



Amministrare i cluster BeeGFS

BeeGFS on NetApp with E-Series Storage

NetApp

January 27, 2026

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

Sommario

Amministrare i cluster BeeGFS	1
Panoramica, concetti chiave e terminologia	1
Panoramica	1
Concetti chiave	1
Terminologia comune	2
Quando utilizzare Ansible rispetto allo strumento PC	2
Esaminare lo stato del cluster	2
Panoramica	3
Comprendere l'output di <code>pcs status</code>	3
Riconfigurare il cluster ha e BeeGFS	4
Panoramica	4
Come disattivare e attivare la funzione di schermo	4
Update dei componenti del cluster ha	5
Aggiorna i servizi BeeGFS	5
Aggiorna a BeeGFS v8	8
Aggiornare i pacchetti pacemaker e Corosync in un cluster ha	18
Aggiornare il firmware dell'adattatore del nodo del file	21
Upgrade dello storage array E-Series	26
Assistenza e manutenzione	27
Servizi di failover e failback	27
Impostare il cluster in modalità di manutenzione	30
Arrestare e avviare il cluster	31
Sostituire i nodi del file	32
Espandere o ridurre il cluster	33
Risolvere i problemi	34
Panoramica	34
Guide per la risoluzione dei problemi	34
Problemi comuni	39
Attività comuni di risoluzione dei problemi	39

Amministrare i cluster BeeGFS

Panoramica, concetti chiave e terminologia

Scopri come amministrare i cluster BeeGFS ha dopo che sono stati implementati.

Panoramica

Questa sezione è destinata agli amministratori dei cluster che devono gestire i cluster BeeGFS ha dopo la loro implementazione. Anche coloro che hanno familiarità con i cluster ha Linux dovrebbero leggere attentamente questa guida, poiché esistono diverse differenze nella gestione del cluster, in particolare per quanto riguarda la riconfigurazione dovuta all'utilizzo di Ansible.

Concetti chiave

Anche se alcuni di questi concetti sono stati introdotti nella ["termini e concetti"](#) pagina principale, è utile reintrodurli nel contesto di un cluster BeeGFS ha:

Nodo cluster: un server che esegue i servizi Pacemaker e Corosync e che partecipa al cluster ha.

Nodo file: nodo cluster utilizzato per eseguire uno o più servizi di gestione, metadati o storage BeeGFS.

Nodo a blocchi: un sistema storage NetApp e-Series che fornisce storage a blocchi ai nodi di file. Questi nodi non fanno parte del cluster BeeGFS ha in quanto offrono funzionalità ha standalone proprie. Ciascun nodo è costituito da due storage controller che forniscono alta disponibilità al livello di blocco.

Servizio BeeGFS: Un servizio di gestione, metadati o storage BeeGFS. Ogni nodo di file eseguirà uno o più servizi che utilizzeranno i volumi sul nodo di blocco per memorizzare i propri dati.

Building Block: implementazione standardizzata di file node BeeGFS, nodi a blocchi e-Series e servizi BeeGFS in esecuzione su di essi che semplifica la scalabilità di un cluster/file system BeeGFS ha in base a un'architettura verificata di NetApp. Sono supportati anche i cluster ha personalizzati, ma spesso seguono un approccio simile a building block per semplificare la scalabilità.

Cluster BeeGFS ha: un numero scalabile di nodi di file utilizzati per eseguire i servizi BeeGFS supportati da nodi a blocchi per memorizzare i dati BeeGFS in modo altamente disponibile. Basato su componenti open-source collaudati nel settore, Pacemaker e Corosync utilizzano Ansible per il packaging e l'implementazione.

Servizi cluster: si riferisce ai servizi Pacemaker e Corosync in esecuzione su ciascun nodo che partecipa al cluster. Nota è possibile che un nodo non esegua alcun servizio BeeGFS e partecipi al cluster come nodo "Tiebreaker" nel caso in cui vi sia solo la necessità di due nodi di file.

Risorse del cluster: per ogni servizio BeeGFS in esecuzione nel cluster vengono visualizzate una risorsa di monitoraggio BeeGFS e un gruppo di risorse contenente risorse per destinazioni BeeGFS, indirizzi IP (IP mobili) e il servizio BeeGFS stesso.

Ansible: Uno strumento per il provisioning del software, la gestione della configurazione e l'implementazione delle applicazioni, che consente l'infrastruttura come codice. È il modo in cui i cluster BeeGFS ha vengono confezionati per semplificare il processo di implementazione, riconfigurazione e aggiornamento di BeeGFS su NetApp.

Pcs: interfaccia a riga di comando disponibile da qualsiasi nodo di file nel cluster utilizzato per eseguire query e controllare lo stato dei nodi e delle risorse nel cluster.

Terminologia comune

Failover: ciascun servizio BeeGFS dispone di un nodo di file preferito su cui verrà eseguito, a meno che tale nodo non si guasti. Quando un servizio BeeGFS viene eseguito su un nodo di file non preferito/secondario, si dice che sia in failover.

Failover: l'atto di spostare i servizi BeeGFS da un nodo di file non preferito al nodo preferito.

Coppia ha: due nodi di file che possono accedere allo stesso insieme di nodi a blocchi sono talvolta indicati come coppia ha. Si tratta di un termine comune utilizzato in NetApp per fare riferimento a due controller o nodi storage che possono "assumere" l'uno con l'altro.

Modalità di manutenzione: disabilita il monitoraggio di tutte le risorse e impedisce al pacemaker di spostare o gestire in altro modo le risorse nel cluster (vedere anche la sezione a ["modalità di manutenzione"](#)).

Cluster ha: uno o più file node che eseguono servizi BeeGFS che possono eseguire il failover tra più nodi nel cluster per creare un file system BeeGFS ad alta disponibilità. Spesso i file node sono configurati in coppie ha in grado di eseguire un sottoinsieme dei servizi BeeGFS nel cluster.

Quando utilizzare Ansible rispetto allo strumento PC

Quando si deve utilizzare Ansible rispetto allo strumento della riga di comando di PC per gestire il cluster ha?

