서는 조사 범위를 더욱 좁히기 위해 로그에 표시할 수 있는 일반적인 텍스트를 보여 줍니다.

조사/해결 단계

1단계**: BeeGFS** 모니터에서 장애를 감지했는지 확인합니다**.**

BeeGFS 모니터에 의해 페일오버가 트리거된 경우 오류가 표시됩니다(다음 단계로 진행되지 않는 경우).

```
journalctl --since "<YYYY-MM-DD HH:MM:SS>" | grep -i unexpected
[...]
Jul 01 15:51:03 beegfs_01 pacemaker-schedulerd[9246]:  warning: Unexpected
result (error: BeeGFS service is not active!) was recorded for monitor of
meta_08-monitor on beegfs_02 at Jul  1 15:51:03 2022
```

이 경우 BeeGFS 서비스 META_08이 어떤 이유로 중지되었습니다. 문제 해결을 계속하려면 beegfs_02를 부팅하고 에서 서비스에 대한 로그를 검토해야 합니다 `/var/log/beegfs-meta-meta_08_tgt_0801.log`. 예를 들어, BeeGFS 서비스에서 노드의 내부 문제 또는 문제로 인해 애플리케이션 오류가 발생했을 수 있습니다.

페이스 메이커의 로그와 달리 BeeGFS 서비스의 로그는 클러스터의 모든 노드에 배포되지 않습니다. 이러한 유형의 장애를 조사하려면 장애가 발생한 원래 노드의 로그가 필요합니다.

모니터에서 보고할 수 있는 문제는 다음과 같습니다.

- 대상에 액세스할 수 없습니다!
  - 설명: 블록 볼륨을 액세스할 수 없음을 나타냅니다.
  - 문제 해결:
    - 대체 파일 노드에서 서비스도 시작하지 못한 경우 블록 노드가 정상 상태인지 확인합니다.
    - 이 파일 노드에서 블록 노드에 액세스하지 못하게 하는 물리적 문제(예: InfiniBand 어댑터 또는 케이블 장애)가 있는지 확인합니다.
- 네트워크에 연결할 수 없습니다!
  - 설명: 클라이언트가 이 BeeGFS 서비스에 연결하는 데 사용하는 어댑터 중 온라인 어댑터가 없습니다.
  - 문제 해결:
    - 여러 파일 노드/모든 파일 노드에 영향을 받은 경우 BeeGFS 클라이언트 및 파일 시스템을 연결하는 데 사용된 네트워크에 장애가 있는지 확인합니다.
    - 이 파일 노드에서 클라이언트에 액세스하지 못하게 하는 물리적 문제(예: InfiniBand 어댑터 또는 케이블 장애)가 있는지 확인합니다.
- BeeGFS 서비스가 활성 상태가 아닙니다.
  - 설명: BeeGFS 서비스가 예기치 않게 중지되었습니다.
  - 문제 해결:
    - 오류를 보고한 파일 노드에서 영향을 받는 BeeGFS 서비스의 로그를 확인하여 충돌이 보고되었는지 확인합니다. 이 경우 NetApp Support에서 케이스를 접수하여 충돌을 조사하십시오.
    - BeeGFS 로그에 보고된 오류가 없는 경우 저널 로그를 확인하여 시스템이 서비스가 중지된 이유를 기록했는지 확인합니다. 일부 시나리오에서 BeeGFS 서비스는 프로세스가 종료되기 전에(예: 누군가 를 실행한 경우) 메시지를 기록할 수 있는 기회를 제공하지 않았을 수 있습니다 `kill -9 <PID>`)를 클릭합니다.

2단계**:** 노드가 예기치 않게 클러스터를 종료했는지 확인합니다

노드에 심각한 하드웨어 장애가 발생하거나(예: 시스템 보드가 작동하지 않음) 커널 패닉 또는 유사한 소프트웨어 문제가 발생한 경우 BeeGFS 모니터에서 오류를 보고하지 않습니다. 대신 호스트 이름을 찾으십시오. 심장박동기에서 노드가 예기치 않게 손실되었음을 나타내는 메시지가 표시됩니다.

```
journalctl --since "<YYYY-MM-DD HH:MM:SS>" | grep -i <HOSTNAME>
[...]
Jul 01 16:18:01 beegfs_01 pacemaker-attrd[9245]:  notice: Node beegfs_02
state is now lost
Jul 01 16:18:01 beegfs_01 pacemaker-controld[9247]:  warning:
Stonith/shutdown of node beegfs_02 was not expected
```

