# BeeGFS-Cluster verwalten

## BeeGFS on NetApp with E-Series Storage

NetApp
January 27, 2026

# Inhalt

# BeeGFS-Cluster verwalten

## Übersicht, Schlüsselkonzepte und Terminologie

Lesen Sie, wie BeeGFS HA-Cluster nach der Implementierung verwaltet werden.

### Überblick

Dieser Abschnitt richtet sich an Cluster-Administratoren, die BeeGFS HA-Cluster nach ihrer Bereitstellung verwalten müssen. Selbst diejenigen, die mit Linux HA-Clustern vertraut sind, sollten diesen Leitfaden genau lesen, da es verschiedene Unterschiede beim Management des Clusters gibt, insbesondere im Hinblick auf die Neukonfiguration aufgrund der Verwendung von Ansible.

### Schlüsselkonzepte

Während einige dieser Konzepte auf der Hauptseite vorgestellt werden"Begriffe und Konzepte", ist es hilfreich, sie im Kontext eines BeeGFS HA-Clusters neu einzuführen:

**Cluster Node:** Ein Server, auf dem Pacemaker- und Corosync-Dienste ausgeführt und am HA-Cluster beteiligt sind.

**Datei-Node:** Ein Clusterknoten, mit dem ein oder mehrere BeeGFS-Management-, Metadaten- oder Storage-Services ausgeführt werden.

**Block-Node:** Ein Storage-System der NetApp E-Series, das Block-Storage für Datei-Nodes bereitstellt. Diese Nodes nehmen nicht am BeeGFS HA-Cluster Teil, da sie eigene Standalone-HA-Funktionen bereitstellen. Jeder Node besteht aus zwei Storage Controllern, die auf Blockebene Hochverfügbarkeit bieten.

**BeeGFS-Service:** Ein BeeGFS-Management, Metadaten- oder Speicherservice. Auf jedem Datei-Node wird ein oder mehrere Services ausgeführt, die Volumes auf dem Block-Node zum Speichern ihrer Daten verwenden.

**Baustein:** Eine standardisierte Implementierung von BeeGFS-Datei-Nodes, E-Series Block-Nodes und auf ihnen ausgeführten BeeGFS-Services zur Vereinfachung der Skalierung eines BeeGFS HA-Clusters/-Filesystems nach einer NetApp Verified Architecture. Kundenspezifische HA-Cluster werden ebenfalls unterstützt, verfolgen jedoch oft einen ähnlichen Bausteinansatz, der die Skalierung vereinfacht.

**BeeGFS HA Cluster:** Eine skalierbare Anzahl von Datei-Nodes, die für die Ausführung von BeeGFS-Diensten verwendet werden, die von Block-Nodes gesichert werden, um BeeGFS-Daten auf hochverfügbare Weise zu speichern. Basiert auf bewährten Open-Source-Komponenten Pacemaker und Corosync mit Ansible für Verpackung und Bereitstellung.

**Cluster Services:** bezieht sich auf Pacemaker- und Corosync-Dienste, die auf jedem Knoten ausgeführt werden, der am Cluster teilnimmt. Hinweis: Es ist möglich, dass ein Node keine BeeGFS-Services ausführen kann und nur als „Tiebreaker“-Node im Cluster teilnimmt, wenn es nur zwei Datei-Nodes benötigt.

**Cluster-Ressourcen:** für jeden BeeGFS-Dienst, der im Cluster ausgeführt wird, wird eine BeeGFS-Monitorressource und eine Ressourcengruppe mit Ressourcen für BeeGFS-Ziele, IP-Adressen (fließende IPs) und den BeeGFS-Service selbst angezeigt.

**Ansible:** Ein Tool für die Softwarebereitstellung, das Konfigurationsmanagement und den Applikationseinsatz, das Infrastruktur als Code ermöglicht. Die Pakete von BeeGFS HA-Clustern vereinfachen die Bereitstellung, Neukonfiguration und Aktualisierung von BeeGFS auf NetApp.

**Stk:** Eine Befehlszeilenoberfläche, die von einem der Dateiknoten im Cluster zur Abfrage und Kontrolle des Status von Knoten und Ressourcen im Cluster verfügbar ist.

### Allgemeine Terminologie

**Failover:** jeder BeeGFS-Dienst hat einen bevorzugten Dateiknoten, auf dem er ausgeführt wird, es sei denn, der Knoten schlägt fehl. Wenn ein BeeGFS-Service auf einem nicht-bevorzugten/sekundären Dateiknoten ausgeführt wird, muss er sich im Failover befinden.

**Failback:** der Akt, BeeGFS-Dienste von einem nicht bevorzugten Dateiknoten zurück zu ihrem bevorzugten Knoten zu verschieben.

**HA-Paar:** zwei Datei-Nodes, die auf den gleichen Satz von Block-Nodes zugreifen können, werden manchmal als HA-Paar bezeichnet. Dieser Begriff wird von NetApp häufig verwendet, um zwei Storage-Controller oder Nodes zu bezeichnen, die untereinander „übernommen“ können.

**Wartungsmodus:** deaktiviert die gesamte Ressourcenüberwachung und verhindert, dass Pacemaker Ressourcen im Cluster verschieben oder anderweitig verwalten kann (siehe auch den Abschnitt auf "Wartungsmodus").

**HA-Cluster:** ein oder mehrere Dateiknoten mit BeeGFS-Diensten, die ein Failover zwischen mehreren Knoten im Cluster ausführen können, um ein hochverfügbares BeeGFS-Dateisystem zu erstellen. Häufig sind Datei-Nodes in HA-Paaren konfiguriert, die in der Lage sind, eine Untergruppe der BeeGFS-Dienste im Cluster auszuführen.

# Wann Ansible im Vergleich zum Tool PCs verwendet werden soll

## Wann sollten Sie Ansible im Vergleich zum PCs-Befehlszeilungstool für das Management des HA-Clusters verwenden?

Alle Cluster-Implementierungs- und Neukonfigurierungsaufgaben sollten mit Ansible von einem externen Ansible-Kontroll-Node abgeschlossen werden. Temporäre Änderungen im Clusterstatus (z. B. ein- und Ausstellen von Knoten in den Standby-Modus) werden in der Regel durch Anmeldung an einem Knoten des Clusters (vorzugsweise einer, der nicht beeinträchtigt ist oder sich über die Wartung befindet) und unter Verwendung des Befehlszeilen-Tools PCs durchgeführt.

Das Ändern einer beliebigen Cluster-Konfiguration einschließlich Ressourcen, Einschränkungen, Eigenschaften und der BeeGFS Services selbst sollte immer mit Ansible erfolgen. Das Verwalten einer aktuellen Kopie des Ansible-Bestands und Playbook (ideal zur Versionskontrolle, um Änderungen zu verfolgen) ist Teil der Wartung des Clusters. Wenn Sie Änderungen an der Konfiguration vornehmen müssen, aktualisieren Sie den Bestand und führen Sie das Ansible-Playbook aus, das die BeeGFS HA-Rolle importiert.

Die HA-Rolle verarbeitet, das Cluster in den Wartungsmodus zu platzieren und anschließend alle erforderlichen Änderungen vorzunehmen, bevor BeeGFS oder Cluster-Services neu gestartet werden, um die neue Konfiguration anzuwenden. Da in der Regel keine vollständigen Node-Neustarts außerhalb der ursprünglichen Implementierung erforderlich sind, wird das Rerunning von Ansible in der Regel als „sicheres“ Verfahren angesehen. Für den Fall, dass BeeGFS-Services neu gestartet werden müssen, wird jedoch immer während Wartungsfenster oder außerhalb der Geschäftszeiten empfohlen. Diese Neustarts sollten in der Regel keine Anwendungsfehler verursachen, können aber die Leistung beeinträchtigen (was einige Anwendungen besser verarbeiten können als andere).

Die erneute Ausführung von Ansible ist auch eine Option, wenn Sie den gesamten Cluster wieder in einen

vollkommen optimalen Zustand zurückversetzen möchten, und kann in einigen Fällen den Status des Clusters einfacher wiederherstellen als PCs. Insbesondere in einem Notfall, in dem der Cluster aus irgendeinem Grund ausgefallen ist, kann, wenn alle Knoten gesichert werden, Ansible neu zu starten, den Cluster schneller und zuverlässiger wiederherstellen, als zu versuchen, PCs zu verwenden.

# Untersuchen Sie den Status des Clusters

Verwenden Sie PCs, um den Status des Clusters anzuzeigen.

## Überblick

Wird Ausgeführt `pcs status` Von jedem Cluster-Node aus können Sie den Gesamtstatus des Clusters und den Status jeder Ressource (z. B. BeeGFS-Services und deren Abhängigkeiten) am einfachsten einsehen. In diesem Abschnitt wird erklärt, was Sie in der Ausgabe von finden `pcs status` Befehl.

## Allgemeines zur Ausgabe von `pcs status`

Laufen `pcs status` Auf jedem Clusterknoten, auf dem die Cluster-Dienste (Pacemaker und Corosync) gestartet werden. Oben in der Ausgabe wird eine Zusammenfassung des Clusters angezeigt:

```
[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
```

Im folgenden Abschnitt werden Nodes im Cluster aufgeführt:

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

Dies zeigt insbesondere alle Knoten an, die sich im Standby- oder Offline-Modus befinden. Nodes im Standby-Modus sind weiterhin am Cluster beteiligt, sind jedoch als nicht zur Ausführung von Ressourcen geeignet. Nodes, die offline sind, geben an, dass auf diesem Node keine Cluster-Services ausgeführt werden, entweder da sie manuell angehalten werden, oder weil der Node neu gebootet/heruntergefahren wurde.

Beim ersten Starten von Nodes werden Cluster-Services angehalten und müssen manuell gestartet werden, um zu vermeiden, dass versehentlich Ressourcen auf einen nicht funktionsuntüchtigen Node zurückfallen.

Wenn sich Knoten aufgrund eines nicht-administrativen Grund im Standby- oder Offline-Modus befinden (zum Beispiel ein Ausfall), wird neben dem Status des Node in Klammern zusätzlicher Text angezeigt. Wenn beispielsweise das Fechten deaktiviert ist und eine Ressource auf einen Fehler stößt, wird angezeigt `Node <HOSTNAME>: standby (on-fail)`. Ein anderer möglicher Zustand ist `Node <HOSTNAME>: UNCLEAN (offline)`, Die kurz als ein Knoten angezeigt wird, wird eingezäunt, aber bleibt bestehen, wenn das Fechten fehlgeschlagen zeigt, dass der Cluster den Status des Knotens nicht bestätigen kann (dies kann verhindern, dass die Ressourcen auf anderen Knoten beginnen).

Im nächsten Abschnitt werden alle Ressourcen im Cluster und ihre Status angezeigt:

```
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
[...]
```

Ähnlich wie bei Knoten wird neben dem Ressourcenzustand in Klammern zusätzlicher Text angezeigt, wenn Probleme mit der Ressource auftreten. Wenn z. B. Pacemaker einen Ressourcenstopp anfordert und dieser nicht innerhalb der zugewiesenen Zeit abgeschlossen werden kann, versucht Pacemaker, den Knoten einzuzäunen. Wenn das Fechten deaktiviert ist oder der Fechten-Vorgang fehlschlägt, wird der Ressourcenzustand angezeigt `FAILED <HOSTNAME> (blocked)` Pacemaker kann ihn nicht auf einem anderen Knoten starten.

Es ist erwähnenswert BeeGFS HA-Cluster nutzen eine Reihe von BeeGFS optimiert benutzerdefinierte OCF-Ressourcen-Agenten. Insbesondere ist der BeeGFS-Monitor für das Auslösen eines Failover verantwortlich, wenn BeeGFS-Ressourcen auf einem bestimmten Knoten nicht verfügbar sind.