Tutte le attività di implementazione e riconfigurazione del cluster devono essere completate utilizzando Ansible da un nodo di controllo Ansible esterno. Le modifiche temporanee nello stato del cluster (ad esempio, l'inserimento e l'uscita dei nodi in standby) vengono eseguite in genere accedendo a un nodo del cluster (preferibilmente uno che non è degradato o che sta per essere sottoposto a manutenzione) e utilizzando la riga di comando di PC.

La modifica di qualsiasi configurazione del cluster, incluse risorse, vincoli, proprietà e i servizi BeeGFS stessi, deve essere eseguita sempre utilizzando Ansible. La manutenzione di una copia aggiornata dell'inventario e del playbook Ansible (idealmente nel controllo del codice sorgente per tenere traccia delle modifiche) fa parte della manutenzione del cluster. Quando è necessario apportare modifiche alla configurazione, aggiornare l'inventario ed eseguire nuovamente il playbook Ansible che importa il ruolo BeeGFS ha.

Il ruolo ha gestirà il posizionamento del cluster in modalità di manutenzione e l'esecuzione delle modifiche necessarie prima di riavviare BeeGFS o i servizi cluster per applicare la nuova configurazione. Poiché i riavvii completi dei nodi non sono generalmente necessari al di fuori dell'implementazione iniziale, il rerunning di Ansible è generalmente considerato una procedura "sicura", ma è sempre consigliato durante le finestre di manutenzione o fuori orario nel caso in cui sia necessario riavviare qualsiasi servizio BeeGFS. In genere, questi riavvii non dovrebbero causare errori nelle applicazioni, ma potrebbero compromettere le prestazioni (che alcune applicazioni potrebbero gestire meglio di altre).

Il rerunning di Ansible è anche un'opzione quando si desidera riportare l'intero cluster a uno stato completamente ottimale e in alcuni casi potrebbe essere in grado di ripristinare lo stato del cluster più facilmente rispetto all'utilizzo di PC. Soprattutto in caso di emergenza in cui il cluster non è attivo per qualche motivo, una volta eseguito il backup di tutti i nodi, Ansible può ripristinare il cluster in modo più rapido e affidabile rispetto al tentativo di utilizzare i PC.

Esaminare lo stato del cluster

Utilizzare i PC per visualizzare lo stato del cluster.

Panoramica

In esecuzione `pcs status` Da qualsiasi nodo del cluster è il modo più semplice per visualizzare lo stato complessivo del cluster e lo stato di ciascuna risorsa (ad esempio i servizi BeeGFS e le relative dipendenze). In questa sezione vengono illustrate le informazioni disponibili nell'output di `pcs status` comando.

Comprendere l'output di `pcs status`

Eseguire `pcs status` Su qualsiasi nodo del cluster in cui vengono avviati i servizi del cluster (pacemaker e Corosync). La parte superiore dell'output mostra un riepilogo del cluster:

```
[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
```

La sezione seguente elenca i nodi nel cluster:

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

Ciò indica in particolare i nodi in standby o offline. I nodi in standby partecipano ancora al cluster ma sono contrassegnati come non idonei per l'esecuzione delle risorse. I nodi offline indicano che i servizi cluster non sono in esecuzione su quel nodo, a causa di un arresto manuale o perché il nodo è stato riavviato/arrestato.



Quando i nodi si avviano per la prima volta, i servizi del cluster vengono arrestati e devono essere avviati manualmente per evitare il failover accidentale delle risorse su un nodo non integro.

Se i nodi sono in standby o non in linea a causa di un motivo non amministrativo (ad esempio un errore), accanto allo stato del nodo viene visualizzato un testo aggiuntivo tra parentesi. Ad esempio, se la funzione di schermo è disattivata e una risorsa rileva un errore, viene visualizzato `Node <HOSTNAME>: standby (on-fail)`. Un altro stato possibile è `Node <HOSTNAME>: UNCLEAN (offline)`, che sarà visto brevemente come un nodo è in fase di recintamento, ma continuerà se la schermata non è riuscita, indicando che il cluster non può confermare lo stato del nodo (questo può bloccare l'avvio delle risorse su altri nodi).

La sezione successiva mostra un elenco di tutte le risorse del cluster e dei relativi stati:

