Aktualisieren Sie die HA-Cluster-Komponenten

BeeGFS on NetApp with E-Series Storage

NetApp
January 27, 2026

Inhalt

Aktualisieren Sie die HA-Cluster-Komponenten . . . 1
- Upgrade der BeeGFS Services . . . 1
  - Überblick . . . 1
  - Getestete Upgrade-Pfade . . . 1
  - Schritte beim BeeGFS-Upgrade . . . 2
  - Hinweise zur Versionsaktualisierung . . . 3
- Upgrade auf BeeGFS v8 . . . 4
  - Überblick . . . 4
  - Wichtige Änderungen in BeeGFS v8 . . . 4
  - Bereiten Sie Ihren BeeGFS Cluster auf das Upgrade vor . . . 5
  - Aktualisieren Sie die BeeGFS-Pakete . . . 6
  - Aktualisieren Sie die Management-Datenbank . . . 6
  - Lizenzierung konfigurieren . . . 8
  - TLS-Verschlüsselung konfigurieren . . . 8
  - Konfiguration des Update-Management-Dienstes . . . 8
  - Aktualisieren Sie das BeeGFS-Monitor-Skript . . . 10
  - Den Cluster wieder online bringen . . . 12
  - BeeGFS-Clients aktualisieren . . . 13
  - Überprüfen Sie das Upgrade . . . 14
- Aktualisieren Sie Pacemaker- und Corosync-Pakete in einem HA-Cluster . . . 14
  - Überblick . . . 14
  - Upgrade-Ansatz . . . 14
- Aktualisiert die Datei-Node-Adapter-Firmware . . . 17
  - Überblick . . . 18
  - Upgrade-Überlegungen . . . 18
  - Vorbereitung des Firmware-Updates . . . 18
  - Rollierender Aktualisierungsansatz . . . 18
  - Update für Cluster mit zwei Nodes . . . 20
- Upgrade von E-Series Storage-Arrays . . . 22
  - Überblick . . . 22
  - Upgrade-Schritte für Block-Nodes . . . 22

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.