# Konfigurieren Sie HA-Cluster und BeeGFS neu

Verwenden Sie Ansible, um das Cluster neu zu konfigurieren.

## Überblick

Generell sollten Sie jeden Aspekt des BeeGFS HA-Clusters neu konfigurieren, indem Sie Ihren Ansible-Bestand aktualisieren und den `ansible-playbook` Befehl erneut ausführen. Dazu gehören das Aktualisieren von Warnungen, das Ändern der Konfiguration für permanente Fechten oder das Anpassen der BeeGFS-Servicekonfiguration. Diese werden über die `group_vars/ha_cluster.yml` Datei angepasst und eine vollständige Liste der Optionen finden Sie im "Festlegen Der Konfiguration Des Gemeinsamen Dateiknotens" Abschnitt.

Weitere Informationen zu ausgewählten Konfigurationsoptionen finden Sie unten, die Administratoren bei der Wartung oder Wartung des Clusters beachten sollten.

## So deaktivieren und aktivieren Sie Fechten

Beim Einrichten des Clusters ist Fechten standardmäßig aktiviert/erforderlich. In einigen Fällen ist es wünschenswert, Fechten vorübergehend zu deaktivieren, um sicherzustellen, dass Knoten nicht versehentlich heruntergefahren werden, wenn bestimmte Wartungsvorgänge ausgeführt werden (z. B. ein Upgrade des Betriebssystems). Auch wenn dies manuell deaktiviert werden kann, sollte es auf die Kompromisse-Administratoren achten.

**OPTION 1: Deaktivieren Sie Fechten mit Ansible (empfohlen).**

Wenn das Fechten mit Ansible deaktiviert wird, wird die on-Fail-Aktion des BeeGFS-Monitors von „Zaun“ in „Standby“ geändert. Wenn der BeeGFS-Monitor einen Fehler erkennt, versucht er, den Knoten in den Standby-Modus zu stellen und alle BeeGFS-Dienste zu ausfallsicher. Außerhalb aktiver Fehlerbehebung/Tests ist dies in der Regel wünschenswerter als Option 2. Der Nachteil ergibt sich daraus, dass eine Ressource auf dem ursprünglichen Knoten nicht stoppt, dass sie an einem anderen Ort gestartet werden kann (weshalb normalerweise ein Fechten für Produktionscluster erforderlich ist).

1. In Ihrem Ansible-Inventar unter `groups_vars/ha_cluster.yml` Fügen Sie die folgende Konfiguration hinzu:

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

2. Führen Sie das Ansible-Playbook erneut aus, um die Änderungen auf das Cluster anzuwenden.

**OPTION 2: Manuelle Abwahl deaktivieren.**

In einigen Fällen möchten Sie die Fechten unter Umständen vorübergehend deaktivieren, ohne Ansible neu zu verwenden, um die Fehlerbehebung oder das Testen des Clusters zu erleichtern.

> Wenn der BeeGFS-Monitor in dieser Konfiguration einen Fehler erkennt, versucht das Cluster, die entsprechende Ressourcengruppe zu stoppen. Es wird KEIN vollständiger Failover ausgelöst oder versucht, die betroffene Ressourcengruppe auf einen anderen Host neu zu starten oder zu verschieben. Zur Wiederherstellung sollten Sie alle Probleme beheben und anschließend ausführen `pcs resource cleanup` Oder setzen Sie den Knoten manuell in den Standby-Modus.

Schritte

1. So legen Sie fest, ob Fechten (stonith) global aktiviert oder deaktiviert ist: `pcs property show stonith-enabled`
2. So deaktivieren Sie den Fechtlauf: `pcs property set stonith-enabled=false`
3. So aktivieren Sie den Fechtlauf: `pcs property set stonith-enabled=true`

> Diese Einstellung wird beim nächsten Ausführen des Ansible-Playbooks überschrieben.

# Aktualisieren Sie die HA-Cluster-Komponenten

## Upgrade der BeeGFS Services

Verwenden Sie Ansible, um die BeeGFS-Version, die auf Ihrem HA-Cluster läuft, zu aktualisieren.

### Überblick

BeeGFS folgt einem `major.minor.patch` Versionsschema. Die BeeGFS HA-Ansible-Rollen werden für jede unterstützte `major.minor` Version (z. B. `beegfs_ha_7_2` und `beegfs_ha_7_3`) bereitgestellt. Jede HA-Rolle ist auf die neueste BeeGFS-Patch-Version fixiert, die zum Zeitpunkt der Veröffentlichung der Ansible Sammlung verfügbar ist.

Ansible sollte für alle BeeGFS-Upgrades verwendet werden, einschließlich des Wechsels zwischen Haupt-, Neben- und Patch-Versionen von BeeGFS. Um BeeGFS zu aktualisieren, müssen Sie zunächst die BeeGFS Ansible Collection aktualisieren, wodurch auch die neuesten Korrekturen und Verbesserungen für die Bereitstellungs-/Verwaltungsautomatisierung und den zugrunde liegenden HA-Cluster übernommen werden. Selbst nach der Aktualisierung auf die neueste Version der Collection wird BeeGFS erst aktualisiert, wenn `ansible-playbook` mit dem `-e "beegfs_ha_force_upgrade=true"`-Set ausgeführt wird. Weitere Details zu jedem Upgrade finden Sie in der "BeeGFS Upgrade-Dokumentation" für Ihre aktuelle Version.

> Wenn Sie auf BeeGFS v8 aktualisieren, beachten Sie stattdessen das "Upgrade auf BeeGFS v8" Verfahren.

### Getestete Upgrade-Pfade

Die folgenden Upgrade-Pfade wurden getestet und verifiziert:

| Originalversion | Upgrade-Version | Multirail | Details |
|---|---|---|---|
| 7.2.6 | 7.3.2 | Ja. | Beegfs-Sammlung von v3.0.1 auf v3.1.0, multirail hinzugefügt |
| 7.2.6 | 7.2.8 | Nein | Beegfs-Sammlung wird von v3.0.1 auf v3.1 aktualisiert |
| 7.2.8 | 7.3.1 | Ja. | Upgrade mit beegfs Collection v3.1.0, multirail hinzugefügt |
| 7.3.1 | 7.3.2 | Ja. | Upgrade mit beegfs Collection v3.1.0 |
| 7.3.2 | 7.4.1 | Ja. | Upgrade mit beegfs Collection v3.2.0 |
| 7.4.1 | 7.4.2 | Ja. | Upgrade mit beegfs Collection v3.2.0 |
| 7.4.2 | 7.4.6 | Ja. | Upgrade mit beegfs Collection v3.2.0 |
| 7.4.6 | 8,0 | Ja. | Führen Sie das Upgrade gemäß den Anweisungen in der "Upgrade auf BeeGFS v8"-Prozedur durch. |
| 7.4.6 | 8,1 | Ja. | Führen Sie das Upgrade gemäß den Anweisungen in der "Upgrade auf BeeGFS v8"-Prozedur durch. |
| 7.4.6 | 8,2 | Ja. | Führen Sie das Upgrade gemäß den Anweisungen in der "Upgrade auf BeeGFS v8"-Prozedur durch. |

### Schritte beim BeeGFS-Upgrade

In den folgenden Abschnitten werden die Schritte zum Aktualisieren der BeeGFS Ansible Sammlung und BeeGFS selbst beschrieben. Achten Sie besonders auf zusätzliche Schritte für die Aktualisierung von BeeGFS

Major oder Minor Versionen.

#### Schritt: Upgrade der BeeGFS-Sammlung

Bei Erfassungs-Upgrades mit Zugriff auf "Ansible-Galaxie", Ausführen des folgenden Befehls:

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

Laden Sie die Sammlung von herunter, um Offline-Sammlungs-Upgrades von zu erhalten "Ansible-Galaxie" Durch Klicken auf das gewünschte `Install Version` Und dann `Download tarball`. Übertragen Sie den Tarball auf Ihren Ansible-Steuerungsknoten und führen Sie den folgenden Befehl aus.

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

Siehe "Sammlungen Werden Installiert" Finden Sie weitere Informationen.

#### Schritt 2: Aktualisieren Sie den Ansible-Bestand

Nehmen Sie alle erforderlichen oder gewünschten Aktualisierungen an den Ansible-Inventardateien Ihres Clusters vor. Siehe den Hinweise zur Versionsaktualisierung Abschnitt unten für Details zu Ihren spezifischen Upgrade-Anforderungen. Siehe den "Ansible-Bestandsübersicht" Abschnitt für allgemeine Informationen zur Konfiguration Ihres BeeGFS HA-Inventars.

#### Schritt 3: Ansible-Playbook aktualisieren (nur bei Aktualisierung von Haupt- oder Nebenversionen)

Wenn Sie zwischen Haupt- oder Unterversionen wechseln, aktualisieren Sie in der `playbook.yml` Datei, die zum Bereitstellen und Warten des Clusters verwendet wird, den Namen der `beegfs_ha_<VERSION>` Rolle, damit die gewünschte Version angezeigt wird. Wenn Sie beispielsweise BeeGFS 7.4 bereitstellen möchten, wäre dies `beegfs_ha_7_4`:

```
- 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
```

Weitere Informationen zum Inhalt dieser Playbook-Datei finden Sie im "Implementieren Sie das BeeGFS HA-Cluster" Abschnitt.

#### Schritt 4: Führen Sie das BeeGFS-Upgrade aus

So wenden Sie das BeeGFS-Update an:

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

Hinter den Kulissen übernimmt die BeeGFS HA-Rolle:

- Stellen Sie sicher, dass sich das Cluster in einem optimalen Zustand befindet, wobei sich jeder BeeGFS-Service auf seinem bevorzugten Node befindet.
- Versetzen Sie das Cluster in den Wartungsmodus.
- Aktualisieren der HA-Cluster-Komponenten (falls erforderlich)
- Aktualisieren Sie jeden Dateiknoten nacheinander wie folgt:
  - Setzen Sie ihn in den Standby-Modus und führen Sie ein Failover seiner Dienste zum sekundären Knoten durch.
  - BeeGFS-Pakete aktualisieren.
  - Fallback-Services.
- Verschieben Sie das Cluster aus dem Wartungsmodus.

### Hinweise zur Versionsaktualisierung

#### Upgrade von BeeGFS Version 7.2.6 oder 7.3.0

### Änderungen an verbindungsbasierter Authentifizierung

BeeGFS Version 7.3.2 und höher erfordert, dass eine verbindungsbasierte Authentifizierung konfiguriert wird. Dienste werden ohne eine der folgenden Optionen nicht gestartet:

- Angabe eines `connAuthFile`, oder
- Einstellung `connDisableAuthentication=true` in der Konfigurationsdatei des Dienstes.

Es wird dringend empfohlen, die verbindungsbasierte Authentifizierung aus Sicherheitsgründen zu aktivieren. Siehe "BeeGFS-Verbindungsbasierte Authentifizierung" für weitere Informationen.

Die `beegfs_ha*`Rollen generieren und verteilen die Authentifizierungsdatei an:

- Alle Dateiknoten im Cluster
- Der Ansible control node bei `<playbook_directory>/files/beegfs/<beegfs_mgmt_ip_address>_connAuthFile`

Die `beegfs_client` Rolle erkennt diese Datei automatisch und wendet sie auf Clients an, wenn sie vorhanden ist.

Wenn Sie die `beegfs_client` Rolle nicht zur Konfiguration der Clients verwendet haben, müssen Sie die Authentifizierungsdatei manuell an jeden Client verteilen und die `connAuthFile` Einstellung in der `beegfs-client.conf` Datei konfigurieren. Beim Upgrade von einer BeeGFS-Version ohne verbindungsbasierte Authentifizierung verlieren Clients den Zugriff, es sei denn, Sie deaktivieren die verbindungsbasierte Authentifizierung während des Upgrades, indem Sie `beegfs_ha_conn_auth_enabled: false` in `group_vars/ha_cluster.yml` setzen (nicht empfohlen).