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-mgmd):            Started beegfs_01
[...]
```

In modo simile ai nodi, viene visualizzato un testo aggiuntivo accanto allo stato della risorsa tra parentesi in caso di problemi con la risorsa. Ad esempio, se il pacemaker richiede un arresto della risorsa e non riesce a completarlo entro il tempo assegnato, il pacemaker tenterà di individuare il nodo. Se la funzione di schermo è disattivata o l'operazione di schermo non riesce, lo stato della risorsa sarà `FAILED <HOSTNAME> (blocked)`. E Pacemaker non sarà in grado di avviarlo su un nodo diverso.

Vale la pena notare che i cluster BeeGFS ha utilizzano diversi agenti di risorse OCF personalizzati ottimizzati per BeeGFS. In particolare, il monitor BeeGFS è responsabile dell'attivazione di un failover quando le risorse BeeGFS su un nodo specifico non sono disponibili.

Riconfigurare il cluster ha e BeeGFS

Utilizzare Ansible per riconfigurare il cluster.

Panoramica

In genere, la riconfigurazione di qualsiasi aspetto del cluster ha BeeGFS deve essere eseguita aggiornando l'inventario Ansible ed eseguendo nuovamente il `ansible-playbook` comando. Ciò include l'aggiornamento degli avvisi, la modifica della configurazione di schermo permanente o la regolazione della configurazione del servizio BeeGFS. Questi vengono regolati utilizzando il `group_vars/ha_cluster.yml` file e un elenco completo delle opzioni è disponibile nella ["Specificare la configurazione del nodo file comune"](#) sezione .

Per ulteriori informazioni sulle opzioni di configurazione selezionate di cui gli amministratori devono essere a conoscenza durante la manutenzione o la manutenzione del cluster, vedere di seguito.

Come disattivare e attivare la funzione di schermo

Per impostazione predefinita, la funzione di schermo viene attivata/richiesta durante la configurazione del cluster. In alcuni casi, potrebbe essere consigliabile disattivare temporaneamente la schermo per garantire che i nodi non vengano accidentalmente arrestati durante determinate operazioni di manutenzione (ad esempio l'aggiornamento del sistema operativo). Anche se questa funzione può essere disattivata manualmente, gli amministratori devono tenere presente che esistono compromessi.

OPZIONE 1: Disattiva schermo utilizzando Ansible (consigliato).

Quando la funzione di schermo viene disattivata utilizzando Ansible, l'azione on-fail del monitor BeeGFS passa da "fence" a "standby". Ciò significa che se il monitor BeeGFS rileva un errore, tenterà di mettere il nodo in standby e di eseguire il failover di tutti i servizi BeeGFS. Al di fuori del troubleshooting/test attivo, questo è in genere più desiderabile dell'opzione 2. Lo svantaggio è che se una risorsa non si ferma sul nodo originale, viene impedita l'avvio da un'altra parte (motivo per cui la schermo è generalmente richiesta per i cluster di produzione).

1. Nel tuo inventario Ansible all'indirizzo `groups_vars/ha_cluster.yml` aggiungere la seguente configurazione:

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

2. Rieseguire il manuale Ansible per applicare le modifiche al cluster.

OPZIONE 2: Disattiva manualmente la funzione di schermo.

In alcuni casi, potrebbe essere necessario disattivare temporaneamente la schermata senza eseguire nuovamente Ansible, ad esempio per facilitare la risoluzione dei problemi o il test del cluster.



In questa configurazione, se il monitor BeeGFS rileva un errore, il cluster tenta di arrestare il gruppo di risorse corrispondente. Non attiverà un failover completo né tenterà di riavviare o spostare il gruppo di risorse interessato in un altro host. Per risolvere il problema, risolvere i problemi che si verificano in seguito `pcs resource cleanup` oppure mettere manualmente il nodo in standby.

Fasi:

1. Per determinare se la schermata (stonith) è attivata o disattivata globalmente, eseguire: `pcs property show stonith-enabled`
2. Per disattivare la funzione di schermo: `pcs property set stonith-enabled=false`
3. Per attivare la funzione di schermo: `pcs property set stonith-enabled=true`



Questa impostazione verrà sovrascritta alla successiva esecuzione del playbook Ansible.

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	Multirail	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-mgmtbd beegfs-storage beegfs-meta libbeegfs-ib
```



In un cluster standard, il `beegfs-mgmtbd` 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-mgmtbd --import-from  
-v7=/mnt/mgmt_tgt_mgmt01/data/
```

Output previsto:

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

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

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

La preparazione del file di configurazione `beegfs-mgmt.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-mgmt.toml`.

1. Sul nodo di gestione con il target di gestione montato, fare riferimento al `/mnt/mgmt_tgt_mgmt01/mgmt_config/beegfs-mgmt.conf` file del servizio di gestione per BeeGFS 7, quindi trasferire tutte le impostazioni al `/mnt/mgmt_tgt_mgmt01/mgmt_config/beegfs-mgmt.toml` file. Per una configurazione di base, il tuo `beegfs-mgmt.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/mgmt.d.sqlite"
license-disable = false
license-cert-file = "/etc/beegfs/license.pem"
tls-disable = false
tls-cert-file = "/etc/beegfs/mgmt.d_tls_cert.pem"
tls-key-file = "/etc/beegfs/mgmt.d_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-mgmt --config-file
/mnt/mgmt_tgt_mgmt01/mgmt_config/beegfs-mgmt.d.toml|'
/etc/systemd/system/beegfs-mgmt.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-mgmt.d|.conf|mgmt_config/beegfs-mgmt.d.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-mgmt.d.conf"  
        ;;
```

E sostituiscilo con:

```
case $type in  
    management)  
        conf_path="{configuration_mount}/mgmt_config/beegfs-mgmt.d.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-mgmt` servizio sia stato avviato correttamente:

```
journalctl -xeu beegfs-mgmt
```

L'output previsto include righe come:

```
Started Cluster Controlled beegfs-mgmt.  
Loaded config file from "/mnt/mgmt_tgt_mgmt01/mgmt_config/beegfs-  
mgmt.toml"  
Successfully initialized certificate verification library.  
Successfully loaded license certificate: TMP-113489268  
Opened database at "/mnt/mgmt_tgt_mgmt01/data/mgmt.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-mgmt.d.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_nvram: "packages/<NVSRAM>.dlp"
  eseries_drive_firmware_firmware_list:
    - "packages/<drive_firmware>.dlp"
  eseries_drive_firmware_upgrade_drives_online: true

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

7. 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.

Assistenza e manutenzione

Servizi di failover e failback

Spostamento dei servizi BeeGFS tra nodi cluster.

Panoramica

I servizi BeeGFS possono eseguire il failover tra i nodi del cluster per garantire che i client possano continuare ad accedere al file system in caso di guasto di un nodo o se è necessario eseguire una manutenzione

pianificata. In questa sezione vengono descritti i vari modi in cui gli amministratori possono riparare il cluster dopo il ripristino da un errore o spostare manualmente i servizi tra i nodi.

Fasi

Failover e failover

Failover (pianificato)

In genere, quando si deve portare un singolo nodo di file offline per la manutenzione, si desidera spostare (o svuotare) tutti i servizi BeeGFS da quel nodo. Per eseguire questa operazione, mettere il nodo in standby:

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

Dopo aver verificato l'utilizzo `pcs status` tutte le risorse sono state riavviate sul nodo di file alternativo, è possibile chiudere o apportare altre modifiche al nodo in base alle necessità.

Failback (dopo un failover pianificato)

Quando si è pronti a ripristinare i servizi BeeGFS sul nodo preferito, eseguire prima `pcs status` E verificare in "Node List" (elenco nodi) che lo stato sia standby. Se il nodo è stato riavviato, viene visualizzato offline fino a quando non si mettono in linea i servizi del cluster:

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

Una volta che il nodo è online, portarlo fuori dallo standby con:

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

Infine, ricollocare tutti i servizi BeeGFS nei nodi preferiti con:

```
pcs resource relocate run
```

Failback (dopo un failover non pianificato)

Se un nodo presenta un guasto hardware o di altro tipo, il cluster ha dovrebbe reagire automaticamente e spostare i propri servizi su un nodo integro, fornendo tempo agli amministratori per intraprendere azioni correttive. Prima di procedere "[risoluzione dei problemi](#)", fare riferimento alla sezione per determinare la causa del failover e risolvere eventuali problemi in sospeso. Una volta riaccesso il nodo e funzionante, è possibile procedere con il failback.

