Update dei componenti del cluster ha

BeeGFS on NetApp with E-Series Storage

NetApp
January 27, 2026

Sommario

Update dei componenti del cluster ha

Aggiorna i servizi BeeGFS

Utilizzare Ansible per aggiornare la versione di BeeGFS in esecuzione sul cluster HA.

Panoramica

BeeGFS segue uno `major.minor.patch` schema di versioning. I ruoli Ansible ha BeeGFS sono forniti per ogni `major.minor` versione supportata (ad esempio, `beegfs_ha_7_2` e `beegfs_ha_7_3`). Ogni ruolo ha è associato all'ultima versione della patch BeeGFS disponibile al momento della release della raccolta Ansible.

Ansible deve essere utilizzato per tutti gli aggiornamenti di BeeGFS, incluso il passaggio tra versioni principali, secondarie e patch di BeeGFS. Per aggiornare BeeGFS, è necessario innanzitutto aggiornare la raccolta Ansible di BeeGFS, che includerà anche le ultime correzioni e miglioramenti all'automazione di distribuzione/gestione e al cluster HA sottostante. Anche dopo l'aggiornamento all'ultima versione della raccolta, BeeGFS non verrà aggiornato finché `ansible-playbook` non verrà eseguito con il `-e "beegfs_ha_force_upgrade=true"` set. Per ulteriori dettagli su ciascun aggiornamento, fare riferimento alla "Documentazione sull'aggiornamento di BeeGFS" per la versione corrente.

> Se si sta eseguendo l'aggiornamento a BeeGFS v8, consultare la "Aggiorna a BeeGFS v8" procedura invece.

Percorsi di aggiornamento testati

Sono stati testati e verificati i seguenti percorsi di aggiornamento:

Versione originale	Versione dell'aggiornamento	Multirrail	Dettagli
7.2.6	7.3.2	Sì	Aggiornamento della raccolta beegfs da v3.0.1 a v3.1.0, aggiunta di multi-rail
7.2.6	7.2.8	No	Aggiornamento della raccolta beegfs da v3.0.1 a v3.1.0
7.2.8	7.3.1	Sì	Aggiornamento con la raccolta beegfs v3.1.0, aggiunta di multi-rail
7.3.1	7.3.2	Sì	Eseguire l'aggiornamento utilizzando la raccolta beegfs v3.1.0
7.3.2	7.4.1	Sì	Eseguire l'aggiornamento utilizzando la raccolta beegfs v3.2.0
7.4.1	7.4.2	Sì	Eseguire l'aggiornamento utilizzando la raccolta beegfs v3.2.0
7.4.2	7.4.6	Sì	Eseguire l'aggiornamento utilizzando la raccolta beegfs v3.2.0
7.4.6	8,0	Sì	Eseguire l'aggiornamento utilizzando le istruzioni nella "Aggiorna a BeeGFS v8" procedura.
7.4.6	8,1	Sì	Eseguire l'aggiornamento utilizzando le istruzioni nella "Aggiorna a BeeGFS v8" procedura.
7.4.6	8,2	Sì	Eseguire l'aggiornamento utilizzando le istruzioni nella "Aggiorna a BeeGFS v8" procedura.

Fasi di aggiornamento di BeeGFS

Nelle sezioni seguenti sono riportati i passaggi per aggiornare la raccolta BeeGFS Ansible e BeeGFS stessa. Prestare particolare attenzione a eventuali passaggi aggiuntivi per l'aggiornamento delle versioni principali o secondarie di BeeGFS.

Passaggio 1: Aggiornamento della raccolta BeeGFS

Per gli aggiornamenti del ritiro con accesso a. "Ansible Galaxy", eseguire il seguente comando:

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

Per gli aggiornamenti offline della raccolta, scarica la raccolta da "Ansible Galaxy" facendo clic sul pulsante desiderato `Install Version`` e poi `Download tarball`. Trasferire il tarball al nodo di controllo Ansible ed eseguire il seguente comando.

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

Vedere "Installazione delle raccolte" per ulteriori informazioni.

Fase 2: Aggiornare l'inventario Ansible

Apporta tutti gli aggiornamenti necessari o desiderati ai file di inventario Ansible del tuo cluster. Consulta la sezione Note sull'aggiornamento della versione qui sotto per i dettagli sui tuoi specifici requisiti di aggiornamento. Consulta la sezione "Panoramica di Ansible Inventory" per informazioni generali sulla configurazione dell'inventario HA di BeeGFS.