Weitere Details und alternative Konfigurationsoptionen finden Sie im Schritt zur Authentifizierung der Verbindungskonfiguration im "Festlegen Der Konfiguration Des Gemeinsamen Dateiknotens" Abschnitt.

## Upgrade auf BeeGFS v8

Führen Sie diese Schritte aus, um Ihren BeeGFS HA Cluster von Version 7.4.6 auf BeeGFS v8 zu aktualisieren.

### Überblick

BeeGFS v8 führt mehrere bedeutende Änderungen ein, die vor dem Upgrade von BeeGFS v7 zusätzliche Konfigurationen erfordern. Dieses Dokument führt Sie durch die Vorbereitung Ihres Clusters auf die neuen Anforderungen von BeeGFS v8 und anschließend durch das Upgrade auf BeeGFS v8.

Vor dem Upgrade auf BeeGFS v8 stellen Sie sicher, dass auf Ihrem System mindestens BeeGFS 7.4.6 ausgeführt wird. Jeder Cluster, auf dem eine Version vor BeeGFS 7.4.6 ausgeführt wird, muss zuerst "Upgrade auf Version 7.4.6" bevor Sie mit diesem Upgrade-Verfahren für BeeGFS v8 fortfahren.

### Wichtige Änderungen in BeeGFS v8

BeeGFS v8 führt die folgenden wesentlichen Änderungen ein:

- **Lizenzbestimmungen:** BeeGFS v8 erfordert eine Lizenz für die Nutzung von Premium-Funktionen wie Speicherpools, Remote-Speicherzielen, BeeOND und mehr. Erwerben Sie vor dem Upgrade eine gültige Lizenz für Ihren BeeGFS-Cluster. Bei Bedarf können Sie eine temporäre BeeGFS v8-Evaluierungslizenz von dem "BeeGFS License Portal" erhalten.
- **Migration der Management-Service-Datenbank:** Um die Konfiguration mit dem neuen TOML-basierten Format in BeeGFS v8 zu ermöglichen, müssen Sie Ihre BeeGFS v7 Management-Service-Datenbank in das aktualisierte BeeGFS v8-Format migrieren.
- **TLS-Verschlüsselung:** BeeGFS v8 führt TLS für die sichere Kommunikation zwischen Diensten ein. Sie müssen TLS-Zertifikate für den BeeGFS-Verwaltungsdienst und das `beegfs` Befehlszeilenprogramm im Rahmen des Upgrades generieren und verteilen.

Weitere Details und zusätzliche Änderungen in BeeGFS 8 finden Sie in der "BeeGFS v8.0.0 Upgrade-Leitfaden".

Für das Upgrade auf BeeGFS v8 ist eine Ausfallzeit des Clusters erforderlich. Darüber hinaus können BeeGFS v7 Clients keine Verbindung zu BeeGFS v8 Clustern herstellen. Koordinieren Sie die Upgrade-Zeitpunkte zwischen dem Cluster und den Clients sorgfältig, um die Auswirkungen auf den Betrieb zu minimieren.

### Bereiten Sie Ihren BeeGFS Cluster auf das Upgrade vor

Bereiten Sie Ihre Umgebung sorgfältig vor, bevor Sie mit dem Upgrade beginnen, um einen reibungslosen Übergang zu gewährleisten und Ausfallzeiten zu minimieren.

1. Stellen Sie sicher, dass sich Ihr Cluster in einem fehlerfreien Zustand befindet und alle BeeGFS-Dienste auf ihren bevorzugten Knoten ausgeführt werden. Überprüfen Sie von einem Dateiknoten, auf dem BeeGFS-Dienste ausgeführt werden, ob alle Pacemaker-Ressourcen auf ihren bevorzugten Knoten ausgeführt werden.

   ```
   pcs status
   ```

2. Zeichnen Sie Ihre Clusterkonfiguration auf und sichern Sie sie.
   a. Siehe die "BeeGFS Backup-Dokumentation" für Anweisungen zum Sichern Ihrer Clusterkonfiguration.
   b. Sichern Sie das bestehende Verwaltungsdatenverzeichnis:

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

   c. Führen Sie die folgenden Befehle von einem beegfs client aus und speichern Sie deren Ausgabe zur Referenz:

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

   d. Wenn Sie die Spiegelung verwenden, erfassen Sie detaillierte Statusinformationen:

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

3. Bereiten Sie Ihre Kunden auf Ausfallzeiten vor und stoppen Sie `beegfs-client` Dienste. Führen Sie für jeden Kunden aus:

   ```
   systemctl stop beegfs-client
   ```

4. Deaktivieren Sie für jeden Pacemaker-Cluster STONITH. Dadurch können Sie die Integrität des Clusters nach dem Upgrade überprüfen, ohne unnötige Knotenneustarts auszulösen.

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

5. Für alle Pacemaker-Cluster im BeeGFS-Namespace verwenden Sie PCS, um den Cluster zu stoppen:

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

### Aktualisieren Sie die BeeGFS-Pakete

Fügen Sie auf allen Dateiknoten im Cluster das BeeGFS v8-Paket-Repository für Ihre Linux-Distribution hinzu. Anweisungen zur Verwendung der offiziellen BeeGFS-Repositories finden Sie unter "BeeGFS Download-Seite". Andernfalls konfigurieren Sie Ihr lokales BeeGFS-Mirror-Repository entsprechend.

Die folgenden Schritte beschreiben die Vorgehensweise anhand des offiziellen BeeGFS 8.2 Repository auf RHEL 9 Dateiknoten. Führen Sie die folgenden Schritte auf allen Dateiknoten im Cluster aus:

1. Importieren Sie den BeeGFS GPG-Schlüssel:

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

2. Importieren Sie das BeeGFS repository:

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

> Entfernen Sie alle zuvor konfigurierten BeeGFS-Repositories, um Konflikte mit dem neuen BeeGFS v8-Repository zu vermeiden.

3. Leeren Sie den Cache Ihres Paketmanagers:

```
dnf clean all
```

4. Aktualisieren Sie auf allen Dateiknoten die BeeGFS-Pakete auf BeeGFS 8.2.

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

> In einem Standardcluster wird das `beegfs-mgmtd` Paket nur auf den ersten beiden Dateiknoten aktualisiert.

### Aktualisieren Sie die Management-Datenbank

Führen Sie auf einem der Dateiknoten, auf denen der BeeGFS-Managementdienst ausgeführt wird, die folgenden Schritte durch, um die Management-Datenbank von BeeGFS v7 auf v8 zu migrieren.

1. Alle NVMe-Geräte auflisten und nach dem Verwaltungsziel filtern:

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

a. Beachten Sie den Gerätepfad aus der Ausgabe.

b. Binden Sie das Management-Zielgerät an den vorhandenen Management-Ziel-Mountpunkt ein (ersetzen Sie `/dev/nvmeXnY` durch Ihren Gerätepfad):

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

2. Importieren Sie Ihre BeeGFS 7-Verwaltungsdaten in das neue Datenbankformat, indem Sie Folgendes ausführen:

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

Erwartete Ausgabe:

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

> Der automatische Import kann in einigen Fällen aufgrund strengerer Validierungsanforderungen in BeeGFS v8 fehlschlagen. Wenn beispielsweise Ziele nicht existierenden Speicherpools zugewiesen werden, schlägt der Import fehl. Wenn die Migration fehlschlägt, führen Sie das Upgrade nicht durch. Wenden Sie sich an den NetApp Support, um Unterstützung bei der Behebung der Datenbankmigration-Probleme zu erhalten. Als Übergangslösung können Sie die BeeGFS v8 Pakete downgraden und BeeGFS v7 weiterhin verwenden, während das Problem behoben wird.

3. Verschieben Sie die generierte SQLite-Datei auf den Management-Service-Mount:

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

4. Verschieben Sie die generierte `beegfs-mgmtd.toml` auf den Mountpunkt des Verwaltungsdienstes:

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

Die Vorbereitung der `beegfs-mgmtd.toml` Konfigurationsdatei erfolgt nach Abschluss der Lizenzierungs- und TLS-Konfigurationsschritte in den nächsten Abschnitten.

### Lizenzierung konfigurieren

1. Installieren Sie die beegfs-Lizenzpakete auf allen Knoten, auf denen der beegfs-Managementdienst ausgeführt wird. Dies sind typischerweise die ersten beiden Knoten des Clusters:

```
dnf install libbeegfs-license
```

2. Laden Sie Ihre BeeGFS v8-Lizenzdatei auf die Management-Knoten herunter und platzieren Sie sie unter:

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

### TLS-Verschlüsselung konfigurieren

BeeGFS v8 erfordert TLS-Verschlüsselung für die sichere Kommunikation zwischen Verwaltungsdiensten und Clients. Es gibt drei Optionen, die TLS-Verschlüsselung für die Netzwerkkommunikation zwischen Verwaltungsdiensten und Clientdiensten zu konfigurieren. Die empfohlene und sicherste Methode ist die Verwendung von Zertifikaten, die von einer vertrauenswürdigen Zertifizierungsstelle signiert wurden. Alternativ können Sie eine eigene lokale Zertifizierungsstelle erstellen, um Zertifikate für Ihren BeeGFS-Cluster zu signieren. Für Umgebungen, in denen keine Verschlüsselung erforderlich ist, oder zur Fehlerbehebung kann TLS vollständig deaktiviert werden, obwohl dies nicht empfohlen wird, da dadurch sensible Informationen im Netzwerk offengelegt werden.

Bevor Sie fortfahren, befolgen Sie die Anweisungen im "TLS-Verschlüsselung für BeeGFS 8 konfigurieren" guide, um die TLS-Verschlüsselung für Ihre Umgebung einzurichten.

### Konfiguration des Update-Management-Dienstes

Bereiten Sie die BeeGFS v8 Management-Service-Konfigurationsdatei vor, indem Sie die Einstellungen manuell aus Ihrer BeeGFS v7 Konfigurationsdatei in die `/mnt/mgmt_tgt_mgmt01/mgmt_config/beegfs-mgmtd.toml` Datei übertragen.

1. Auf dem Management-Knoten, auf dem das Management-Ziel eingebunden ist, referenzieren Sie die `/mnt/mgmt_tgt_mgmt01/mgmt_config/beegfs-mgmtd.conf` Management-Service-Datei für BeeGFS 7 und übertragen Sie anschließend alle Einstellungen in die `/mnt/mgmt_tgt_mgmt01/mgmt_config/beegfs-mgmtd.toml` Datei. Für eine grundlegende Einrichtung könnte Ihre `beegfs-mgmtd.toml` wie folgt aussehen:

```
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']
```

Passen Sie alle Pfade nach Bedarf an Ihre Umgebung und TLS-Konfiguration an.