Quando un nodo viene avviato in seguito a un riavvio non pianificato (o pianificato), i servizi cluster non vengono impostati per avviarsi automaticamente, quindi è necessario prima portare il nodo online con:

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

Quindi, ripulire gli eventuali errori delle risorse e reimpostare la cronologia delle schermata del nodo:

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

Verificare in `pcs status` il nodo è online e integro. Per impostazione predefinita, i servizi BeeGFS non eseguono automaticamente il failback per evitare di spostare accidentalmente le risorse in un nodo non integro. Quando si è pronti, restituire tutte le risorse del cluster ai nodi preferiti con:

```
pcs resource relocate run
```

Spostamento di singoli servizi BeeGFS in nodi di file alternativi

Spostare in modo permanente un servizio BeeGFS in un nuovo nodo di file

Se si desidera modificare in modo permanente il nodo di file preferito per un singolo servizio BeeGFS, regolare l'inventario Ansible in modo che il nodo preferito venga elencato per primo ed eseguire nuovamente il playbook Ansible.

Ad esempio, in questo file di esempio `inventory.yml`, `beegfs_01` è il nodo file preferito per eseguire il servizio di gestione BeeGFS:

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

L'inversione dell'ordine causerebbe la preferenza dei servizi di gestione su `beegfs_02`:

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

Spostare temporaneamente un servizio BeeGFS in un nodo di file alternativo

In genere, se un nodo è in fase di manutenzione, utilizzare i [passi di failover e failback] (failover e failback) per spostare tutti i servizi da quel nodo.

Se per qualche motivo è necessario spostare un singolo servizio in un nodo di file diverso, eseguire:

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



Non specificare singole risorse o il gruppo di risorse. Specificare sempre il nome del monitor per il servizio BeeGFS che si desidera trasferire. Ad esempio, per spostare il servizio di gestione BeeGFS in `beegfs_02` eseguire: `pcs resource move mgmt-monitor beegfs_02`. Questo processo può essere ripetuto per spostare uno o più servizi lontano dai nodi preferiti. Verificare che l'utilizzo dei `pcs status` servizi sia stato ricollocato/avviato sul nuovo nodo.

Per spostare di nuovo un servizio BeeGFS nel nodo preferito, eliminare prima i vincoli di risorsa temporanei (ripetendo questa operazione in base alle necessità per più servizi):

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

Quindi, quando si è pronti a spostare di nuovo i servizi sui nodi preferiti, eseguire:

```
pcs resource relocate run
```

Nota: Questo comando consente di spostare i servizi che non hanno più vincoli di risorse temporanee e che non si trovano nei nodi preferiti.

Impostare il cluster in modalità di manutenzione

Evitare che il cluster ha reagisca accidentalmente alle modifiche previste nell'ambiente.

Panoramica

Impostando il cluster in modalità di manutenzione si disattiva il monitoraggio di tutte le risorse e si impedisce a Pacemaker di spostare o gestire in altro modo le risorse nel cluster. Tutte le risorse rimarranno in esecuzione sui nodi originali, indipendentemente dalla presenza di una condizione di guasto temporanea che ne impedirebbe l'accesso. Gli scenari in cui questo è consigliato/utile includono:

- Manutenzione della rete che potrebbe interrompere temporaneamente le connessioni tra i nodi di file e i servizi BeeGFS.
- Aggiornamenti del nodo a blocchi.
- File Node per aggiornamenti di sistemi operativi, kernel o altri pacchetti.

In genere, l'unico motivo per attivare manualmente la modalità di manutenzione è impedire che il cluster reagisca alle modifiche esterne dell'ambiente. Se un singolo nodo del cluster richiede una riparazione fisica, non utilizzare la modalità di manutenzione e posizionare semplicemente tale nodo in standby seguendo la procedura descritta in precedenza. Si noti che la riesecuzione di Ansible attiva automaticamente la modalità di manutenzione del cluster, facilitando la maggior parte della manutenzione del software, inclusi aggiornamenti e modifiche alla configurazione.

Fasi

Per verificare se il cluster è in modalità di manutenzione, eseguire:

```
pcs property config
```


La `maintenance-mode` proprietà non viene visualizzata se il cluster funziona normalmente. Se il cluster è attualmente in modalità di manutenzione, la proprietà visualizzerà come `true`. Per attivare la modalità di manutenzione, eseguire:

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

Puoi verificare eseguendo lo stato dei PC e assicurandoti che tutte le risorse mostrino "(unmanaged)". Per uscire dalla modalità di manutenzione del cluster, eseguire:

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

Arrestare e avviare il cluster

Arresto e avvio del cluster ha senza problemi.

Panoramica

In questa sezione viene descritto come arrestare e riavviare il cluster BeeGFS. Esempi di scenari in cui ciò potrebbe essere necessario includono la manutenzione elettrica o la migrazione tra datacenter o rack.

Fasi

Se per qualsiasi motivo è necessario arrestare l'intero cluster BeeGFS e arrestare tutti i servizi eseguiti:

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

È anche possibile arrestare il cluster su singoli nodi (che eseguiranno automaticamente il failover dei servizi su un altro nodo), anche se si consiglia di mettere il nodo in standby (vedere la ["failover"](#) sezione):

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

Per avviare i servizi e le risorse del cluster su tutti i nodi eseguire:

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

Oppure avviare i servizi su un nodo specifico con:

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

A questo punto eseguire `pcs status` Verificare inoltre che i servizi del cluster e BeeGFS vengano avviati su tutti i nodi e che i servizi siano in esecuzione sui nodi previsti.