Fase 3: Aggiornamento del playbook Ansible (solo per l'aggiornamento delle versioni principali o secondarie)

Se si passa da una versione principale a una versione secondaria, nel `playbook.yml` file utilizzato per distribuire e gestire il cluster, aggiornare il nome del `beegfs_ha_<VERSION>` ruolo in modo che rifletta la versione desiderata. Ad esempio, se si desidera distribuire BeeGFS 7,4 questo sarà `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
```

Per ulteriori dettagli sul contenuto del file del presente manuale, consulta "Implementare il cluster BeeGFS

ha"la sezione.

Passaggio 4: Eseguire l'aggiornamento BeeGFS

Per applicare l'aggiornamento BeeGFS:

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

Dietro le quinte, il ruolo di BeeGFS ha gestirà:

- Assicurarsi che il cluster si trovi in uno stato ottimale con ciascun servizio BeeGFS situato sul nodo preferito.
- Impostare il cluster in modalità di manutenzione.
- Aggiornare i componenti del cluster ha (se necessario).
- Aggiornare ciascun nodo di file uno alla volta come segue:
  - Metterlo in standby e eseguire il failover dei servizi sul nodo secondario.
  - Aggiornare i pacchetti BeeGFS.
  - Servizi di fallback.
- Spostare il cluster fuori dalla modalità di manutenzione.

Note sull'aggiornamento della versione

Aggiornamento da BeeGFS versione 7.2.6 o 7.3.0

Modifiche all'autenticazione basata su connessione

BeeGFS versione 7.3.2 e successive richiedono che l'autenticazione basata sulla connessione sia configurata. I servizi non si avvieranno senza una delle seguenti opzioni:

- Specificando un `connAuthFile`, o
- Impostazione `connDisableAuthentication=true` nel file di configurazione del servizio.

Si consiglia vivamente di abilitare l'autenticazione basata sulla connessione per la sicurezza. Vedere "Autenticazione basata su connessione BeeGFS" per ulteriori informazioni.

I `beegfs_ha*` ruoli generano e distribuiscono automaticamente il file di autenticazione a:

- Tutti i nodi file nel cluster
- Il nodo di controllo Ansible a `<playbook_directory>/files/beegfs/<beegfs_mgmt_ip_address>_connAuthFile`

Il `beegfs_client` ruolo rileverà automaticamente e applicherà questo file ai client quando sarà presente.

Se non hai utilizzato il `beegfs_client` ruolo per configurare i client, devi distribuire manualmente il file di autenticazione a ciascun client e configurare l'impostazione `connAuthFile` nel file `beegfs-client.conf`. Quando si esegue l'aggiornamento da una versione di BeeGFS senza autenticazione basata sulla connessione, i client perderanno l'accesso a meno che non si disabiliti l'autenticazione basata sulla connessione durante l'aggiornamento impostando `beegfs_ha_conn_auth_enabled: false` in `group_vars/ha_cluster.yml` (scelta non consigliata).

Per ulteriori dettagli e opzioni di configurazione alternative, vedere il passaggio di configurazione dell'autenticazione della connessione nella sezione "Specificare la configurazione del nodo file comune".

Aggiorna a BeeGFS v8

Segui questi passaggi per aggiornare il tuo cluster BeeGFS HA dalla versione 7.4.6 a BeeGFS v8.

Panoramica

BeeGFS v8 introduce diverse modifiche significative che richiedono una configurazione aggiuntiva prima dell'aggiornamento da BeeGFS v7. Questo documento guida l'utente nella preparazione del cluster per i nuovi requisiti di BeeGFS v8 e quindi nell'aggiornamento a BeeGFS v8.

Prima di eseguire l'aggiornamento a BeeGFS v8, assicurati che il tuo sistema esegua almeno BeeGFS 7.4.6. Qualsiasi cluster che esegue una release precedente a BeeGFS 7.4.6 deve prima "aggiorna alla versione 7.4.6" procedere con questa procedura di aggiornamento a BeeGFS v8.

Modifiche chiave in BeeGFS v8

BeeGFS v8 introduce le seguenti modifiche principali:

- **Applicazione della licenza:** BeeGFS v8 richiede una licenza per utilizzare funzionalità premium come pool di archiviazione, destinazioni di archiviazione remote, BeeOND e altro ancora. Acquisisci una licenza valida per il tuo cluster BeeGFS prima di effettuare l'aggiornamento. Se necessario, puoi ottenere una licenza di valutazione temporanea di BeeGFS v8 dal "Portale delle licenze BeeGFS".
- **Migrazione del database del servizio di gestione:** per abilitare la configurazione con il nuovo formato basato su TOML in BeeGFS v8, è necessario migrare manualmente il database del servizio di gestione BeeGFS v7 al formato BeeGFS v8 aggiornato.
- **Crittografia TLS:** BeeGFS v8 introduce TLS per la comunicazione sicura tra i servizi. Sarà necessario generare e distribuire certificati TLS per il servizio di gestione BeeGFS e la `beegfs` command-line utility come parte dell'aggiornamento.

Per maggiori dettagli e ulteriori modifiche in BeeGFS 8, vedere la "Guida all'aggiornamento di BeeGFS v8.0.0".

L'aggiornamento a BeeGFS v8 richiede il downtime del cluster. Inoltre, i client BeeGFS v7 non possono connettersi ai cluster BeeGFS v8. Coordinare attentamente i tempi di aggiornamento tra il cluster e i client per ridurre al minimo l'impatto sulle operazioni.

Prepara il tuo cluster BeeGFS per l'aggiornamento

Prima di iniziare l'upgrade, prepara attentamente il tuo ambiente per garantire una transizione fluida e ridurre al minimo i tempi di inattività.

1. Assicurati che il tuo cluster sia in uno stato integro, con tutti i servizi BeeGFS in esecuzione sui nodi preferiti. Da un nodo file che esegue i servizi BeeGFS, verifica che tutte le risorse Pacemaker siano in esecuzione sui nodi preferiti:

```
pcs status
```

2. Registra ed esegui il backup della configurazione del cluster.
   a. Fare riferimento a "Documentazione BeeGFS Backup" per le istruzioni sul backup della configurazione del cluster.
   b. Eseguire il backup della directory dei dati di gestione esistente:

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

   c. Eseguire i seguenti comandi da un client beegfs e salvare il loro output per riferimento:

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

   d. Se si utilizza il mirroring, raccogliere informazioni dettagliate sullo stato:

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

3. Prepara i tuoi clienti ai tempi di inattività e interrompi `beegfs-client` i servizi. Per ogni cliente, esegui:

```
systemctl stop beegfs-client
```

4. Per ogni cluster Pacemaker, disabilitare STONITH. Ciò consentirà di convalidare l'integrità del cluster dopo l'aggiornamento senza innescare inutili riavvii dei nodi.

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

5. Per tutti i cluster Pacemaker nello spazio dei nomi BeeGFS, utilizzare PCS per arrestare il cluster:

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

Aggiorna i pacchetti BeeGFS

Su tutti i nodi file del cluster, aggiungi il repository del pacchetto BeeGFS v8 per la tua distribuzione Linux. Le istruzioni per l'utilizzo dei repository ufficiali di BeeGFS sono disponibili al "Pagina di download di BeeGFS". In caso contrario, configura di conseguenza il tuo repository mirror locale beegfs.

I seguenti passaggi illustrano l'utilizzo del repository ufficiale BeeGFS 8.2 sui nodi file RHEL 9. Eseguire i seguenti passaggi su tutti i nodi file del cluster:

1. Importa la chiave GPG BeeGFS:

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

2. Importa il repository BeeGFS:

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

> Rimuovere tutti i repository BeeGFS configurati in precedenza per evitare conflitti con il nuovo repository BeeGFS v8.

3. Pulisci la cache del tuo gestore di pacchetti:

```
dnf clean all
```

4. Su tutti i nodi file, aggiornare i pacchetti BeeGFS a BeeGFS 8.2.

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

> In un cluster standard, il `beegfs-mgmtd` package verrà aggiornato solo sui primi due nodi file.

Aggiornare il database di gestione

Su uno dei nodi file che eseguono il servizio di gestione BeeGFS, eseguire i passaggi seguenti per migrare il database di gestione da BeeGFS v7 a v8.

1. Elenca tutti i dispositivi NVMe e filtra per la management target:

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

a. Annotare il percorso del dispositivo dall'output.

b. Montare il dispositivo di destinazione di gestione sul punto di montaggio della destinazione di gestione esistente (sostituire `/dev/nvmeXnY` con il percorso del dispositivo):

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

2. Importa i dati di gestione BeeGFS 7 nel nuovo formato di database eseguendo:

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

Output previsto:

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

> L'importazione automatica potrebbe non riuscire in tutti i casi a causa dei requisiti di convalida più rigorosi in BeeGFS v8. Ad esempio, se le destinazioni sono assegnate a pool di archiviazione inesistenti, l'importazione non riuscirà. Se la migrazione del database non riesce, non procedere con l'aggiornamento. Contattare il supporto NetApp per assistenza nella risoluzione dei problemi di migrazione del database. Come soluzione provvisoria, è possibile effettuare il downgrade dei pacchetti BeeGFS v8 e continuare a eseguire BeeGFS v7 mentre il problema viene risolto.

3. Sposta il file SQLite generato nel mount del servizio di gestione:

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

4. Sposta il file generato `beegfs-mgmtd.toml` sul mount del servizio di gestione:

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

La preparazione del file di configurazione `beegfs-mgmtd.toml` verrà effettuata dopo aver completato i passaggi di licensing e configurazione TLS nelle prossime sezioni.

Configurare le licenze

1. Installare i pacchetti di licenza beegfs su tutti i nodi che eseguono il servizio di gestione beegfs. In genere si tratta dei primi due nodi del cluster:

```
dnf install libbeegfs-license
```

2. Scarica il file di licenza BeeGFS v8 nei nodi di gestione e posizionalo in:

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

Configura la crittografia TLS

BeeGFS v8 richiede la crittografia TLS per comunicazioni sicure tra servizi di gestione e client. Esistono tre opzioni per configurare la crittografia TLS sulle comunicazioni di rete tra servizi di gestione e servizi client. Il metodo consigliato e più sicuro è utilizzare certificati firmati da una Certificate Authority attendibile. In alternativa, puoi creare una CA locale per firmare i certificati per il tuo cluster BeeGFS. Per gli ambienti in cui la crittografia non è richiesta o per la risoluzione dei problemi, TLS può essere completamente disabilitato, anche se questa opzione è sconsigliata poiché espone informazioni sensibili alla rete.

Prima di procedere, segui le istruzioni nella "Configura la crittografia TLS per BeeGFS 8" guida per configurare la crittografia TLS per il tuo ambiente.

Aggiorna la configurazione del servizio di gestione

Prepara il file di configurazione del servizio di gestione BeeGFS v8 trasferendo manualmente le impostazioni dal tuo file di configurazione BeeGFS v7 nel file `/mnt/mgmt_tgt_mgmt01/mgmt_config/beegfs-mgmtd.toml`.

1. Sul nodo di gestione con il target di gestione montato, fare riferimento al `/mnt/mgmt_tgt_mgmt01/mgmt_config/beegfs-mgmtd.conf` file del servizio di gestione per BeeGFS 7, quindi trasferire tutte le impostazioni al `/mnt/mgmt_tgt_mgmt01/mgmt_config/beegfs-mgmtd.toml` file. Per una configurazione di base, il tuo `beegfs-mgmtd.toml` potrebbe essere simile al seguente:

```
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']
```

Adatta tutti i percorsi secondo necessità in modo che corrispondano al tuo ambiente e alla configurazione

TLS.

2. Su ogni file node che esegue servizi di gestione, modifica il file di servizio systemd in modo che punti alla nuova posizione del file di configurazione.

```
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. Ricarica systemd:

```
systemctl daemon-reload
```

3. Per ogni file node che esegue servizi di gestione, aprire la porta 8010 per la comunicazione gRPC del servizio di gestione.

   a. Aggiungi la porta 8010/tcp alla zona beegfs:

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

   b. Ricarica il firewall per applicare la modifica:

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

Aggiorna lo script monitor di BeeGFS

Lo script OCF di Pacemaker `beegfs-monitor` richiede aggiornamenti per supportare il nuovo formato di configurazione TOML e la gestione dei servizi systemd. Aggiorna lo script su un nodo del cluster, quindi copia lo script aggiornato su tutti gli altri nodi.

1. Crea un backup dello script corrente:

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

2. Aggiorna il percorso del file di configurazione di gestione da `.conf` a `.toml`:

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

   In alternativa, individua manualmente il seguente blocco nello script:

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

E sostituiscilo con:

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

3. Aggiorna le funzioni `get_interfaces()` e `get_subnet_ips()` per supportare la configurazione TOML:
   a. Apri lo script in un editor di testo:

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

   b. Individua le due funzioni: `get_interfaces()` e `get_subnet_ips()`.
   c. Eliminare entrambe le funzioni intere, iniziando da `get_interfaces()` fino alla fine di `get_subnet_ips()`.
   d. Copia e incolla le seguenti funzioni aggiornate al loro posto:

```
# 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. Salva e esci dall'editor di testo.

f. Eseguire il comando seguente per verificare la presenza di errori di sintassi nello script prima di procedere. L'assenza di output indica che lo script è sintatticamente corretto.

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

4. Copia lo script OCF aggiornato `beegfs-monitor` su tutti gli altri nodi del cluster per garantire la coerenza:

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

Riporta il cluster online

1. Una volta completati tutti i passaggi di aggiornamento precedenti, riportare il cluster online avviando i servizi BeeGFS su tutti i nodi.

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

2. Verificare che il `beegfs-mgmtd` servizio sia stato avviato correttamente:

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

L'output previsto include righe come:

```
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
```

Se nei log del journal vengono visualizzati degli errori, rivedere i percorsi del file di configurazione di gestione e assicurarsi che tutti i valori siano stati trasferiti correttamente dal file di configurazione BeeGFS 7.

3. Eseguire `pcs status` e verificare che il cluster sia integro e che i servizi siano avviati sui nodi preferiti.

4. Una volta verificato che il cluster è integro, riattivare STONITH:

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

5. Passare alla sezione successiva per aggiornare i client BeeGFS nel cluster e verificare lo stato di integrità del cluster BeeGFS.

Aggiorna i client BeeGFS

Dopo aver aggiornato correttamente il tuo cluster a BeeGFS v8, devi anche aggiornare tutti i client BeeGFS.

I passaggi seguenti illustrano il processo per aggiornare i client BeeGFS su un sistema basato su Ubuntu.

1. Se non è già stato fatto, arrestare il servizio BeeGFS client:

```
systemctl stop beegfs-client
```

2. Aggiungi il repository del pacchetto BeeGFS v8 per la tua distribuzione Linux. Le istruzioni per l'utilizzo dei repository ufficiali BeeGFS sono disponibili al "^Pagina di download di BeeGFS". In caso contrario, configura di conseguenza il tuo repository mirror locale BeeGFS.

   I passaggi seguenti utilizzano il repository ufficiale BeeGFS 8.2 su un sistema basato su Ubuntu:

3. Importa la chiave GPG BeeGFS:

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

4. Scarica il file del repository:

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

> Rimuovere tutti i repository BeeGFS configurati in precedenza per evitare conflitti con il nuovo repository BeeGFS v8.

5. Aggiorna i pacchetti client BeeGFS:

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

6. Configurare TLS per il client. TLS è necessario per utilizzare la CLI di BeeGFS. Fare riferimento alla "Configura la crittografia TLS per BeeGFS 8" procedura per configurare TLS sul client.
7. Avvia il servizio client BeeGFS:

```
systemctl start beegfs-client
```

Verificare l'upgrade

Dopo aver completato l'upgrade a BeeGFS v8, eseguire i seguenti comandi per verificare che l'upgrade sia stato completato con successo.

1. Verificare che l'inode root sia di proprietà dello stesso nodo metadati di prima. Questo dovrebbe avvenire automaticamente se si è utilizzata la `import-from-v7` funzionalità nel servizio di gestione:

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

2. Verificare che tutti i nodi e le destinazioni siano online e in buono stato:

```
beegfs health check
```

> Se il controllo "Capacità disponibile" avvisa che i target hanno poco spazio libero, è possibile modificare le soglie del "capacity pool" definite nel `beegfs-mgmtd.toml` file in modo che siano più adatte al proprio ambiente.

Aggiornare i pacchetti pacemaker e Corosync in un cluster ha

Per aggiornare i pacchetti pacemaker e Corosync in un cluster ha, procedere come segue.

Panoramica

L'aggiornamento di Pacemaker e Corosync garantisce al cluster i vantaggi derivanti da nuove funzioni, patch di sicurezza e miglioramenti delle prestazioni.

Approccio all'upgrade

Ci sono due approcci consigliati per aggiornare un cluster: Un aggiornamento in corso o un arresto completo del cluster. Ogni approccio ha i propri vantaggi e svantaggi. La procedura di aggiornamento può variare a seconda della versione del pacemaker in uso. Fare riferimento alla documentazione di ClusterLabs "Aggiornamento di un quadro pacemaker"per determinare l'approccio da utilizzare. Prima di adottare un approccio all'aggiornamento, verificare che:

- I nuovi pacchetti pacemaker e Corosync sono supportati all'interno della soluzione BeeGFS di NetApp.
- Esistono backup validi per il file system BeeGFS e la configurazione del cluster pacemaker.
- Il cluster è in uno stato sano.

Rolling upgrade

Questo metodo prevede la rimozione di ciascun nodo dal cluster, l'aggiornamento e la reintroduzione nel cluster fino a quando tutti i nodi non eseguono la nuova versione. Questo approccio mantiene operativo il cluster, ideale per cluster ha di maggiori dimensioni, con il rischio di eseguire versioni miste durante il processo. Questo approccio deve essere evitato in un cluster a due nodi.

1. Verificare che il cluster sia in uno stato ottimale, con ogni servizio BeeGFS in esecuzione sul nodo preferito. Per ulteriori informazioni, fare riferimento alla "Esaminare lo stato del cluster" sezione.
2. Per aggiornare il nodo, impostarlo in modalità standby per scaricare (o spostare) tutti i servizi BeeGFS:

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

3. Verificare che i servizi del nodo siano esauriti eseguendo:

```
pcs status
```

   Assicurarsi che non vengano segnalati servizi come `Started` sul nodo in standby.

   A seconda delle dimensioni del cluster, possono essere necessari secondi o minuti per lo spostamento dei servizi nel nodo sorella. Se un servizio BeeGFS non si avvia sul nodo gemellato, fare riferimento a "Guide per la risoluzione dei problemi".

4. Arrestare il cluster sul nodo:

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

5. Aggiornare i pacchetti pacemaker, Corosync e pz sul nodo:

   I comandi del gestore dei pacchetti variano a seconda del sistema operativo. I seguenti comandi si riferiscono ai sistemi che eseguono RHEL 8 e successivi.

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

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

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

6. Avviare i servizi del gruppo pacemaker sul nodo:

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

7. Se il `pcs` pacchetto è stato aggiornato, autenticare nuovamente il nodo con il cluster:

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

8. Verificare che la configurazione del pacemaker sia ancora valida con `crm_verify` lo strumento.

   Questa operazione deve essere verificata solo una volta durante l'upgrade del cluster.

```
crm_verify -L -V
```

9. Porta il nodo fuori dallo standby:

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

10. Riposizionare tutti i servizi BeeGFS nel nodo preferito:

```
pcs resource relocate run
```

11. Ripetere i passaggi precedenti per ciascun nodo del cluster finché tutti i nodi non eseguono le versioni Pacemaker, Corosync e pz desiderate.
12. Infine, eseguire `pcs status` e verificare che il quadro strumenti sia in buone condizioni e `Current DC` riporta la versione pacemaker desiderata.

   Se il `Current DC` report "versione fissa", un nodo nel quadro strumenti è ancora in esecuzione con la versione precedente di pacemaker e deve essere aggiornato. Se un nodo aggiornato non è in grado di riconnettersi al cluster o se le risorse non si avviano, controllare i registri del cluster e consultare le note di rilascio del pacemaker o le guide dell'utente per problemi noti relativi all'aggiornamento.

Arresto completo del cluster

Con questo approccio, tutti i nodi e le risorse del cluster vengono arrestati, i nodi vengono aggiornati e il cluster viene riavviato. Questo approccio è necessario se le versioni Pacemaker e Corosync non supportano una configurazione a versione mista.

1. Verificare che il cluster sia in uno stato ottimale, con ogni servizio BeeGFS in esecuzione sul nodo preferito. Per ulteriori informazioni, fare riferimento alla "Esaminare lo stato del cluster" sezione.
2. Arrestare il software del cluster (pacemaker e Corosync) su tutti i nodi.

   A seconda delle dimensioni del cluster, l'arresto dell'intero cluster può richiedere secondi o minuti.

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

3. Una volta arrestati i servizi cluster su tutti i nodi, aggiornare i pacchetti pacemaker, Corosync e pz su ciascun nodo in base alle proprie esigenze.

> I comandi del gestore dei pacchetti variano a seconda del sistema operativo. I seguenti comandi si riferiscono ai sistemi che eseguono RHEL 8 e successivi.

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

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

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

4. Dopo aver eseguito l'upgrade di tutti i nodi, avviare il software cluster su tutti i nodi:

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

5. Se il `pcs` pacchetto è stato aggiornato, eseguire nuovamente l'autenticazione di ciascun nodo nel cluster:

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

6. Infine, eseguire `pcs status` e verificare che il quadro strumenti funzioni correttamente e `Current DC` riporta la versione pacemaker corretta.

> Se il `Current DC` report "versione fissa", un nodo nel quadro strumenti è ancora in esecuzione con la versione precedente di pacemaker e deve essere aggiornato.

Aggiornare il firmware dell'adattatore del nodo del file

Per aggiornare gli adattatori ConnectX-7 del nodo file al firmware più recente, procedere come segue.

Panoramica

Potrebbe essere necessario aggiornare il firmware della scheda ConnectX-7 per supportare un nuovo driver MLNX_OFED, abilitare nuove funzioni o correggere bug. Questa guida utilizzerà l'utilità di NVIDIA `mlxfwmanager` per gli aggiornamenti delle schede, grazie alla sua facilità d'uso ed efficienza.

Considerazioni sull'upgrade

In questa guida vengono descritti due approcci per l'aggiornamento del firmware della scheda ConnectX-7: Un aggiornamento in corso e un aggiornamento del cluster a due nodi. Scegliere l'approccio di aggiornamento appropriato in base alle dimensioni del cluster. Prima di eseguire gli aggiornamenti del firmware, verificare che:

- È installato un driver MLNX_OFED supportato. Fare riferimento alla "requisiti tecnologici".
- Esistono backup validi per il file system BeeGFS e la configurazione del cluster pacemaker.
- Il cluster è in uno stato sano.

Preparazione dell'aggiornamento del firmware

Si consiglia di utilizzare l'utilità di NVIDIA `mlxfwmanager` per aggiornare il firmware dell'adattatore di un nodo, fornito con il driver MLNX_OFED di NVIDIA. Prima di avviare gli aggiornamenti, scaricare l'immagine del firmware della scheda da "Sito di supporto di NVIDIA" e memorizzarla su ciascun nodo file.

Per gli adattatori Lenovo ConnectX-7, utilizzare `mlxfwmanager_LES` lo strumento, disponibile alla pagina di NVIDIA"Firmware OEM".

Approccio di aggiornamento continuo

Questo approccio è consigliato per qualsiasi cluster ha con più di due nodi. Questo approccio implica l'aggiornamento del firmware dell'adattatore su un file nodo alla volta, consentendo al cluster ha di mantenere le richieste di assistenza, anche se durante questo periodo si consiglia di non eseguire interventi di i/O.

1. Verificare che il cluster sia in uno stato ottimale, con ogni servizio BeeGFS in esecuzione sul nodo preferito. Per ulteriori informazioni, fare riferimento alla "Esaminare lo stato del cluster" sezione.
2. Scegliere un nodo file da aggiornare e impostarlo in modalità standby, che svuota (o sposta) tutti i servizi BeeGFS da quel nodo:

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

3. Verificare che i servizi del nodo siano esauriti eseguendo:

```
pcs status
```

   Verificare che non vi siano servizi che segnalano come `Started` sul nodo in standby.

A seconda delle dimensioni del cluster, lo spostamento dei servizi BeeGFS nel nodo sorella può richiedere secondi o minuti. Se un servizio BeeGFS non si avvia sul nodo gemellato, fare riferimento a "Guide per la risoluzione dei problemi".

4. Aggiornare il firmware dell'adattatore utilizzando `mlxfwmanager`.

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

Tenere presente `PCI Device Name` per ciascun adattatore che riceve gli aggiornamenti del firmware.

5. Ripristinare ciascuna scheda di rete utilizzando l' `mlxfwreset`utilità per applicare il nuovo firmware.

> Alcuni aggiornamenti del firmware potrebbero richiedere un riavvio per applicare l'aggiornamento. Fare riferimento alla "Le limitazioni di mlxfwreset di NVIDIA" per le istruzioni. Se è necessario riavviare il sistema, riavviare il sistema invece di reimpostare gli adattatori.

a. Arrestare il servizio opensm:

```
systemctl stop opensm
```

b. Eseguire il seguente comando per ognuno di quelli `PCI Device Name` annotati in precedenza.

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

c. Avviare il servizio opensm:

```
systemctl start opensm
```

d. Riavviare il `eseries_nvme_ib.service` .

```
systemctl restart eseries_nvme_ib.service
```

e. Verificare che i volumi dell'array di archiviazione E-Series siano presenti.

```
multipath -ll
```

1. Eseguire `ibstat` e verificare che tutti gli adattatori funzionino alla versione firmware desiderata:

```
ibstat
```

2. Avviare i servizi del gruppo pacemaker sul nodo:

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

3. Porta il nodo fuori dallo standby:

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

4. Riposizionare tutti i servizi BeeGFS nel nodo preferito:

```
pcs resource relocate run
```

Ripetere questi passaggi per ciascun nodo file nel cluster fino a quando tutte le schede di rete non sono state aggiornate.

Approccio all'update del cluster a due nodi

Questo approccio è consigliato per i cluster ha con solo due nodi. Questo approccio è simile a un aggiornamento in corso, ma include passaggi aggiuntivi per evitare tempi di inattività del servizio quando i servizi cluster di un nodo vengono arrestati.

1. Verificare che il cluster sia in uno stato ottimale, con ogni servizio BeeGFS in esecuzione sul nodo preferito. Per ulteriori informazioni, fare riferimento alla "Esaminare lo stato del cluster" sezione.
2. Scegliere un nodo file da aggiornare e posizionare il nodo in modalità standby, che svuota (o sposta) tutti i servizi BeeGFS da quel nodo:

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

3. Verificare che le risorse del nodo siano esaurite eseguendo:

```
pcs status
```

Verificare che non vi siano servizi che segnalano come `Started` sul nodo in standby.

> A seconda delle dimensioni del cluster, possono essere necessari secondi o minuti affinché i servizi BeeGFS eseguano il report come `Started` sul nodo sorella. Se un servizio BeeGFS non si avvia, fare riferimento alla "Guide per la risoluzione dei problemi".

4. Portare il quadro strumenti in modalità di manutenzione.

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

5. Aggiornare il firmware dell'adattatore utilizzando `mlxfwmanager`.

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

Tenere presente `PCI Device Name` per ciascun adattatore che riceve gli aggiornamenti del firmware.

6. Ripristinare ciascuna scheda di rete utilizzando l' `mlxfwreset`utilità per applicare il nuovo firmware.

> Alcuni aggiornamenti del firmware potrebbero richiedere un riavvio per applicare l'aggiornamento. Fare riferimento alla "Le limitazioni di mlxfwreset di NVIDIA" per le istruzioni. Se è necessario riavviare il sistema, riavviare il sistema invece di reimpostare gli adattatori.

   a. Arrestare il servizio opensm:

```
systemctl stop opensm
```

   b. Eseguire il seguente comando per ognuno di quelli `PCI Device Name` annotati in precedenza.

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

   c. Avviare il servizio opensm:

```
systemctl start opensm
```

7. Eseguire `ibstat` e verificare che tutti gli adattatori funzionino alla versione firmware desiderata:

```
ibstat
```

8. Avviare i servizi del gruppo pacemaker sul nodo:

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

9. Porta il nodo fuori dallo standby:

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

10. Portare il quadro strumenti fuori dalla modalità di manutenzione.

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

11. Riposizionare tutti i servizi BeeGFS nel nodo preferito:

```
pcs resource relocate run
```

Ripetere questi passaggi per ciascun nodo file nel cluster fino a quando tutte le schede di rete non sono state aggiornate.

Upgrade dello storage array E-Series

Per aggiornare i componenti dello storage array e-Series del cluster ha, procedere come segue.

Panoramica

Mantenere aggiornati gli array di storage NetApp E-Series del cluster ha con il firmware più recente garantisce prestazioni ottimali e maggiore sicurezza. Gli aggiornamenti del firmware per l'array di storage vengono applicati attraverso il sistema operativo SANtricity, l'NVSRAM e i file del firmware del disco.

> Durante gli upgrade degli storage array con il cluster ha online, si consiglia di impostare il cluster in modalità di manutenzione per tutti gli upgrade.

Passaggi di aggiornamento del nodo a blocchi

I seguenti passaggi descrivono come aggiornare il firmware degli storage array utilizzando la `Netapp_Eseries.Santricity` raccolta Ansible. Prima di continuare, consultare "Considerazioni sull'upgrade"per l'aggiornamento dei sistemi E-Series.

> L'aggiornamento a SANtricity OS 11,80 o versioni successive è possibile solo a partire da 11.70.5P1. Prima di eseguire ulteriori upgrade, lo storage array deve essere aggiornato a 11.70.5P1.

1. Conferma che il tuo nodo di controllo Ansible sta utilizzando la raccolta Ansible SANtricity più recente.
   - Per gli aggiornamenti del ritiro con accesso a. "Ansible Galaxy", eseguire il seguente comando:

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

   - Per gli aggiornamenti offline, scaricare il tarball "Ansible Galaxy"della raccolta da , trasferirlo al nodo di controllo ed eseguire:

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

     Vedere "Installazione delle raccolte" per ulteriori informazioni.

2. Procurarsi il firmware più recente per l'array di archiviazione e le unità.
   a. Scaricare i file del firmware.
      - **SANtricity OS e NVSRAM:** accedere alla "Sito di supporto NetApp"e scaricare la versione più recente del sistema operativo SANtricity e NVSRAM per il modello di array di storage.
      - **Firmware dell'unità:** accedere a "Sito del firmware del disco E-Series"e scaricare il firmware più recente per ciascun modello di unità dell'array di archiviazione.

b. Memorizza i file del firmware del disco, NVSRAM e del sistema operativo SANtricity nella `<inventory_directory>/packages` directory del nodo di controllo Ansible.

3. Se necessario, aggiorna i file di inventario Ansible del tuo cluster per includere tutti gli storage array (nodi a blocchi) che richiedono aggiornamenti. Per indicazioni, vedere la "Panoramica di Ansible Inventory" sezione.
4. Assicurarsi che il cluster sia in uno stato ottimale con ogni servizio BeeGFS sul nodo preferito. Per ulteriori informazioni, fare riferimento alla "Esaminare lo stato del cluster" sezione.
5. Portare il quadro strumenti in modalità di manutenzione seguendo le istruzioni riportate in "Impostare il cluster in modalità di manutenzione".
6. Crea un nuovo playbook Ansible chiamato `update_block_node_playbook.yml`. Popola il playbook con i seguenti contenuti, sostituendo il sistema operativo SANtricity, NVSRAM e le versioni del firmware del disco nel percorso di upgrade desiderato:

```
- 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. Per avviare gli aggiornamenti, esegui il seguente comando dal nodo di controllo Ansible:

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

8. Una volta completato il playbook, verifica che ciascuno storage array si trovi in uno stato ottimale.
9. Spostare il cluster dalla modalità di manutenzione e verificare che sia in uno stato ottimale con ogni servizio BeeGFS sul nodo preferito.

Informazioni sul copyright

Informazioni sul marchio commerciale