2. Ändern Sie auf jedem Dateiknoten, auf dem Verwaltungsdienste ausgeführt werden, Ihre systemd-Dienstdatei so, dass sie auf den neuen Speicherort der Konfigurationsdatei verweist.

```
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. Systemd neu laden:

```
systemctl daemon-reload
```

3. Für jeden Dateiknoten, auf dem Verwaltungsdienste ausgeführt werden, öffnen Sie Port 8010 für die gRPC-Kommunikation des Verwaltungsdienstes.

   a. Fügen Sie Port 8010/tcp zur beegfs zone hinzu:

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

   b. Laden Sie die Firewall neu, um die Änderung anzuwenden:

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

### Aktualisieren Sie das BeeGFS-Monitor-Skript

Das Pacemaker `beegfs-monitor` OCF-Skript muss aktualisiert werden, um das neue TOML-Konfigurationsformat und die systemd-Dienstverwaltung zu unterstützen. Aktualisieren Sie das Skript auf einem Knoten im Cluster und kopieren Sie das aktualisierte Skript dann auf alle anderen Knoten.

1. Erstellen Sie eine Sicherungskopie des aktuellen Skripts:

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

2. Aktualisieren Sie den Pfad der Management-Konfigurationsdatei von `.conf` zu `.toml`:

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

   Alternativ suchen Sie den folgenden Block im Skript manuell:

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

   Und ersetzen Sie es durch:

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

3. Aktualisieren Sie die `get_interfaces()` und `get_subnet_ips()` Funktionen, um die TOML-Konfiguration zu unterstützen:
   a. Öffnen Sie das Skript in einem Texteditor:

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

   b. Finden Sie die beiden Funktionen: `get_interfaces()` und `get_subnet_ips()`.
   c. Löschen Sie beide gesamten Funktionen, beginnend bei `get_interfaces()` bis zum Ende von `get_subnet_ips()`.
   d. Kopieren Sie die folgenden aktualisierten Funktionen und fügen Sie sie an ihrer Stelle ein:

```
# 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. Speichern und beenden Sie den Texteditor.

f. Führen Sie den folgenden Befehl aus, um das Skript vor der Fortsetzung auf Syntaxfehler zu überprüfen. Keine Ausgabe zeigt an, dass das Skript syntaktisch korrekt ist.

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

4. Kopieren Sie das aktualisierte `beegfs-monitor` OCF-Skript auf alle anderen Knoten im Cluster, um die Konsistenz zu gewährleisten:

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

### Den Cluster wieder online bringen

1. Sobald alle vorherigen Upgrade-Schritte abgeschlossen sind, bringen Sie das Cluster wieder online, indem Sie die BeeGFS-Dienste auf allen Knoten starten.

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

2. Überprüfen Sie, ob der `beegfs-mgmtd` Service erfolgreich gestartet wurde:

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

Die erwartete Ausgabe umfasst Zeilen wie:

```
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
```

Falls Fehler in den Journalprotokollen auftreten, überprüfen Sie die Pfade der Verwaltungskonfigurationsdatei und stellen Sie sicher, dass alle Werte korrekt aus der BeeGFS 7 Konfigurationsdatei übernommen wurden.

3. Führen Sie `pcs status` aus und überprüfen Sie, ob der Cluster fehlerfrei ist und die Dienste auf den bevorzugten Knoten gestartet wurden.
4. Sobald die einwandfreie Funktion des Clusters bestätigt ist, aktivieren Sie STONITH wieder:

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

5. Fahren Sie mit dem nächsten Abschnitt fort, um die BeeGFS-Clients im Cluster zu aktualisieren und die Gesundheit des BeeGFS-Clusters zu überprüfen.

**BeeGFS-Clients aktualisieren**

Nach erfolgreichem Upgrade Ihres Clusters auf BeeGFS v8 müssen Sie auch alle BeeGFS Clients aktualisieren.

Die folgenden Schritte beschreiben den Prozess zum Upgrade von BeeGFS Clients auf einem Ubuntu-basierten System.

1. Falls noch nicht geschehen, stoppen Sie den BeeGFS client service:

```
systemctl stop beegfs-client
```

2. Fügen Sie das BeeGFS v8-Paket-Repository für Ihre Linux-Distribution hinzu. Anweisungen zur Verwendung der offiziellen BeeGFS-Repositories finden Sie unter "^BeeGFS Download-Seite". Andernfalls konfigurieren Sie Ihr lokales BeeGFS-Mirror-Repository entsprechend.

   Die folgenden Schritte verwenden das offizielle BeeGFS 8.2 Repository auf einem Ubuntu-basierten System:

3. Importieren Sie den BeeGFS GPG-Schlüssel:

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

4. Laden Sie die Repository-Datei herunter:

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

   Entfernen Sie alle zuvor konfigurierten BeeGFS-Repositories, um Konflikte mit dem neuen BeeGFS v8-Repository zu vermeiden.

5. Aktualisieren Sie die BeeGFS client packages:

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

6. Konfigurieren Sie TLS für den Client. TLS ist für die Verwendung der BeeGFS CLI erforderlich. Beziehen Sie sich auf das "TLS-Verschlüsselung für BeeGFS 8 konfigurieren" Verfahren, um TLS auf dem Client zu konfigurieren.

7. Starten Sie den BeeGFS Client Service:

```
systemctl start beegfs-client
```

#### Überprüfen Sie das Upgrade

Nach Abschluss des Upgrades auf BeeGFS v8 führen Sie die folgenden Befehle aus, um zu überprüfen, ob das Upgrade erfolgreich war.

1. Überprüfen Sie, ob der Root-Inode demselben Metadatenknoten wie zuvor gehört. Dies sollte automatisch erfolgen, wenn Sie die `import-from-v7` Funktionalität im Verwaltungsdienst verwendet haben:

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

2. Überprüfen Sie, ob alle Knoten und Ziele online und in einwandfreiem Zustand sind:

```
beegfs health check
```

> Wenn die Überprüfung „Verfügbare Kapazität“ darauf hinweist, dass auf den Zielen nur noch wenig freier Speicherplatz vorhanden ist, können Sie die in der `beegfs-mgmtd.toml` Konfigurationsdatei definierten Schwellenwerte für den „Kapazitätspool“ so anpassen, dass sie besser zu Ihrer Umgebung passen.

## Aktualisieren Sie Pacemaker- und Corosync-Pakete in einem HA-Cluster

Führen Sie diese Schritte aus, um Pacemaker- und Corosync-Pakete in einem HA-Cluster zu aktualisieren.

#### Überblick

Durch ein Upgrade von Pacemaker und Corosync wird sichergestellt, dass der Cluster von neuen Funktionen, Sicherheits-Patches und Leistungsverbesserungen profitiert.

#### Upgrade-Ansatz

Es gibt zwei empfohlene Ansätze für das Upgrade eines Clusters: Ein rollierendes Upgrade oder eine vollständige Abschaltung des Clusters. Jeder Ansatz hat seine eigenen vor- und Nachteile. Der Aktualisierungsvorgang kann je nach Ihrer Pacemaker-Version variieren. Bestimmen Sie anhand der Dokumentation von ClusterLabs"Aktualisieren eines Pacemaker-Clusters", welche Vorgehensweise verwendet werden soll. Bevor Sie einen Upgrade-Ansatz verfolgen, müssen Sie Folgendes überprüfen:

- Die neuen Pacemaker- und Corosync-Pakete werden von der NetApp BeeGFS-Lösung unterstützt.
- Für das BeeGFS-Dateisystem und die Pacemaker-Cluster-Konfiguration sind gültige Backups vorhanden.
- Das Cluster befindet sich in einem ordnungsgemäßen Zustand.

#### Rollierendes Upgrade

Bei dieser Methode wird jeder Node aus dem Cluster entfernt, aktualisiert und anschließend wieder in das Cluster eingeführt, bis die neue Version auf allen Nodes ausgeführt wird. Dieser Ansatz sorgt für einen unterbrechungsfreien Cluster, was ideal für größere HA-Cluster ist, birgt aber auch das Risiko, dass während des Prozesses gemischte Versionen ausgeführt werden. Dieser Ansatz sollte in einem Cluster mit zwei Nodes vermieden werden.

1. Vergewissern Sie sich, dass sich das Cluster in einem optimalen Zustand befindet, wobei jeder BeeGFS-Service auf seinem bevorzugten Node ausgeführt wird. Weitere Informationen finden Sie unter "Untersuchen Sie den Status des Clusters" .
2. Platzieren Sie den Node für das Upgrade in den Standby-Modus, um alle BeeGFS-Services zu leeren (oder zu verschieben):

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

3. Überprüfen Sie, ob die Services des Node durch Ausführen von abgelaufen sind:

   ```
   pcs status
   ```

   Stellen Sie sicher, dass keine Dienste als auf dem Node im Standby gemeldet werden `Started`.

   > Je nach Clustergröße kann es Sekunden oder Minuten dauern, bis Dienste zum Schwesterknoten verschoben werden. Wenn ein BeeGFS-Dienst auf dem Schwesterknoten nicht gestartet werden kann, lesen Sie die "Leitfäden Zur Fehlerbehebung".

4. Fahren Sie das Cluster auf dem Node herunter:

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

5. Aktualisieren Sie die Pacemaker-, Corosync- und PCs-Pakete auf dem Knoten:

   > Die Befehle des Package Managers variieren je nach Betriebssystem. Die folgenden Befehle gelten für Systeme, auf denen RHEL 8 und höher ausgeführt wird.

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

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

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

6. Starten Sie die Pacemaker-Clusterdienste auf dem Knoten:

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

7. Wenn das `pcs` Paket aktualisiert wurde, authentifizieren Sie den Node erneut beim Cluster:

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

8. Überprüfen Sie, ob die Pacemaker-Konfiguration mit dem Werkzeug noch gültig `crm_verify` ist.

   Dies muss nur einmal während des Cluster-Upgrades überprüft werden.

```
crm_verify -L -V
```

9. Beenden Sie den Standby-Modus des Node:

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

10. Verschieben Sie alle BeeGFS-Services zurück auf ihren bevorzugten Node:

```
pcs resource relocate run
```

11. Wiederholen Sie die vorherigen Schritte für jeden Knoten im Cluster, bis auf allen Knoten die gewünschten Pacemaker-, Corosync- und PCs-Versionen ausgeführt werden.
12. Führen Sie abschließend den Cluster aus `pcs status`, und überprüfen Sie, ob er ordnungsgemäß ist, und der `Current DC` meldet die gewünschte Pacemaker-Version.

   Wenn der `Current DC` Bericht „Misted-Version“ meldet, wird ein Knoten im Cluster weiterhin mit der vorherigen Pacemaker-Version ausgeführt und muss aktualisiert werden. Wenn ein aktualisierter Node nicht in der Lage ist, dem Cluster erneut beizutreten, oder wenn die Ressourcen nicht gestartet werden können, prüfen Sie die Cluster-Protokolle, und lesen Sie die Pacemaker-Versionshinweise oder Benutzerhandbücher nach bekannten Upgrade-Problemen.

#### Schließen Sie den Cluster ab

Bei diesem Ansatz werden alle Cluster Nodes und Ressourcen heruntergefahren, die Nodes aktualisiert und das Cluster anschließend neu gestartet. Dieser Ansatz ist erforderlich, wenn die Pacemaker- und Corosync-Versionen keine Konfiguration mit gemischten Versionen unterstützen.

1. Vergewissern Sie sich, dass sich das Cluster in einem optimalen Zustand befindet, wobei jeder BeeGFS-Service auf seinem bevorzugten Node ausgeführt wird. Weitere Informationen finden Sie unter "Untersuchen Sie den Status des Clusters" .
2. Fahren Sie die Cluster-Software (Pacemaker und Corosync) auf allen Knoten herunter.

Je nach Cluster-Größe kann es Sekunden oder Minuten dauern, bis das gesamte Cluster angehalten wurde.

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

3. Sobald Cluster-Services auf allen Knoten heruntergefahren sind, aktualisieren Sie die Pacemaker-, Corosync- und PCs-Pakete auf jedem Knoten entsprechend Ihren Anforderungen.

Die Befehle des Package Managers variieren je nach Betriebssystem. Die folgenden Befehle gelten für Systeme, auf denen RHEL 8 und höher ausgeführt wird.

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

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

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

4. Starten Sie nach dem Upgrade aller Nodes die Cluster-Software auf allen Nodes:

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

5. Wenn das `pcs` Paket aktualisiert wurde, authentifizieren Sie jeden Node im Cluster erneut:

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

6. Führen Sie abschließend den Cluster aus `pcs status`, und überprüfen Sie, ob er in Ordnung ist, und der `Current DC` meldet die korrekte Pacemaker-Version.