A seconda delle dimensioni del cluster, l'arresto dell'intero cluster può richiedere secondi o minuti o la visualizzazione come avviato in `pcs status`. se `pcs cluster <COMMAND>` si blocca per più di cinque minuti, prima di eseguire "Ctrl+C" per annullare il comando, accedere a ciascun nodo del cluster e utilizzare `pcs status` per verificare se i servizi cluster (Corosync/Pacemaker) sono ancora in esecuzione su quel nodo. Da qualsiasi nodo in cui il cluster è ancora attivo, è possibile controllare quali risorse bloccano il cluster. Risolvere manualmente il problema e il comando dovrebbe essere completo o può essere rieseguito per interrompere eventuali servizi rimanenti.

Sostituire i nodi del file

Sostituzione di un nodo di file se il server originale è guasto.

Panoramica

Di seguito viene fornita una panoramica dei passaggi necessari per sostituire un nodo di file nel cluster. Questi passaggi presumono che il nodo del file non sia riuscito a causa di un problema hardware ed è stato sostituito con un nuovo nodo del file identico.

Fasi:

1. Sostituire fisicamente il nodo del file e ripristinare tutti i cavi al nodo a blocchi e alla rete di storage.
2. Reinstallare il sistema operativo sul nodo di file, aggiungendo anche le sottoscrizioni Red Hat.
3. Configurare la gestione e la rete BMC sul nodo file.
4. Aggiornare l'inventario di Ansible se il nome host, l'IP, le mappature dell'interfaccia PCIe-to-logical o qualsiasi altra cosa è stata modificata in relazione al nuovo nodo del file. In genere, questo non è necessario se il nodo è stato sostituito con un hardware server identico e si sta utilizzando la configurazione di rete originale.
 - a. Ad esempio, se il nome host è cambiato, creare (o rinominare) il file di inventario del nodo (`host_vars/<NEW_NODE>.yaml`) Quindi nel file di inventario Ansible (`inventory.yaml`), sostituire il nome del vecchio nodo con il nuovo nome del nodo:

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

5. Rimuovere il nodo precedente da uno degli altri nodi del cluster: `pcs cluster node remove <HOSTNAME>`.



NON PROCEDERE PRIMA DI ESEGUIRE QUESTO PASSAGGIO.

6. Sul nodo di controllo Ansible:

a. Rimuovere la vecchia chiave SSH con:

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

b. Configurare SSH senza password nel nodo di sostituzione con:

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

7. Eseguire nuovamente il playbook Ansible per configurare il nodo e aggiungerlo al cluster:

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

8. A questo punto, eseguire `pcs status` e verificare che il nodo sostituito sia ora elencato e che i servizi siano in esecuzione.

Espandere o ridurre il cluster

Aggiungere o rimuovere i building block dal cluster.

Panoramica

Questa sezione descrive varie considerazioni e opzioni per regolare le dimensioni del cluster BeeGFS ha. In genere, la dimensione del cluster viene regolata aggiungendo o rimuovendo gli elementi di base, che in genere sono due nodi di file configurati come coppia ha. È inoltre possibile aggiungere o rimuovere singoli nodi di file (o altri tipi di nodi di cluster), se necessario.

Aggiunta di un building block al cluster

Considerazioni

La crescita del cluster mediante l'aggiunta di ulteriori building block è un processo semplice. Prima di iniziare, tenere presenti le restrizioni relative al numero minimo e massimo di nodi del cluster in ciascun cluster ha singolo e determinare se è necessario aggiungere nodi al cluster ha esistente o creare un nuovo cluster ha. In genere, ciascun building block è costituito da due nodi di file, ma tre nodi sono il numero minimo di nodi per cluster (per stabilire il quorum) e dieci sono il numero massimo consigliato (testato). Per gli scenari avanzati è possibile aggiungere un singolo nodo "Tiebreaker" che non esegue alcun servizio BeeGFS durante l'implementazione di un cluster a due nodi. Contatta il supporto NetApp se stai prendendo in considerazione un'implementazione di questo tipo.

Quando si decide come espandere il cluster, tenere presente queste restrizioni e qualsiasi crescita futura prevista del cluster. Ad esempio, se si dispone di un cluster a sei nodi e si desidera aggiungere altri quattro nodi, si consiglia di avviare un nuovo cluster ha.



Tenere presente che un singolo file system BeeGFS può essere costituito da più cluster ha indipendenti. Ciò consente ai file system di continuare a scalare oltre i limiti consigliati/rigidi dei componenti del cluster ha sottostanti.

Fasi

Quando si aggiunge un building block al cluster, sarà necessario creare `host_vars` i file per ogni nuovo nodo di file e nodo dei blocchi (array E-Series). I nomi di questi host devono essere aggiunti all'inventario, insieme alle nuove risorse da creare. I `group_vars` file corrispondenti dovranno essere creati per ogni nuova risorsa. Per "[utilizzo di architetture personalizzate](#)" ulteriori informazioni, vedere la sezione.

Dopo aver creato i file corretti, è sufficiente eseguire nuovamente l'automazione utilizzando il comando:

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

Rimozione di un Building Block dal cluster

È necessario tenere presente una serie di considerazioni quando è necessario dismettere un building block, ad esempio:

- Quali servizi BeeGFS vengono eseguiti in questo building block?
- I nodi di file vengono ritirati e i nodi di blocco devono essere collegati ai nuovi nodi di file?
- Se l'intero building block viene ritirato, i dati devono essere spostati in un nuovo building block, dispersi in nodi esistenti nel cluster o spostati in un nuovo file system BeeGFS o in un altro sistema storage?
- Questo può accadere durante un'interruzione o dovrebbe essere fatto senza interruzioni?
- Il building block è attivamente in uso o contiene principalmente dati che non sono più attivi?

A causa dei diversi possibili punti di partenza e degli stati finali desiderati, contatta il supporto NetApp in modo da poter identificare e implementare la strategia migliore in base al tuo ambiente e ai tuoi requisiti.

Risolvere i problemi

Risoluzione dei problemi di un cluster BeeGFS ha.

Panoramica

Questa sezione illustra come analizzare e risolvere i problemi di vari guasti e altri scenari che potrebbero verificarsi quando si utilizza un cluster BeeGFS ha.