3단계: 심장박동기가 노드를 울타리로 만들 수 있는지 확인합니다

모든 시나리오에서 노드가 실제로 오프라인 상태인지 확인하기 위해 심장박동기 펜스(pencing)를 시도하는 것을 볼 수 있습니다(정확한 메시지는 펜싱 원인에 따라 다를 수 있음).

```
Jul 01 16:18:02 beegfs_01 pacemaker-schedulerd[9246]:  warning: Cluster
node beegfs_02 will be fenced: peer is no longer part of the cluster
Jul 01 16:18:02 beegfs_01 pacemaker-schedulerd[9246]:  warning: Node
beegfs_02 is unclean
Jul 01 16:18:02 beegfs_01 pacemaker-schedulerd[9246]:  warning: Scheduling
Node beegfs_02 for STONITH
```

펜싱 작업이 성공적으로 완료되면 다음과 같은 메시지가 표시됩니다.

```
Jul 01 16:18:14 beegfs_01 pacemaker-fenced[9243]:  notice: Operation 'off'
[2214070] (call 27 from pacemaker-controld.9247) for host 'beegfs_02' with
device 'fence_redfish_2' returned: 0 (OK)
Jul 01 16:18:14 beegfs_01 pacemaker-fenced[9243]:  notice: Operation 'off'
targeting beegfs_02 on beegfs_01 for pacemaker-
controld.9247@beegfs_01.786df3a1: OK
Jul 01 16:18:14 beegfs_01 pacemaker-controld[9247]:  notice: Peer
beegfs_02 was terminated (off) by beegfs_01 on behalf of pacemaker-
controld.9247: OK
```

어떤 이유로 펜싱 작업이 실패한 경우 데이터 손상을 방지하기 위해 다른 노드에서 BeeGFS 서비스를 다시 시작할 수 없습니다. 예를 들어 펜싱 장치(PDU 또는 BMC)에 액세스할 수 없거나 잘못 구성된 경우 별도로 조사하는 것이 문제입니다.

실패한 리소스 작업 해결**(PCS** 상태 하단에 있음**)**

BeeGFS 서비스를 실행하는 데 필요한 리소스에 장애가 발생하면 BeeGFS 모니터에서 페일오버가 트리거됩니다. 이 경우 의 하단에 "실패한 리소스 작업"이 나열되지 않을 수 `pcs status` 있으므로 방법 단계를 참조해야 "계획되지 않은 페일오버 후 페일백"합니다.

그렇지 않으면 일반적으로 "실패한 리소스 작업"이 표시되는 두 가지 시나리오만 있어야 합니다.

시나리오 **1:** 펜싱 에이전트에서 일시적 또는 영구적인 문제가 감지되어 이를 다시 시작하거나 다른 노드로 이동했습니다**.**

일부 펜싱 에이전트는 다른 펜싱 장치보다 신뢰성이 높으며 각 펜싱 장치가 준비되도록 자체 모니터링 방법을 구현합니다. 특히 Redfish 펜싱 에이전트가 여전히 started로 표시되더라도 다음과 같은 실패한 리소스 작업을 보고하는 것으로 나타났습니다.

```
  * fence_redfish_2_monitor_60000 on beegfs_01 'not running' (7):
call=2248, status='complete', exitreason='', last-rc-change='2022-07-26
08:12:59 -05:00', queued=0ms, exec=0ms
```

특정 노드에서 장애가 발생한 리소스 작업을 보고하는 펜싱 에이전트가 해당 노드에서 실행되는 BeeGFS 서비스의 페일오버를 트리거하지 않습니다. 동일한 노드 또는 다른 노드에서 자동으로 다시 시작하기만 하면 됩니다.