Wenn der `Current DC` Bericht „Misted-Version“ meldet, wird ein Knoten im Cluster weiterhin mit der vorherigen Pacemaker-Version ausgeführt und muss aktualisiert werden.

## Aktualisiert die Datei-Node-Adapter-Firmware

Führen Sie die folgenden Schritte aus, um die ConnectX-7-Adapter des Datei-Knotens auf die neueste Firmware zu aktualisieren.

### Überblick

Um einen neuen MLNX_OFED-Treiber zu unterstützen, neue Funktionen zu aktivieren oder Fehler zu beheben, ist möglicherweise eine Aktualisierung der ConnectX-7-Adapter-Firmware erforderlich. In diesem

Handbuch wird das Dienstprogramm von NVIDIA für Adapteraktualisierungen aufgrund seiner Benutzerfreundlichkeit und Effizienz verwendet `mlxfwmanager`.

### Upgrade-Überlegungen

In diesem Handbuch werden zwei Ansätze zur Aktualisierung der ConnectX-7-Adapter-Firmware beschrieben: Ein laufendes Update und ein zwei-Knoten-Cluster-Update. Wählen Sie den passenden Aktualisierungsansatz gemäß der Clustergröße aus. Bevor Sie Firmware-Aktualisierungen durchführen, stellen Sie sicher, dass:

- Ein unterstützter MLNX_OFED-Treiber ist installiert, siehe "Technologieanforderungen erfüllt".
- Für das BeeGFS-Dateisystem und die Pacemaker-Cluster-Konfiguration sind gültige Backups vorhanden.
- Das Cluster befindet sich in einem ordnungsgemäßen Zustand.

### Vorbereitung des Firmware-Updates

Es wird empfohlen, das NVIDIA-Dienstprogramm zu verwenden `mlxfwmanager`, um die Adapter-Firmware eines Knotens zu aktualisieren, die mit dem NVIDIA-Treiber MLNX_OFED gebündelt ist. Laden Sie vor dem Starten der Updates das Firmware-Image des Adapters von herunter"Die Support-Website von NVIDIA", und speichern Sie es auf jedem Datei-Node.

Für Lenovo ConnectX-7 Adapter, verwenden Sie das `mlxfwmanager_LES` Tool, das auf der NVIDIA-Seite zur Verfügung steht"OEM-Firmware".

### Rollierender Aktualisierungsansatz

Dieser Ansatz wird für alle HA-Cluster mit mehr als zwei Nodes empfohlen. Dieser Ansatz beinhaltet die Aktualisierung der Adapter-Firmware auf einem Datei-Node, sodass das HA-Cluster Anforderungen weiterhin erfüllen kann. Allerdings wird empfohlen, um I/O-Anfragen während dieser Zeit zu vermeiden.

1. Vergewissern Sie sich, dass sich das Cluster in einem optimalen Zustand befindet, wobei jeder BeeGFS-Service auf seinem bevorzugten Node ausgeführt wird. Weitere Informationen finden Sie unter "Untersuchen Sie den Status des Clusters" .
2. Wählen Sie einen Datei-Node aus, um ihn zu aktualisieren und in den Standby-Modus zu versetzen, der alle BeeGFS-Services von diesem Node entfernt (oder verschiebt):

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

3. Überprüfen Sie, ob die Dienste des Node abgelaufen sind, indem Sie Folgendes ausführen:

   ```
   pcs status
   ```

   Vergewissern Sie sich, dass keine Services als auf dem Node im Standby-Modus melden `Started`.

   Je nach Cluster-Größe kann es Sekunden oder Minuten dauern, bis die BeeGFS-Dienste zum Schwesterknoten verschoben werden. Wenn ein BeeGFS-Dienst auf dem Schwesterknoten nicht gestartet werden kann, lesen Sie die "Leitfäden Zur Fehlerbehebung".

4. Aktualisieren Sie die Adapter-Firmware mit `mlxfwmanager`.

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

Beachten Sie `PCI Device Name` für jeden Adapter, der Firmware-Updates empfängt.

5. Setzen Sie jeden Adapter mithilfe des Dienstprogramms zurück `mlxfwreset`, um die neue Firmware anzuwenden.

> Einige Firmware-Aktualisierungen erfordern möglicherweise einen Neustart, um das Update anzuwenden. Weitere Informationen finden Sie unter"Die Einschränkungen von NVIDIA mlxfwreset". Wenn ein Neustart erforderlich ist, führen Sie einen Neustart durch, anstatt die Adapter zurückzusetzen.

a. Beenden Sie den opensm-Dienst:

```
systemctl stop opensm
```

b. Führen Sie den folgenden Befehl für jeden `PCI Device Name` zuvor genannten aus.

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

c. Starten Sie den opensm-Dienst:

```
systemctl start opensm
```

d. Starten Sie den `eseries_nvme_ib.service` .

```
systemctl restart eseries_nvme_ib.service
```

e. Überprüfen Sie, ob die Volumes des E-Series-Speicherarrays vorhanden sind.

```
multipath -ll
```

1. Führen Sie aus `ibstat`, und überprüfen Sie, ob alle Adapter mit der gewünschten Firmware-Version ausgeführt werden:

```
ibstat
```

2. Starten Sie die Pacemaker-Clusterdienste auf dem Knoten:

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

3. Beenden Sie den Standby-Modus des Node:

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

4. Verschieben Sie alle BeeGFS-Services zurück auf ihren bevorzugten Node:

```
pcs resource relocate run
```

Wiederholen Sie diese Schritte für jeden Datei-Node im Cluster, bis alle Adapter aktualisiert wurden.

**Update für Cluster mit zwei Nodes**

Dieser Ansatz wird für HA-Cluster mit nur zwei Nodes empfohlen. Dieser Ansatz ähnelt einem rollierenden Update, enthält jedoch zusätzliche Schritte zur Vermeidung von Service-Ausfallzeiten, wenn die Cluster-Services eines Node angehalten werden.

1. Vergewissern Sie sich, dass sich das Cluster in einem optimalen Zustand befindet, wobei jeder BeeGFS-Service auf seinem bevorzugten Node ausgeführt wird. Weitere Informationen finden Sie unter "Untersuchen Sie den Status des Clusters" .
2. Wählen Sie einen Datei-Node aus, um den Node zu aktualisieren und in den Standby-Modus zu versetzen, der alle BeeGFS-Services von diesem Node entfernt (oder verschiebt):

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

3. Überprüfen Sie, ob die Ressourcen des Node abgelaufen sind, indem Sie Folgendes ausführen:

```
pcs status
```

   Vergewissern Sie sich, dass keine Services als auf dem Node im Standby-Modus melden `Started`.

   Je nach Cluster-Größe kann es Sekunden oder Minuten dauern, bis BeeGFS-Dienste als auf dem Schwesternknoten melden `Started`. Wenn ein BeeGFS-Dienst nicht gestartet werden kann, lesen Sie die "Leitfäden Zur Fehlerbehebung".

4. Versetzen Sie das Cluster in den Wartungsmodus.

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

5. Aktualisieren Sie die Adapter-Firmware mit `mlxfwmanager`.

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

Beachten Sie `PCI Device Name` für jeden Adapter, der Firmware-Updates empfängt.

6. Setzen Sie jeden Adapter mithilfe des Dienstprogramms zurück `mlxfwreset`, um die neue Firmware anzuwenden.

> Einige Firmware-Aktualisierungen erfordern möglicherweise einen Neustart, um das Update anzuwenden. Weitere Informationen finden Sie unter"Die Einschränkungen von NVIDIA mlxfwreset". Wenn ein Neustart erforderlich ist, führen Sie einen Neustart durch, anstatt die Adapter zurückzusetzen.

   a. Beenden Sie den opensm-Dienst:

```
systemctl stop opensm
```

   b. Führen Sie den folgenden Befehl für jeden `PCI Device Name` zuvor genannten aus.

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

   c. Starten Sie den opensm-Dienst:

```
systemctl start opensm
```

7. Führen Sie aus `ibstat`, und überprüfen Sie, ob alle Adapter mit der gewünschten Firmware-Version ausgeführt werden:

```
ibstat
```

8. Starten Sie die Pacemaker-Clusterdienste auf dem Knoten:

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

9. Beenden Sie den Standby-Modus des Node:

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

10. Beenden Sie das Cluster aus dem Wartungsmodus.

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

11. Verschieben Sie alle BeeGFS-Services zurück auf ihren bevorzugten Node:

```
pcs resource relocate run
```

Wiederholen Sie diese Schritte für jeden Datei-Node im Cluster, bis alle Adapter aktualisiert wurden.

## Upgrade von E-Series Storage-Arrays

Führen Sie die folgenden Schritte aus, um die Komponenten des HA-Clusters des E-Series Storage-Arrays zu aktualisieren.

### Überblick

Die NetApp E-Series Storage Arrays Ihres HA Clusters mit der neuesten Firmware auf dem neuesten Stand zu halten, gewährleistet optimale Performance und verbesserte Sicherheit. Firmware-Updates für das Storage Array werden über SANtricity OS-, NVSRAM- und Festplatten-Firmware-Dateien angewendet.

> Obwohl ein Upgrade der Storage Arrays während des Online-Betriebs des HA-Clusters möglich ist, sollte das Cluster bei allen Upgrades in den Wartungsmodus versetzt werden.

### Upgrade-Schritte für Block-Nodes

Im Folgenden wird beschrieben, wie die Firmware der Storage-Arrays mithilfe der `Netapp_Eseries.Santricity` Ansible-Sammlung aktualisiert wird. Bevor Sie fortfahren, lesen "Upgrade-Überlegungen"Sie das zur Aktualisierung von E-Series Systemen.

> Ein Upgrade auf SANtricity OS 11.80 oder höhere Versionen ist nur ab 11.70.5P1 möglich. Das Speicher-Array muss vor der Anwendung weiterer Upgrades zuerst auf 11.70.5P1 aktualisiert werden.

1. Überprüfen Sie den Ansible Control-Node mithilfe der neuesten SANtricity Ansible Sammlung.
   - Bei Erfassungs-Upgrades mit Zugriff auf "Ansible-Galaxie", Ausführen des folgenden Befehls:

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

   - Laden Sie für Offline-Upgrades den Sammeltarball von herunter"Ansible-Galaxie", übertragen Sie ihn auf Ihren Steuerungsknoten und führen Sie Folgendes aus:

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

     Siehe "Sammlungen Werden Installiert" Finden Sie weitere Informationen.