Guide per la risoluzione dei problemi

Analisi di guasti imprevisti

Quando un nodo viene inaspettatamente recintato e i relativi servizi vengono spostati in un altro nodo, il primo passo dovrebbe essere vedere se il cluster indica eventuali guasti alle risorse nella parte inferiore di `pcs status`. In genere, se la schermata è stata completata correttamente e le risorse sono state riavviate su un altro nodo, non sarà presente nulla.

In genere, il passaggio successivo consiste nell'eseguire una ricerca nei log di sistema utilizzando `journalctl` Su uno qualsiasi dei nodi di file rimanenti (i registri di pacemaker sono sincronizzati su tutti i nodi). Se si conosce l'ora in cui si è verificato l'errore, è possibile avviare la ricerca poco prima che si sia verificato l'errore (generalmente si consiglia di almeno dieci minuti prima):

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

Le sezioni seguenti mostrano il testo comune che è possibile inserire nei registri per restringere ulteriormente l'analisi.

Procedure per investigare/risolvere

Fase 1: Controllare se il monitor BeeGFS ha rilevato un guasto:

Se il failover è stato attivato dal monitor BeeGFS, viene visualizzato un errore (in caso contrario, passare alla fase successiva).

```
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 questo caso, il servizio BeeGFS meta_08 si è arrestato per qualche motivo. Per continuare la risoluzione dei problemi si dovrebbe avviare beegfs_02 ed esaminare i log per il servizio a `/var/log/beegfs-meta-meta_08_tgt_0801.log`. Ad esempio, il servizio BeeGFS potrebbe aver riscontrato un errore dell'applicazione a causa di un problema interno o del nodo.



A differenza dei registri di Pacemaker, i registri dei servizi BeeGFS non vengono distribuiti a tutti i nodi del cluster. Per analizzare questi tipi di errori, sono necessari i log del nodo originale in cui si è verificato l'errore.

I possibili problemi che potrebbero essere segnalati dal monitor includono:

- Destinazioni non accessibili.
 - Descrizione: Indica che i volumi a blocchi non erano accessibili.
 - Risoluzione dei problemi:
 - Se anche il servizio non è stato avviato sul nodo di file alternativo, verificare che il nodo di blocco sia integro.
 - Verificare l'eventuale presenza di problemi fisici che impediscano l'accesso ai nodi di blocco da questo nodo di file, ad esempio adattatori o cavi InfiniBand difettosi.
- Rete non raggiungibile.
 - Descrizione: Nessuno degli adattatori utilizzati dai client per connettersi a questo servizio BeeGFS era in linea.
 - Risoluzione dei problemi:
 - In caso di impatto su più/tutti i nodi di file, controllare se si è verificato un errore nella rete utilizzata per collegare i client BeeGFS e il file system.
 - Verificare l'eventuale presenza di problemi fisici che impediscano l'accesso ai client da questo nodo di file, ad esempio cavi o adattatori InfiniBand difettosi.
- Servizio BeeGFS non attivo.

- Descrizione: Un servizio BeeGFS si è arrestato inaspettatamente.
- Risoluzione dei problemi:
 - Nel nodo del file che ha riportato l'errore, controllare i log del servizio BeeGFS interessato per verificare se ha rilevato un blocco. In questo caso, aprire un caso con il supporto NetApp per poter indagare sul crash.
 - Se non vengono segnalati errori nel log di BeeGFS, controllare i log del journal per verificare se systemd ha registrato un motivo per cui il servizio è stato arrestato. In alcuni scenari, il servizio BeeGFS potrebbe non aver avuto la possibilità di registrare alcun messaggio prima che il processo venisse terminato (ad esempio, se qualcuno ha eseguito `kill -9 <PID>`).

Fase 2: Controllare se il nodo ha lasciato inaspettatamente il cluster

Nel caso in cui il nodo abbia subito un guasto hardware catastrofico (ad esempio, la scheda di sistema è morta) o si sia verificato un problema di kernel panic o software simile, il monitor BeeGFS non segnala alcun errore. Cercare invece il nome host e dovrebbero essere visualizzati messaggi da Pacemaker che indicano che il nodo è stato perso inaspettatamente:

```
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
```

Fase 3: Verificare che il pacemaker sia in grado di individuare il nodo

In tutti gli scenari si dovrebbe vedere il tentativo di pacemaker di recinzione del nodo per verificare che sia effettivamente offline (i messaggi esatti possono variare a seconda della causa del recinzione):

```
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
```

Se l'azione di schermo viene completata correttamente, vengono visualizzati messaggi come:

```
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
```

Se l'azione di schermo non è riuscita per qualche motivo, i servizi BeeGFS non potranno essere riavviati su un altro nodo per evitare il rischio di corruzione dei dati. Si tratta di un problema da analizzare separatamente, ad esempio se il dispositivo di schermo (PDU o BMC) non fosse accessibile o non fosse configurato correttamente.

Address Failed Resource Actions (azioni risorsa indirizzo non riuscito) (trovato in fondo allo stato di pcs)

Se una risorsa richiesta per eseguire un servizio BeeGFS non riesce, il monitor BeeGFS attiva un failover. In questo caso, è probabile che nella parte inferiore di non siano elencate "azioni di risorsa non riuscite" pcs status e che sia necessario fare riferimento alla procedura descritta in ["failback dopo un failover non pianificato"](#).

In caso contrario, dovrebbero essere presenti solo due scenari in cui verranno visualizzate le "azioni delle risorse non riuscite".

Procedure per investigare/risolvere

Scenario 1: È stato rilevato un problema temporaneo o permanente con un agente di schermo che è stato riavviato o spostato in un altro nodo.

Alcuni agenti di schermo sono più affidabili di altri e ciascuno implementerà il proprio metodo di monitoraggio per garantire che il dispositivo di schermo sia pronto. In particolare, l'agente Redfish schermo ha rilevato azioni di risorse non riuscite come le seguenti, anche se continuerà a mostrare avviato:

```
* 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
```

Un agente di schermo che segnala azioni di risorse non riuscite su un nodo particolare non dovrebbe attivare un failover dei servizi BeeGFS in esecuzione su quel nodo. Dovrebbe semplicemente essere riavviato automaticamente sullo stesso nodo o su un altro nodo.

Procedura per la risoluzione:

1. Se l'agente di schermo rifiuta costantemente di essere eseguito su tutti i nodi o su un sottoinsieme di nodi, controllare se tali nodi sono in grado di connettersi all'agente di schermo e verificare che l'agente di schermo sia configurato correttamente nell'inventario Ansible.
 - a. Ad esempio, se un agente di schermo Redfish (BMC) è in esecuzione sullo stesso nodo in cui è

responsabile della schermata e la gestione del sistema operativo e gli IP BMC si trovano sulla stessa interfaccia fisica, alcune configurazioni dello switch di rete non consentono la comunicazione tra le due interfacce (per evitare loop di rete). Per impostazione predefinita, il cluster ha tentato di evitare di posizionare gli agenti di schermata sul nodo che sono responsabili della schermata, ma questo può accadere in alcuni scenari/configurazioni.

2. Una volta risolti tutti i problemi (o se il problema sembrava essere effimero), eseguire `pcs resource cleanup` per ripristinare le azioni delle risorse non riuscite.

Scenario 2: Il monitor BeeGFS ha rilevato un problema e ha attivato un failover, ma per qualche motivo le risorse non sono state avviate su un nodo secondario.

A condizione che sia attivata la funzione di schermata e che la risorsa non sia stata bloccata dall'arresto sul nodo originale (vedere la sezione relativa alla risoluzione dei problemi per "standby (on-fail)"), i motivi più probabili includono problemi di avvio della risorsa su un nodo secondario perché:

- Il nodo secondario era già offline.
- Un problema di configurazione fisica o logica ha impedito al secondario di accedere ai volumi di blocco utilizzati come destinazioni BeeGFS.

Procedura per la risoluzione:

1. Per ogni voce nelle azioni delle risorse non riuscite:
 - a. Confermare che l'azione della risorsa non riuscita era un'operazione di avvio.
 - b. In base alla risorsa indicata e al nodo specificato nelle azioni delle risorse non riuscite:
 - i. Cercare e correggere eventuali problemi esterni che impediscano al nodo di avviare la risorsa specificata. Ad esempio, se l'indirizzo IP BeeGFS (floating IP) non si avvia, verificare che almeno una delle interfacce richieste sia connessa/online e cablata allo switch di rete corretto. Se una destinazione BeeGFS (dispositivo a blocchi / volume e-Series) non funziona, verificare che le connessioni fisiche ai nodi di blocco back-end siano collegate come previsto e verificare che i nodi di blocco siano integri.
 - c. Se non ci sono problemi esterni evidenti e si desidera una causa principale per questo incidente, si consiglia di aprire un caso con il supporto NetApp per indagare prima di procedere, in quanto i seguenti passaggi potrebbero rendere difficile/impossibile l'analisi della causa principale (RCA).
2. Dopo aver risolto eventuali problemi esterni:
 - a. Commentare eventuali nodi non funzionali dal file Ansible `inventory.yml` ed eseguire nuovamente il playbook Ansible completo per assicurarsi che tutte le configurazioni logiche siano configurate correttamente sui nodi secondari.
 - i. Nota: Non dimenticare di rimuovere il commento da questi nodi e di eseguire nuovamente il playbook una volta che i nodi sono in buono stato e sei pronto per il failback.
 - b. In alternativa, è possibile tentare di ripristinare manualmente il cluster:
 - i. Posizionare di nuovo online i nodi offline utilizzando: `pcs cluster start <HOSTNAME>`
 - ii. Cancellare tutte le azioni delle risorse non riuscite utilizzando: `pcs resource cleanup`
 - iii. Eseguire lo stato dei PC e verificare che tutti i servizi iniziano come previsto.
 - iv. Se necessario, eseguire `pcs resource relocate run` per spostare nuovamente le risorse nel nodo preferito (se disponibile).

Problemi comuni

I servizi BeeGFS non eseguono il failover o il failback quando richiesto

Probabile problema: il `pcs resource relocate` il comando run è stato eseguito, ma non è mai stato completato correttamente.

Come controllare: Esegui `pcs constraint --full` E verificare la presenza di eventuali vincoli di posizione con un ID di `pcs-relocate-<RESOURCE>`.

Come risolvere: Esegui `pcs resource relocate clear` quindi rieseguire `pcs constraint --full` per verificare che i vincoli aggiuntivi vengano rimossi.

Un nodo nello stato di PC mostra "standby (on-fail)" quando la scherma è disattivata

Probabile problema: pacemaker non è riuscito a confermare che tutte le risorse sono state interrotte sul nodo che ha avuto esito negativo.

Come risolvere:

1. Esegui `pcs status` e verificare la presenza di risorse che non sono "avviate" o che mostrano errori nella parte inferiore dell'output e risolvere eventuali problemi.
2. Per riportare il nodo in linea eseguire `pcs resource cleanup --node=<HOSTNAME>`.

Dopo un failover imprevisto, le risorse mostrano "Started (on-fail)" (avviato (on-fail)) in stato PC quando la scherma è attivata

Probabile problema: si è verificato Un problema che ha attivato un failover, ma Pacemaker non è riuscito a verificare che il nodo sia stato recintato. Questo potrebbe verificarsi a causa di una configurazione errata del recinto o di un problema con l'agente di recinzione (ad esempio: La PDU è stata disconnessa dalla rete).

Come risolvere:

1. Verificare che il nodo sia effettivamente spento.



Se il nodo specificato non è effettivamente disattivato, ma esegue risorse o servizi cluster, si VERIFICHERÀ un danneggiamento dei dati o un errore del cluster.

2. Confermare manualmente la scherma con: `pcs stonith confirm <NODE>`