해결 단계:

1. 펜싱 에이전트가 노드 전체 또는 하위 집합에서 지속적으로 실행을 거부하는 경우 해당 노드가 펜싱 에이전트에 연결할 수 있는지 확인하고 펜싱 에이전트가 Ansible 인벤토리에서 올바르게 구성되었는지 확인합니다.
   a. 예를 들어, BMC(Redfish) 펜싱 에이전트가 펜싱을 담당하는 동일한 노드에서 실행되고 있고 OS 관리 및 BMC IP가 동일한 물리적 인터페이스에 있는 경우 일부 네트워크 스위치 구성에서는 두 인터페이스 간의 통신을 허용하지 않습니다(네트워크 루프 방지). 기본적으로 HA 클러스터는 펜싱을 담당하는 노드에 펜싱 에이전트를 배치하는 것을 피하려고 하지만 일부 시나리오/구성에서는 이러한 문제가 발생할 수 있습니다.
2. 모든 문제가 해결되거나 문제가 일시적인 것으로 나타나는 경우 를 실행합니다 `pcs resource cleanup` 실패한 리소스 작업을 재설정합니다.

시나리오 **2: BeeGFS** 모니터가 문제를 감지하여 페일오버를 트리거했지만**,** 어떤 이유로 보조 노드에서 리소스를 시작할 수 없습니다**.**

펜싱이 활성화되고 리소스가 원래 노드에서 정지하는 것을 차단하지 않은 경우("대기(장애 발생 시)"의 문제 해결 섹션 참조), 보조 노드에서 리소스를 시작하는 데 다음과 같은 문제가 원인일 수 있습니다.

- 보조 노드가 이미 오프라인 상태입니다.
- 물리적 또는 논리적 구성 문제로 인해 보조 시스템에서 BeeGFS 타겟으로 사용되는 블록 볼륨에 액세스하지 못했습니다.

해결 단계:

1. 실패한 리소스 작업의 각 항목에 대해 다음을 수행합니다.
   a. 실패한 리소스 작업이 시작 작업인지 확인합니다.
   b. 표시된 리소스와 실패한 리소스 작업에 지정된 노드를 기반으로 합니다.
      i. 노드가 지정된 리소스를 시작하지 못하는 외부 문제를 찾아 해결합니다. 예를 들어 BeeGFS IP 주소(부동 IP)를 시작하지 못한 경우 필요한 인터페이스 중 하나 이상이 온라인으로 연결되어 있고 올바른 네트워크 스위치에 케이블로 연결되어 있는지 확인합니다. BeeGFS 타겟(블록 디바이스/E-Series 볼륨)에 장애가 발생한 경우 백엔드 블록 노드에 대한 물리적 접속이 예상대로 접속되어 있는지 확인하고 블록 노드가 정상 상태인지 확인합니다.
   c. 명확한 외부 문제가 없고 이 인시던트에 대한 근본 원인이 필요한 경우, 다음 단계로 인해 근본 원인 분석

(RCA)이 어렵거나 불가능할 수 있으므로 계속하기 전에 NetApp Support에서 케이스를 열어 조사하는 것이 좋습니다.

2. 외부 문제 해결 후:
   a. Anabilities inventory.yml 파일에서 작동하지 않는 노드를 모두 제거하고 전체 Ansible 플레이북을 다시 실행하여 모든 논리적 구성이 보조 노드에 올바르게 설정되었는지 확인합니다.
      i. 참고: 노드 상태가 양호하고 페일백할 준비가 되면 이러한 노드의 주석을 해제하고 플레이북을 다시 실행하십시오.
   b. 또는 클러스터를 수동으로 복구할 수도 있습니다.
      i. 다음을 사용하여 오프라인 노드를 다시 온라인 상태로 전환: `pcs cluster start <HOSTNAME>`
      ii. 다음을 사용하여 실패한 모든 리소스 작업을 지웁니다. `pcs resource cleanup`
      iii. PCS 상태를 실행하고 모든 서비스가 예상대로 시작되는지 확인합니다.
      iv. 필요한 경우 실행합니다 `pcs resource relocate run` 리소스를 원하는 노드로 다시 이동하려면 (사용 가능한 경우)

일반적인 문제

BeeGFS 서비스는 요청 시 페일오버 또는 페일백을 수행하지 않습니다

- 가능성 높은 문제: * `pcs resource relocate` 실행 명령이 실행되었지만 성공적으로 완료되지 않았습니다.
- 확인 방법: * 실행 `pcs constraint --full` ID가 인 위치 제약 조건이 있는지 확인합니다 `pcs-relocate-<RESOURCE>`.
- 해결 방법: * 실행 `pcs resource relocate clear` 그런 다음 다시 실행합니다 `pcs constraint --full` 추가 구속조건이 제거되었는지 확인합니다.