2. Holen Sie sich die neueste Firmware für Ihr Speicher-Array und die Laufwerke.
   a. Laden Sie die Firmware-Dateien herunter.
      - **SANtricity OS und NVSRAM:** Navigieren "NetApp Support Website"Sie zum und laden Sie die neueste Version von SANtricity OS und NVSRAM für Ihr Speicherarray-Modell herunter.
      - **Laufwerksfirmware:** Navigieren "E-Series Festplatten-Firmware-Website"Sie zum und laden Sie die neueste Firmware für jedes Laufwerkmodell Ihres Speicherarrays herunter.
   b. Speichern Sie SANtricity OS-, NVSRAM- und Laufwerk-Firmware-Dateien im `<inventory_directory>/packages` Verzeichnis Ihres Ansible Control Node.
3. Bei Bedarf aktualisieren Sie die Ansible-Bestandsdateien Ihres Clusters, damit alle Storage-Arrays (Block-Nodes), die aktualisiert werden müssen, einbezogen werden. Weitere Informationen finden Sie im "Ansible-Bestandsübersicht" Abschnitt.
4. Stellen Sie sicher, dass sich das Cluster mit jedem BeeGFS-Service auf seinem bevorzugten Node in einem optimalen Zustand befindet. Weitere Informationen finden Sie unter "Untersuchen Sie den Status des Clusters" .
5. Versetzen Sie das Cluster gemäß den Anweisungen in in "Versetzen Sie das Cluster in den Wartungsmodus"den Wartungsmodus.
6. Erstellen Sie ein neues Ansible-Playbook mit dem Namen `update_block_node_playbook.yml`. Füllen Sie das Playbook mit den folgenden Inhalten aus und ersetzen Sie die Versionen des SANtricity Betriebssystems, des NVSRAM und der Festplatten-Firmware auf Ihren gewünschten Upgrade-Pfad:

```
- 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. Führen Sie über Ihren Ansible-Steuerungsknoten den folgenden Befehl aus, um die Updates zu starten:

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

8. Überprüfen Sie nach Abschluss des Playbook, ob sich jedes Speicher-Array in einem optimalen Zustand befindet.
9. Entfernen Sie das Cluster aus dem Wartungsmodus und überprüfen Sie, ob sich das Cluster in einem

optimalen Zustand befindet, wobei sich jeder BeeGFS-Service auf seinem bevorzugten Node befindet.

# Service und Wartung

## Failover- und Failback-Services

### BeeGFS-Services zwischen Cluster-Nodes verschieben

#### Überblick

BeeGFS-Services können ein Failover zwischen den Nodes im Cluster durchführen, um sicherzustellen, dass die Clients weiterhin auf das Filesystem zugreifen können, wenn ein Node einen Fehler aufweist, oder Sie müssen eine geplante Wartung durchführen. In diesem Abschnitt werden verschiedene Möglichkeiten beschrieben, wie Administratoren das Cluster nach der Wiederherstellung nach einem Ausfall reparieren oder Services manuell zwischen Nodes verschieben können.

#### Schritte

##### Failover und Failback

#### Failover (Geplant)

Wenn Sie einen einzelnen Datei-Node zur Wartung offline schalten müssen, möchten Sie in der Regel alle BeeGFS-Dienste von diesem Node verschieben (oder ablassen). Dies kann erreicht werden, indem zunächst der Knoten in den Standby-Modus versetzt wird:

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

Nach der Überprüfung mit `pcs status` Alle Ressourcen wurden auf dem alternativen Datei-Node neu gestartet. Sie können je nach Bedarf weitere Änderungen am Node vornehmen.

#### Failback (nach einem geplanten Failover)

Wenn Sie bereit sind, die BeeGFS-Dienste zuerst auf den bevorzugten Knoten wiederherzustellen `pcs status` Und überprüfen Sie in der „Knotenliste“, ob der Status Standby lautet. Wenn der Node neu gebootet wurde, wird er offline angezeigt, bis Sie die Cluster-Services in den Online-Modus versetzen:

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

Sobald der Node online ist, bringen Sie ihn aus dem Standby-Modus mit:

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

Schließlich verlagern alle BeeGFS-Dienste wieder auf ihre bevorzugten Knoten mit:

```
pcs resource relocate run
```

### Failback (nach einem ungeplanten Failover)

Wenn auf einem Node ein Hardware- oder ein anderer Fehler auftritt, sollte der HA-Cluster automatisch reagieren und seine Services auf einen gesunden Node verschieben. So bleibt den Administratoren Zeit für Korrekturmaßnahmen. Bevor Sie fortfahren, lesen "Fehlerbehebung"Sie den Abschnitt, um die Ursache des Failovers zu ermitteln und alle offenen Probleme zu beheben. Sobald der Knoten wieder eingeschaltet ist und sich in einem ordnungsgemäßen Zustand befindet, können Sie mit dem Failback fortfahren.

Wenn ein Node nach einem ungeplanten (oder geplanten) Neubooten gebootet wird, werden Cluster-Services nicht automatisch gestartet. Sie müssen daher den Node zuerst in den Online-Modus versetzen:

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

Bei der nächsten Bereinigung werden alle Ressourcenfehler behoben, und der Fechtverlauf des Node wird zurückgesetzt:

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

Verifizieren in `pcs status` Der Knoten ist online und in einem ordnungsgemäßen Zustand. Standardmäßig werden BeeGFS-Dienste nicht automatisch Failback durchführen, um zu vermeiden, dass Ressourcen versehentlich auf einen ungesunden Knoten zurückverschoben werden. Wenn Sie bereit sind, alle Ressourcen im Cluster wieder an die bevorzugten Nodes zurückzugeben, mit den folgenden Funktionen:

```
pcs resource relocate run
```

#### Einzelne BeeGFS-Services werden auf alternative Datei-Nodes verschoben

### Verschieben Sie einen BeeGFS-Service dauerhaft auf einen neuen Datei-Node

Wenn Sie den bevorzugten Datei-Node für einen einzelnen BeeGFS-Service dauerhaft ändern möchten, passen Sie den Ansible-Bestand an, sodass der bevorzugte Node zuerst aufgelistet wird, und führen Sie das Ansible-Playbook erneut aus.

In dieser Beispieldatei ist beegfs_01 beispielsweise `inventory.yml` der bevorzugte Datei-Node zum Ausführen des BeeGFS-Managementservice:

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

Durch eine Umkehrung des Auftrags würden die Managementservices am beegfs_02 bevorzugt werden:

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

#### Verschieben Sie einen BeeGFS-Service vorübergehend auf einen alternativen Datei-Node

Im Allgemeinen, wenn ein Knoten gerade gewartet wird, möchten Sie die Schritte [Failover und Failback](#Failover-and-Failback) verwenden, um alle Dienste von diesem Knoten weg zu verschieben.

Wenn Sie aus irgendeinem Grund einen einzelnen Service auf einen anderen Dateiknoten verschieben müssen, führen Sie:

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

> Geben Sie keine einzelnen Ressourcen oder die Ressourcengruppe an. Geben Sie immer den Namen des Monitors für den BeeGFS-Dienst an, den Sie verschieben möchten. Um zum Beispiel den BeeGFS-Managementdienst auf beegfs_02 zu verschieben, führen Sie: Aus `pcs resource move mgmt-monitor beegfs_02`. Dieser Prozess kann wiederholt werden, um einen oder mehrere Services von den bevorzugten Nodes weg zu verschieben. Überprüfen Sie, ob `pcs status` die Services auf dem neuen Node verlegt/gestartet wurden.

Wenn Sie einen BeeGFS-Service wieder auf den bevorzugten Node verschieben möchten, löschen Sie zuerst die temporären Ressourcenbeschränkungen (diesen Schritt wird bei mehreren Services wiederholt):

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

Wenn Sie bereit sind, den Service(s) dann wieder zurück zu den bevorzugten Knoten zu verschieben, werden die folgenden Aktionen ausgeführt:

```
pcs resource relocate run
```

Hinweis: Mit diesem Befehl werden Services verschoben, bei denen keine temporären Ressourcenbeschränkungen mehr vorhanden sind, die sich nicht auf den bevorzugten Nodes befinden.

## Versetzen Sie das Cluster in den Wartungsmodus

Verhindern Sie, dass das HA-Cluster versehentlich auf geplante Änderungen in der Umgebung reagiert.

#### Überblick

Wenn Sie das Cluster in den Wartungsmodus versetzen, werden die gesamte Ressourcenüberwachung deaktiviert und Pacemaker kann nicht mehr Ressourcen im Cluster verschieben oder anderweitig verwalten. Alle Ressourcen werden auf den ursprünglichen Nodes weiterhin ausgeführt, unabhängig davon, ob es eine

temporäre Ausfallbedingung gibt, die den Zugriff auf sie verhindern würde. Dies wird empfohlen/ist u. a.:

- Netzwerkwartung, die vorübergehend Verbindungen zwischen Datei-Nodes und BeeGFS-Diensten unterbrechen kann.
- Block-Node-Upgrades:
- Dateiknoten-Betriebssystem, Kernel oder andere Paketaktualisierungen.

Im Allgemeinen ist der einzige Grund, das Cluster manuell in den Wartungsmodus zu versetzen, um zu verhindern, dass es auf externe Änderungen in der Umgebung reagiert. Wenn für einen einzelnen Node im Cluster die physische Reparatur erforderlich ist, verwenden Sie keinen Wartungsmodus und platzieren Sie den Node einfach gemäß dem oben beschriebenen Verfahren in Standby. Beachten Sie, dass bei der Umleitung von Ansible der Cluster automatisch der Wartungsmodus für die meisten Softwarewartungsarbeiten einschließlich Upgrades und Konfigurationsänderungen durchgeführt wird.

**Schritte**

So überprüfen Sie, ob das Cluster sich im Wartungsmodus befindet:

```
pcs property config
```

Die `maintenance-mode` Eigenschaft wird nicht angezeigt, wenn das Cluster ordnungsgemäß ausgeführt wird. Wenn sich der Cluster derzeit im Wartungsmodus befindet, wird die Eigenschaft als gemeldet `true`. Um den Wartungsmodus zu aktivieren, führen Sie folgende Schritte aus:

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

Sie können überprüfen, indem Sie den PC-Status ausführen und sicherstellen, dass alle Ressourcen „(nicht verwaltet)“ anzeigen. Um das Cluster aus dem Wartungsmodus zu nehmen, führen Sie folgende Schritte aus:

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

## Beenden Sie den Cluster und starten Sie den Cluster

Graziös wird das HA-Cluster angehalten und gestartet.

**Überblick**

In diesem Abschnitt wird beschrieben, wie das BeeGFS-Cluster ordnungsgemäß heruntergefahren und neu gestartet wird. Beispielszenarien, bei denen dies möglicherweise erforderlich ist, sind beispielsweise die elektrische Wartung oder die Migration zwischen Rechenzentren oder Racks.

**Schritte**

Wenn Sie aus irgendeinem Grund das gesamte BeeGFS-Cluster beenden und alle Dienste herunterfahren müssen, laufen:

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

Es ist auch möglich, das Cluster auf einzelnen Nodes anzuhalten (wodurch automatisch ein Failover von Services auf einen anderen Node erfolgt). Es wird jedoch empfohlen, den Node zunächst in den Standby-Modus zu versetzen (siehe "Failover"Abschnitt):

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

So starten Sie Cluster Services und Ressourcen auf allen Nodes:

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

Oder starten Sie Services auf einem bestimmten Knoten mit:

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

An dieser Stelle Lauf `pcs status` Überprüfen Sie, ob die Cluster- und BeeGFS-Services auf allen Nodes gestartet werden und die Services auf den erwarteten Nodes ausgeführt werden.

Je nach Clustergröße kann es Sekunden oder Minuten dauern, bis der gesamte Cluster angehalten wird oder wie gestartet in angezeigt `pcs status` wird. Wenn `pcs cluster <COMMAND>` länger als fünf Minuten hängt, bevor Sie den Befehl mit „Strg+C“ abbrechen, melden Sie sich bei jedem Knoten des Clusters an und prüfen Sie mit `pcs status`, ob Clusterdienste (Corosync/Pacemaker) auf diesem Knoten noch ausgeführt werden. Von jedem Node, der das Cluster noch aktiv ist, können Sie überprüfen, welche Ressourcen das Cluster blockieren. Lösen Sie das Problem manuell, und der Befehl sollte entweder abgeschlossen werden oder kann erneut ausgeführt werden, um alle verbleibenden Services zu beenden.

## Datei-Nodes ersetzen

### Ersetzen eines Dateiknotens, wenn der ursprüngliche Server fehlerhaft ist.

#### Überblick

Dies bietet einen Überblick über die Schritte, die zum Austausch eines Datei-Nodes im Cluster erforderlich sind. Diese Schritte setzen voraus, dass der Datei-Node aufgrund eines Hardwareproblems ausgefallen ist und dass er durch einen neuen identischen File-Node ersetzt wurde.

#### Schritte

1. Ersetzen Sie den Datei-Node physisch und stellen Sie alle Kabel auf den Block-Node und das Storage-Netzwerk wieder her.
2. Installieren Sie das Betriebssystem auf dem Dateiknoten neu, einschließlich Hinzufügen von Red hat Subskriptionen.
3. Konfiguration von Management und BMC Networking auf dem Datei-Node

4. Aktualisieren Sie die Ansible-Bestandsaufnahme, wenn sich der Hostname, die IP, die Zuordnung der PCIe-zu-logischen Schnittstelle oder eine weitere Änderung bezüglich des neuen Datei-Nodes ergeben. Im Allgemeinen ist dies nicht erforderlich, wenn der Node durch identische Serverhardware ersetzt wurde und Sie die ursprüngliche Netzwerkkonfiguration verwenden.
   a. Wenn sich beispielsweise der Hostname geändert hat, erstellen Sie die Bestandsdatei des Node (oder benennen Sie sie um) (`host_vars/<NEW_NODE>.yml`) Und dann in der Ansible-Bestandsdatei (`inventory.yml`), ersetzen Sie den Namen des alten Knotens durch den neuen Knotennamen:

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

5. Entfernen Sie den alten Node von einem der anderen Nodes im Cluster: `pcs cluster node remove <HOSTNAME>`.

   ! FAHREN SIE VOR AUSFÜHRUNG DIESES SCHRITTS NICHT FORT.

6. Auf dem Ansible-Steuerungsknoten:
   a. Entfernen Sie den alten SSH-Schlüssel mit:

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

   b. Konfigurieren Sie passwortloses SSH auf den Knoten Ersetzen mit:

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

7. Führen Sie das Ansible-Playbook erneut aus, um den Node zu konfigurieren und dem Cluster hinzuzufügen:

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

8. An dieser Stelle, Lauf `pcs status` Und überprüfen Sie, ob der ersetzte Node jetzt aufgeführt ist und Services ausführt.

## Erweitern oder verkleinern Sie den Cluster

Fügen Sie dem Cluster Bausteine hinzu oder entfernen Sie diese.

## Überblick

In diesem Abschnitt werden verschiedene Überlegungen und Optionen dokumentiert, um die Größe Ihres BeeGFS HA-Clusters anzupassen. Normalerweise wird die Cluster-Größe durch Hinzufügen oder Entfernen von Bausteinen angepasst. Bei diesen handelt es sich in der Regel um zwei Datei-Nodes, die als HA-Paar eingerichtet wurden. Bei Bedarf können auch einzelne Datei-Nodes (oder andere Cluster-Nodes) hinzugefügt oder entfernt werden.

## Hinzufügen eines Bausteins zum Cluster

### Überlegungen

Das erweitern des Clusters durch Hinzufügen weiterer Bausteine ist ein unkomplizierter Prozess. Beachten Sie zunächst die Einschränkungen der minimalen und maximalen Anzahl von Cluster-Nodes in jedem einzelnen HA-Cluster und bestimmen Sie, ob Sie Nodes zum vorhandenen HA-Cluster hinzufügen oder ein neues HA-Cluster erstellen sollten. Normalerweise besteht jeder Baustein aus zwei Datei-Nodes, aber drei Nodes sind die Mindestanzahl an Nodes pro Cluster (um ein Quorum zu schaffen). Zehn davon ist das empfohlene Maximum (getestete). Für erweiterte Szenarien ist es möglich, einen einzelnen „Tiebreaker“ Node hinzuzufügen, auf dem keine BeeGFS-Services ausgeführt werden, wenn ein Cluster mit zwei Nodes implementiert wird. Bitte wenden Sie sich an den NetApp Support, wenn Sie eine solche Implementierung in Betracht ziehen.

Beachten Sie diese Einschränkungen und das erwartete zukünftige Cluster-Wachstum bei Ihrer Entscheidung über das erweitern des Clusters. Wenn Sie beispielsweise einen sechs-Node-Cluster haben und vier weitere Nodes hinzufügen müssen, empfiehlt es sich, nur einen neuen HA-Cluster zu starten.

Denken Sie daran, dass ein einziges BeeGFS-Dateisystem aus mehreren unabhängigen HA-Clustern bestehen kann. Dadurch können Filesysteme weit über die empfohlenen/harten Grenzen der zugrunde liegenden HA-Cluster-Komponenten hinaus skaliert werden.

### Schritte

Wenn Sie dem Cluster einen Baustein hinzufügen, müssen Sie die `host_vars` Dateien für jeden der neuen Datei-Nodes und Block-Nodes (E-Series-Arrays) erstellen. Die Namen dieser Hosts müssen dem Bestand hinzugefügt werden, zusammen mit den neuen Ressourcen, die erstellt werden sollen. Die entsprechenden `group_vars` Dateien müssen für jede neue Ressource erstellt werden. "Nutzung benutzerdefinierter Architekturen"Weitere Informationen finden Sie im Abschnitt.

Nach dem Erstellen der richtigen Dateien müssen alle erforderlichen Dateien die Automatisierung mit dem Befehl erneut ausführen:

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

## Entfernen eines Bausteins aus dem Cluster

Beachten Sie bei der Außerbetriebnahme eines Baublocks verschiedene Aspekte, z. B.:

- Welche BeeGFS-Services laufen in diesem Baustein?
- Werden nur die File-Nodes ausgemustert und die Block-Nodes mit neuen Datei-Nodes verbunden?
- Wenn der gesamte Baustein außer Betrieb genommen wird, sollten die Daten in einen neuen Baustein verschoben, in vorhandene Nodes im Cluster verteilt oder auf ein neues BeeGFS Filesystem oder ein anderes Storage-System verschoben werden?

- Kann dies bei einem Ausfall oder ohne Unterbrechung geschehen?
- Ist der Baustein aktiv genutzt oder enthält er in erster Linie Daten, die nicht mehr aktiv sind?

Aufgrund der vielfältigen möglichen Ausgangspunkte und gewünschten Endzustände wenden Sie sich bitte an den NetApp Support, damit wir die optimale Strategie basierend auf Ihrer Umgebung und Ihren Anforderungen identifizieren und implementieren können.

# Fehlerbehebung

Fehlerbehebung für ein BeeGFS HA-Cluster.

## Überblick

In diesem Abschnitt wird erläutert, wie verschiedene Fehler und andere Szenarien untersucht und behoben werden können, die beim Betrieb eines BeeGFS HA-Clusters auftreten können.

## Leitfäden Zur Fehlerbehebung

### Untersuchen Unerwarteter Failover

Wenn ein Node unerwartet eingezäunt ist und seine Services auf einen anderen Node verschoben werden, sollte der erste Schritt darin bestehen, zu überprüfen, ob das Cluster auf mögliche Ressourcenausfälle an der Unterseite von hinweist `pcs status`. Normalerweise gibt es keine Daten, wenn das Fechten erfolgreich abgeschlossen wurde und die Ressourcen auf einem anderen Knoten neu gestartet wurden.

Im Allgemeinen wird der nächste Schritt sein, durch die systemd-Logs mit zu suchen `journalctl` Auf einem beliebigen der übrigen Dateiknoten (Pacemaker-Protokolle werden auf allen Knoten synchronisiert). Wenn Sie wissen, wann der Fehler aufgetreten ist, können Sie die Suche kurz vor dem Auftreten des Fehlers starten (in der Regel mindestens zehn Minuten vor dem Auftreten des Fehlers empfohlen):

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

Die folgenden Abschnitte zeigen einen gemeinsamen Text, den Sie in den Protokollen grep können, um die Untersuchung weiter einzugrenzen.

#### Schritte zur Untersuchung/Lösung

**Schritt 1: Prüfen, ob der BeeGFS-Monitor einen Fehler festgestellt hat:**

Wenn das Failover vom BeeGFS-Monitor ausgelöst wurde, sollte ein Fehler angezeigt werden (wenn nicht mit dem nächsten Schritt fortfahren).

```
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
```

In diesem Fall hat der BeeGFS-Service meta_08 aus irgendeinem Grund gestoppt. Um mit der Fehlerbehebung fortzufahren, sollten wir beegfs_02 booten und Protokolle für den Dienst unter überprüfen `/var/log/beegfs-meta-meta_08_tgt_0801.log`. Beispiel: Aufgrund eines internen Problems oder eines Problems mit dem Node konnte für den BeeGFS-Service ein Applikationsfehler aufgetreten sein.

Im Gegensatz zu den Protokollen von Pacemaker werden Protokolle für BeeGFS-Services nicht auf alle Knoten im Cluster verteilt. Um diese Arten von Ausfällen zu untersuchen, sind die Protokolle vom ursprünglichen Knoten, auf dem der Fehler aufgetreten ist, erforderlich.

Mögliche Fehler, die vom Monitor gemeldet werden könnten:

- Auf Ziel(e) kann(n) nicht zugegriffen werden!
  - Beschreibung: Gibt an, auf die Block-Volumes nicht zugegriffen werden konnte.
  - Fehlerbehebung:
    - Wenn auch der Service am alternativen Datei-Node nicht gestartet werden konnte, vergewissern Sie sich, dass der Block-Node ordnungsgemäß ist.
    - Prüfen Sie auf physische Probleme, die den Zugriff auf die Block-Nodes durch diesen Datei-Node verhindern würden, z. B. fehlerhafte InfiniBand-Adapter oder Kabel.
- Netzwerk ist nicht erreichbar!
  - Beschreibung: Keiner der Adapter, die von Clients verwendet wurden, um sich mit diesem BeeGFS-Dienst zu verbinden, war online.
  - Fehlerbehebung:
    - Wenn mehrere/alle Dateiknoten betroffen waren, überprüfen Sie, ob ein Fehler im Netzwerk vorhanden ist, das zum Verbinden der BeeGFS-Clients und des Dateisystems verwendet wurde.
    - Prüfen Sie, ob physikalische Probleme den Zugriff auf die Clients durch diesen Dateiknoten verhindern würden, z. B. fehlerhafte InfiniBand-Adapter oder Kabel.
- BeeGFS-Service ist nicht aktiv!
  - Beschreibung: Ein BeeGFS-Dienst hat unerwartet gestoppt.
  - Fehlerbehebung:
    - Überprüfen Sie auf dem Datei-Node, der den Fehler gemeldet hat, die Protokolle für den betroffenen BeeGFS-Dienst, ob er einen Absturz gemeldet hat. Öffnen Sie in diesem Fall einen Fall mit NetApp Support, damit der Absturz untersucht werden kann.
    - Wenn im BeeGFS-Protokoll keine Fehler gemeldet werden, prüfen Sie in den Journalprotokollen, ob systemd einen Grund protokolliert hat, warum der Dienst angehalten wurde. In einigen Fällen wurde dem BeeGFS-Dienst möglicherweise keine Chance gegeben, Nachrichten zu protokollieren, bevor der Prozess beendet wurde (z. B. wenn jemand ausgeführt wurde `kill -9 <PID>`).

**Schritt 2: Prüfen Sie, ob der Node das Cluster unerwartet verlassen hat**

Falls auf dem Node ein schwerwiegender Hardware-Ausfall auftritt (z. B. die Systemplatine gestorben) oder ein Kernel-Panic oder ein ähnliches Softwareproblem auftritt, wird der BeeGFS-Monitor keinen Fehler melden. Suchen Sie stattdessen nach dem Hostnamen und Sie sollten Meldungen von Pacemaker sehen, die darauf hinweisen, dass der Knoten unerwartet verloren gegangen ist:

```
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
```

**Schritt 3: Überprüfen Sie, ob Pacemaker in der Lage war, den Knoten einzuzäunen**

In allen Szenarien sollten Sie sehen, dass Pacemaker versucht, den Knoten einzuzäunen, um zu überprüfen, ob er tatsächlich offline ist (genaue Meldungen können von der Ursache des Fechts abweichen):

```
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
```

Wenn die Fechtaktion erfolgreich abgeschlossen ist, werden folgende Meldungen angezeigt:

```
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
```

Wenn die Fechten-Aktion aus irgendeinem Grund fehlgeschlagen ist, können die BeeGFS-Dienste auf einem anderen Node nicht neu starten, um Datenkorruption zu vermeiden. Das wäre ein Problem, separat zu untersuchen, wenn zum Beispiel das Fechten-Gerät (PDU oder BMC) unzugänglich oder falsch konfiguriert war.

**Adressen fehlgeschlagener Ressourcen Aktionen (am Ende des Stk-Status gefunden)**

Wenn eine Ressource, die zum Ausführen eines BeeGFS-Dienstes erforderlich ist, ausfällt, wird ein Failover durch den BeeGFS-Monitor ausgelöst. Wenn dies der Fall ist, werden wahrscheinlich keine „fehlgeschlagenen Ressourcenaktionen“ am Ende von aufgeführt `pcs status`, und Sie sollten die Schritte zum Thema "Failback nach einem ungeplanten Failover"lesen.

Ansonsten sollte es in der Regel nur zwei Szenarien geben, in denen Sie „Aktionen für fehlgeschlagene Ressourcen“ sehen.

#### Schritte zur Untersuchung/Lösung

**Szenario 1: Bei einem Fechten-Agent wurde ein temporäres oder dauerhaftes Problem erkannt und es wurde neu gestartet oder auf einen anderen Knoten verschoben.**

Einige Fechten-Agenten sind zuverlässiger als andere, und jeder implementiert seine eigene Überwachungsmethode, um sicherzustellen, dass die Fechtvorrichtung bereit ist. Insbesondere wurde festgestellt, dass der Fechtagent von Redfish fehlgeschlagene Ressourcenaktionen wie die folgenden meldet, obwohl er immer noch gestartet wird:

```
  * 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
```

Ein Fechten-Agent, der fehlgeschlagene Ressourcen-Aktionen auf einem bestimmten Knoten meldet, wird nicht erwartet, dass ein Failover der BeeGFS-Dienste ausgelöst wird, die auf diesem Knoten ausgeführt werden. Es sollte einfach automatisch auf demselben oder einem anderen Knoten neu gestartet werden.

Schritte zur Lösung:

1. Wenn der Fechtagent sich immer wieder weigert, auf allen oder einer Untermenge von Knoten ausgeführt zu werden, überprüfen Sie, ob diese Knoten eine Verbindung zum Fechtagenten herstellen können, und überprüfen Sie, ob der Fechtagent im Ansible-Bestand korrekt konfiguriert ist.
   a. Wenn z. B. ein Fechten-Agent von Redfish (BMC) auf demselben Knoten ausgeführt wird, wie er für das Fechten verantwortlich ist, und die Betriebssystemverwaltung und BMC-IPs auf derselben physischen Schnittstelle sind, ermöglichen einige Netzwerk-Switch-Konfigurationen keine Kommunikation zwischen den beiden Schnittstellen (um Netzwerkschleifen zu verhindern). Standardmäßig versucht das HA-Cluster, keine Fechten-Agenten auf dem Node zu platzieren, den sie für Fechten verantwortlich sind, aber dies kann in einigen Szenarien/Konfigurationen geschehen.
2. Sobald alle Probleme behoben sind (oder das Problem scheinbar kurzlebig zu sein schien), führen Sie den folgenden Lauf aus `pcs resource cleanup` So setzen Sie die fehlgeschlagenen Ressourcenaktionen zurück.

**Szenario 2: Der BeeGFS-Monitor hat ein Problem erkannt und ein Failover ausgelöst, aber aus irgendeinem Grund konnte das System nicht auf einem sekundären Knoten starten.**

Sofern das Fechten aktiviert ist und die Ressource nicht vom Stoppen auf dem ursprünglichen Knoten blockiert wurde (siehe Abschnitt Fehlerbehebung für „Standby (on-fail)“)), sind die wahrscheinlichsten Gründe, warum Probleme auftreten, die die Ressource auf einem sekundären Knoten zu starten, weil:

- Der sekundäre Node war bereits offline.
- Ein physisches oder logisches Konfigurationsproblem verhindert, dass das sekundäre System auf die als BeeGFS-Ziele verwendeten Block-Volumes zugreift.

Schritte zur Lösung:

1. Für jeden Eintrag in den Aktionen für fehlgeschlagene Ressourcen:
   a. Bestätigen Sie, dass die fehlgeschlagene Ressourcenaktion ein Startvorgang war.
   b. Basierend auf der in den Aktionen für fehlgeschlagene Ressourcen angegebenen Ressource und dem in den Knoten angegebenen Ressource:

i. Suchen Sie nach externen Problemen, die verhindern würden, dass der Knoten die angegebene Ressource startet, und beheben Sie diese. Wenn zum Beispiel BeeGFS IP-Adresse (Floating IP) nicht gestartet werden konnte, vergewissern Sie sich, dass mindestens eine der erforderlichen Schnittstellen angeschlossen/online ist und mit dem richtigen Netzwerk-Switch verbunden ist. Wenn ein BeeGFS-Ziel (Blockgerät/E-Series-Volume) fehlgeschlagen ist, überprüfen Sie, ob die physischen Verbindungen zu den Backend-Block-Nodes wie erwartet verbunden sind, und überprüfen Sie, ob die Block-Nodes ordnungsgemäß sind.

c. Wenn es keine offensichtlichen externen Probleme gibt und Sie eine Ursache für diesen Vorfall wünschen, sollten Sie einen Case mit dem NetApp Support eröffnen, um ihn zu untersuchen, bevor Sie fortfahren, da die folgenden Schritte eine Ursachenanalyse (Root Cause Analysis, RCA) schwierig/unmöglich machen können.

2. Nach der Lösung externer Probleme:

a. Kommentieren Sie alle nicht funktionierenden Nodes aus der Ansible Inventory.yml-Datei und führen Sie das vollständige Ansible-Playbook erneut aus, um sicherzustellen, dass die logische Konfiguration auf den/den sekundären Nodes korrekt eingerichtet ist.

i. Hinweis: Vergessen Sie nicht, diese Nodes zu kommentieren und das Playbook erneut auszuführen, sobald sich die Nodes in einem ordnungsgemäßen Zustand befinden und Sie zum Failback bereit sind.

b. Alternativ können Sie versuchen, das Cluster manuell wiederherzustellen:

i. Platzieren Sie alle Offline-Nodes wieder online mithilfe von: `pcs cluster start <HOSTNAME>`

ii. Löschen Sie alle fehlgeschlagenen Ressourcenaktionen mit: `pcs resource cleanup`

iii. Stk-Status ausführen und überprüfen, ob alle Dienste wie erwartet beginnen.

iv. Bei Bedarf ausführen `pcs resource relocate run` Verschieben von Ressourcen zurück auf den bevorzugten Node (sofern verfügbar)

## Häufige Probleme

#### BeeGFS-Services führen bei Anforderung kein Failover oder Failback durch

**Wahrscheinliche Ausgabe:** das `pcs resource relocate` Befehl ausführen wurde ausgeführt, aber nie erfolgreich abgeschlossen.

**So überprüfen Sie:** Lauf `pcs constraint --full` Und überprüfen Sie auf alle Standortbeschränkungen mit einer ID von `pcs-relocate-<RESOURCE>`.

**Wie löst man:** Lauf `pcs resource relocate clear` Wiederholen Sie anschließend den Test `pcs constraint --full` Um zu überprüfen, ob die zusätzlichen Bedingungen entfernt wurden.

#### Ein Knoten im Stk-Status zeigt „Standby (ein-aus)“ an, wenn das Fechten deaktiviert ist

**Wahrscheinliche Ursache:** Pacemaker konnte nicht erfolgreich bestätigen, dass alle Ressourcen auf dem Knoten, der ausgefallen ist, angehalten wurden.

**Wie löst man:**

1. Laufen `pcs status` Und überprüfen Sie, ob die Ressourcen nicht „gestartet“ sind, oder zeigen Sie Fehler an der Unterseite der Ausgabe an, und beheben Sie eventuelle Probleme.
2. Um den Node wieder in den Online-Modus zu versetzen, wird ausgeführt `pcs resource cleanup --node=<HOSTNAME>`.

#### Nach einem unerwarteten Failover zeigen die Ressourcen „gestartet (ein-Fehler)“ im Stk-Status an, wenn das Fechten aktiviert ist

**Wahrscheinliches Problem:** Es trat ein Problem auf, das einen Failover auslöste, Pacemaker konnte jedoch nicht überprüfen, ob der Knoten eingezäunt war. Dies kann passieren, weil Fechten falsch konfiguriert war oder es ein Problem mit dem Fechten Agent gab (Beispiel: Die PDU wurde vom Netzwerk getrennt).

**Wie löst man:**

1. Vergewissern Sie sich, dass der Node tatsächlich ausgeschaltet ist.

   

   Wenn der von Ihnen angegebene Node nicht aktiv ist, der aber Cluster-Services oder -Ressourcen ausführt, treten Datenbeschädigungen/Cluster-Ausfälle auf.

2. Fechten manuell bestätigen mit: `pcs stonith confirm <NODE>`

An diesem Punkt sollten die Dienste den Failover beenden und auf einem anderen gesunden Knoten neu gestartet werden.

## Häufige Fehlerbehebungsaufgaben

#### Starten Sie individuelle BeeGFS-Dienste neu

Normalerweise, wenn ein BeeGFS-Service neu gestartet werden muss (z. B. um eine Konfigurationsänderung zu ermöglichen), sollte dies durch Aktualisierung des Ansible-Bestands und durch erneute Ausführung des Playbooks geschehen. In manchen Szenarien kann es wünschenswert sein, einzelne Services neu zu starten, um eine schnellere Fehlerbehebung zu ermöglichen, beispielsweise um das Protokollierungsniveau zu ändern, ohne auf die Ausführung des gesamten Playbooks zu warten.

Wenn nicht auch manuelle Änderungen am Ansible-Inventar hinzugefügt werden, werden diese bei der nächsten Ausführung des Ansible-Playbooks zurückgesetzt.

##### Option 1: Systemgesteuerter Neustart

Wenn das Risiko besteht, dass der BeeGFS-Service mit der neuen Konfiguration nicht ordnungsgemäß neu gestartet wird, versetzen Sie das Cluster zuerst in den Wartungsmodus, um zu verhindern, dass der BeeGFS-Monitor den Service erkennt, angehalten wird und ein unerwünschtes Failover ausgelöst wird:

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

Nehmen Sie ggf. Änderungen an der Servicekonfiguration unter vor `/mnt/<SERVICE_ID>/_config/beegfs-.conf` (Beispiel: `/mnt/meta_01_tgt_0101/metadata_config/beegfs-meta.conf`) Dann systemd verwenden, um es neu zu starten:

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

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

#### Option 2: Schrittmachergesteuerter Neustart

Wenn Sie keine Sorge haben, dass die neue Konfiguration dazu führen könnte, dass der Service unerwartet angehalten wird (z. B. einfach die Protokollierungsebene ändern), oder Sie sich in einem Wartungsfenster befinden und sich keine Gedanken über Ausfallzeiten machen, können Sie den BeeGFS-Monitor einfach für den Service neu starten, den Sie neu starten möchten:

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

Zum Beispiel zum Neustart des BeeGFS-Managementdienstes: `pcs resource restart mgmt-monitor`