A questo punto i servizi dovrebbero terminare il failover e essere riavviati su un altro nodo integro.

Attività comuni di risoluzione dei problemi

Riavviare i singoli servizi BeeGFS

In genere, se un servizio BeeGFS deve essere riavviato (ad esempio per facilitare una modifica della configurazione), questa operazione deve essere eseguita aggiornando l'inventario Ansible e rieseguendo il manuale. In alcuni scenari potrebbe essere consigliabile riavviare singoli servizi per facilitare la risoluzione dei problemi più rapida, ad esempio per modificare il livello di registrazione senza dover attendere l'esecuzione dell'intero playbook.



A meno che non vengano aggiunte modifiche manuali all'inventario Ansible, queste verranno ripristinate alla prossima esecuzione del playbook Ansible.

Opzione 1: Riavvio controllato dal sistema

Se esiste il rischio che il servizio BeeGFS non si riavvii correttamente con la nuova configurazione, impostare innanzitutto il cluster in modalità di manutenzione per evitare che il monitor BeeGFS rilevi che il servizio è stato arrestato e che venga attivato un failover indesiderato:

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

Se necessario, apportare eventuali modifiche alla configurazione dei servizi all'indirizzo

/mnt/<SERVICE_ID>/_config/beegfs-.conf (esempio:

/mnt/meta_01_tgt_0101/metadata_config/beegfs-meta.conf) quindi utilizzare systemd per riavviarlo:

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

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

Opzione 2: Riavvio controllato da pacemaker

Se non si è preoccupati per la nuova configurazione, il servizio potrebbe arrestarsi in modo imprevisto (ad esempio, semplicemente cambiando il livello di registrazione) oppure ci si trova in una finestra di manutenzione e non si è preoccupati per i tempi di inattività, è sufficiente riavviare il monitor BeeGFS per il servizio che si desidera riavviare:

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

Ad esempio, per riavviare il servizio di gestione BeeGFS: `pcs resource restart mgmt-monitor`

Informazioni sul copyright

Copyright © 2026 NetApp, Inc. Tutti i diritti riservati. Stampato negli Stati Uniti d'America. Nessuna porzione di questo documento soggetta a copyright può essere riprodotta in qualsiasi formato o mezzo (grafico, elettronico o meccanico, inclusi fotocopie, registrazione, nastri o storage in un sistema elettronico) senza previo consenso scritto da parte del detentore del copyright.

Il software derivato dal materiale sottoposto a copyright di NetApp è soggetto alla seguente licenza e dichiarazione di non responsabilità:

IL PRESENTE SOFTWARE VIENE FORNITO DA NETAPP "COSÌ COM'È" E SENZA QUALSIVOGLIA TIPO DI GARANZIA IMPLICITA O ESPRESSA FRA CUI, A TITOLO ESEMPLIFICATIVO E NON ESAUSTIVO, GARANZIE IMPLICITE DI COMMERCIALIZZABILITÀ E IDONEITÀ PER UNO SCOPO SPECIFICO, CHE VENGONO DECLINATE DAL PRESENTE DOCUMENTO. NETAPP NON VERRÀ CONSIDERATA RESPONSABILE IN ALCUN CASO PER QUALSIVOGLIA DANNO DIRETTO, INDIRETTO, ACCIDENTALE, SPECIALE, ESEMPLARE E CONSEGUENZIALE (COMPRESI, A TITOLO ESEMPLIFICATIVO E NON ESAUSTIVO, PROCUREMENT O SOSTITUZIONE DI MERCI O SERVIZI, IMPOSSIBILITÀ DI UTILIZZO O PERDITA DI DATI O PROFITTI OPPURE INTERRUZIONE DELL'ATTIVITÀ AZIENDALE) CAUSATO IN QUALSIVOGLIA MODO O IN RELAZIONE A QUALUNQUE TEORIA DI RESPONSABILITÀ, SIA ESSA CONTRATTUALE, RIGOROSA O DOVUTA A INSOLVENZA (COMPRESA LA NEGLIGENZA O ALTRO) INSORTA IN QUALSIASI MODO ATTRAVERSO L'UTILIZZO DEL PRESENTE SOFTWARE ANCHE IN PRESENZA DI UN PREAVVISO CIRCA L'EVENTUALITÀ DI QUESTO TIPO DI DANNI.

NetApp si riserva il diritto di modificare in qualsiasi momento qualunque prodotto descritto nel presente documento senza fornire alcun preavviso. NetApp non si assume alcuna responsabilità circa l'utilizzo dei prodotti o materiali descritti nel presente documento, con l'eccezione di quanto concordato espressamente e per iscritto da NetApp. L'utilizzo o l'acquisto del presente prodotto non comporta il rilascio di una licenza nell'ambito di un qualche diritto di brevetto, marchio commerciale o altro diritto di proprietà intellettuale di NetApp.

Il prodotto descritto in questa guida può essere protetto da uno o più brevetti degli Stati Uniti, esteri o in attesa di approvazione.

LEGENDA PER I DIRITTI SOTTOPOSTI A LIMITAZIONE: l'utilizzo, la duplicazione o la divulgazione da parte degli enti governativi sono soggetti alle limitazioni indicate nel sottoparagrafo (b)(3) della clausola Rights in Technical Data and Computer Software del DFARS 252.227-7013 (FEB 2014) e FAR 52.227-19 (DIC 2007).

I dati contenuti nel presente documento riguardano un articolo commerciale (secondo la definizione data in FAR 2.101) e sono di proprietà di NetApp, Inc. Tutti i dati tecnici e il software NetApp forniti secondo i termini del presente Contratto sono articoli aventi natura commerciale, sviluppati con finanziamenti esclusivamente privati. Il governo statunitense ha una licenza irrevocabile limitata, non esclusiva, non trasferibile, non cedibile, mondiale, per l'utilizzo dei Dati esclusivamente in connessione con e a supporto di un contratto governativo statunitense in base al quale i Dati sono distribuiti. Con la sola esclusione di quanto indicato nel presente documento, i Dati non possono essere utilizzati, divulgati, riprodotti, modificati, visualizzati o mostrati senza la previa approvazione scritta di NetApp, Inc. I diritti di licenza del governo degli Stati Uniti per il Dipartimento della Difesa sono limitati ai diritti identificati nella clausola DFARS 252.227-7015(b) (FEB 2014).

Informazioni sul marchio commerciale

NETAPP, il logo NETAPP e i marchi elencati alla pagina <http://www.netapp.com/TM> sono marchi di NetApp, Inc. Gli altri nomi di aziende e prodotti potrebbero essere marchi dei rispettivi proprietari.