펜싱이 비활성화된 경우 **PCS** 상태의 노드 중 하나에 **"STANDBY(ON-FAIL)"**가 표시됩니다

- 가능성 높은 문제: * 심장박동기가 실패한 노드에서 모든 리소스가 중지되었는지 확인할 수 없습니다.
- 해결 방법: *
  1. 실행 `pcs status` 그리고 출력 하단에 "시작"되지 않은 리소스 또는 오류가 표시되는지 확인하고 모든 문제를 해결합니다.
  2. 노드를 다시 온라인 상태로 전환하려면 다음을 수행합니다 `pcs resource cleanup --node=<HOSTNAME>`.

예기치 않은 장애 조치 후 펜싱이 활성화된 경우 **PCS** 상태에 **"started (on-fail)"**가 표시됩니다

- 가능성 높은 문제: * 장애 조치를 트리거한 문제가 발생했지만 심장박동기가 노드 펜싱되었는지 확인할 수 없었습니다. 펜싱이 잘못 구성되었거나 펜싱 에이전트(예: 네트워크에서 PDU 연결이 끊어짐)에 문제가 있기 때문에 이 문제가 발생할 수 있습니다.
- 해결 방법: *
  1. 노드의 전원이 실제로 꺼져 있는지 확인합니다.

지정하는 노드가 실제로 꺼져 있지 않지만 클러스터 서비스 또는 리소스를 실행하는 경우 데이터 손상/클러스터 장애가 발생합니다.

2. 다음을 사용하여 펜싱을 수동으로 확인합니다. `pcs stonith confirm <NODE>`

이 시점에서 서비스는 장애 조치를 완료하고 다른 정상 노드에서 다시 시작해야 합니다.

일반적인 문제 해결 작업

개별 **BeeGFS** 서비스를 다시 시작합니다

일반적으로 BeeGFS 서비스를 다시 시작해야 하는 경우(예: 구성 변경을 용이하게 함) Ansible 인벤토리를 업데이트하고 플레이북을 다시 실행하여 이 작업을 수행해야 합니다. 경우에 따라 전체 Playbook을 실행할 때까지 기다릴 필요 없이 로깅 수준을 변경하는 등 더 빠른 문제 해결을 위해 개별 서비스를 다시 시작하는 것이 좋습니다.

수동 변경 사항도 Ansible 인벤토리에 추가되지 않으면 다음 번에 Ansible 플레이북을 실행할 때 되돌릴 수 있습니다.

옵션 **1:** 시스템 **d**가 재시작을 제어했습니다

BeeGFS 서비스가 새 구성으로 제대로 재시작되지 않을 위험이 있는 경우 먼저 클러스터를 유지 관리 모드로 전환하여 BeeGFS 모니터가 서비스를 감지하지 못하게 하고 원치 않는 페일오버를 트리거하는 것을 방지하십시오.

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

필요한 경우 에서 서비스 구성을 변경합니다 `/mnt/<SERVICE_ID>/_config/beegfs-.conf` (예: `/mnt/meta_01_tgt_0101/metadata_config/beegfs-meta.conf`) 그런 다음 systemd를 사용하여 다시 시작합니다.

```
systemctl restart beegfs-*@<SERVICE_ID>.service
```

예: `systemctl restart beegfs-meta@meta_01_tgt_0101.service`

옵션 **2:** 심장박동기 제어 재시작

새로운 구성으로 인해 서비스가 예기치 않게 중지되거나(예: 로깅 수준 변경) 유지 보수 기간에 있고 다운타임이 염려되지 않는 경우 다시 시작할 서비스에 대해 BeeGFS 모니터를 다시 시작하면 됩니다.

```
pcs resource restart <SERVICE>-monitor
```

예를 들어 BeeGFS 관리 서비스를 다시 시작하려면 `pcs resource restart mgmt-monitor`

저작권 정보

Copyright © 2026 NetApp, Inc. All Rights Reserved. 미국에서 인쇄됨 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 라이센스와 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 라이센스도 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 라이센스를 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 라이센스는 DFARS 조항 252.227-7015(b)(2014년 2월)에 명시된 권한으로 제한됩니다.

상표 정보

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