Administrar clusters BeeGFS

BeeGFS on NetApp with E-Series Storage

NetApp
January 27, 2026

Índice

Administrar clusters BeeGFS

Visão geral, conceitos-chave e terminologia

Saiba como administrar clusters de HA do BeeGFS após a implantação.

Visão geral

Esta seção destina-se aos administradores de cluster que precisam gerenciar clusters de HA do BeeGFS após a implantação. Mesmo aqueles que estão familiarizados com os clusters de HA do Linux devem ler cuidadosamente este guia, pois há várias diferenças em como gerenciar o cluster, especialmente em relação à reconfiguração devido ao uso do Ansible.

Conceitos-chave

Embora alguns desses conceitos sejam introduzidos na página principal"termos e conceitos", é útil reintroduzi-los no contexto de um cluster BeeGFS HA:

Nó de cluster: Um servidor executando serviços de pacemaker e Corosync e participando do cluster HA.

Nó de arquivo: Um nó de cluster usado para executar um ou mais serviços de gerenciamento, metadados ou storage do BeeGFS.

Nó de bloco: Um sistema de storage do NetApp e-Series que fornece storage de bloco para nós de arquivos. Esses nós não participam do cluster BeeGFS HA, pois fornecem suas próprias funcionalidades de HA autônomas. Cada nó consiste em duas controladoras de storage que fornecem alta disponibilidade na camada de bloco.

Serviço BeeGFS: Um serviço de gerenciamento, metadados ou storage do BeeGFS. Cada nó de arquivo executará um ou mais serviços que usarão volumes no nó de bloco para armazenar seus dados.

Bloco básico: Uma implantação padronizada de nós de arquivo BeeGFS, nós de bloco de e-Series e serviços BeeGFS executados neles, o que simplifica o dimensionamento de um sistema de arquivos/cluster BeeGFS HA seguindo uma arquitetura verificada pelo NetApp. Clusters de HA personalizados também são compatíveis, mas geralmente seguem uma abordagem de componentes básicos semelhante para simplificar o dimensionamento.

Cluster do BeeGFS HA: Um número dimensionável de nós de arquivo usados para executar serviços do BeeGFS com o respaldo de nós de bloco para armazenar dados do BeeGFS de maneira altamente disponível. Desenvolvido com base em componentes de código aberto comprovados pela indústria, a Pacemaker e o Corosync, usando o Ansible para embalagem e implantação.

Serviços de cluster: refere-se aos serviços de pacemaker e Corosync executados em cada nó que participa do cluster. Observação é possível que um nó não execute nenhum serviço BeeGFS e apenas participe do cluster como nó "tiebreaker", caso haja apenas a necessidade de dois nós de arquivo.

Recursos de cluster: para cada serviço BeeGFS executado no cluster, você verá um recurso de monitoramento BeeGFS e um grupo de recursos contendo recursos para destino(s) BeeGFS, endereço(s) IP (IPs flutuantes) e o próprio serviço BeeGFS.

Ansible: Uma ferramenta para provisionamento de software, gerenciamento de configurações e implantação de aplicações, habilitando a infraestrutura como código. É como os clusters do BeeGFS são empacotados para simplificar o processo de implantação, reconfiguração e atualização do BeeGFS no NetApp.

PCs: Uma interface de linha de comando disponível a partir de qualquer um dos nós de arquivo no cluster usado para consultar e controlar o estado de nós e recursos no cluster.

Terminologia comum

Failover: cada serviço BeeGFS tem um nó de arquivo preferido em que será executado, a menos que esse nó falhe. Quando um serviço BeeGFS está sendo executado no nó de arquivo secundário/não preferencial, diz-se que está em failover.

Failback: o ato de mover os serviços BeeGFS de um nó de arquivo não preferencial de volta para o nó preferido.

Par de HA: dois nós de arquivos que podem acessar o mesmo conjunto de nós de bloco às vezes são referidos como um par de HA. Este é um termo comum usado em todo o NetApp para se referir a dois controladores de storage ou nós que podem "assumir o controle" uns dos outros.

Modo de manutenção: desativa todo o monitoramento de recursos e impede que o pacemaker mova ou gerencie recursos no cluster (consulte também a seção em "modo de manutenção").

Cluster de HA: um ou mais nós de arquivos executando serviços BeeGFS que podem fazer o failover entre vários nós no cluster para criar um sistema de arquivos BeeGFS altamente disponível. Normalmente, os nós de arquivos são configurados em pares de HA que podem executar um subconjunto dos serviços BeeGFS no cluster.

Quando usar o Ansible versus a ferramenta PCs

Quando você deve usar o Ansible versus a ferramenta de linha de comando PCs para gerenciar o cluster de HA?

Todas as tarefas de implantação e reconfiguração de cluster devem ser concluídas usando o Ansible a partir de um nó de controle externo do Ansible. Alterações temporárias no estado do cluster (por exemplo, colocar nós dentro e fora de espera) normalmente serão realizadas fazendo login em um nó do cluster (de preferência um que não esteja degradado ou prestes a ser submetido a manutenção) e usando a ferramenta de linha de comando PCs.

Usar o Ansible, modificar qualquer configuração do cluster, incluindo recursos, restrições, propriedades e serviços do BeeGFS. Manter uma cópia atualizada do inventário e do manual de estratégia do Ansible (idealmente no controle de origem para controlar alterações) faz parte da manutenção do cluster. Quando você precisar fazer alterações na configuração, atualize o inventário e execute novamente o manual de estratégia do Ansible que importa a função de HA do BeeGFS.

A função de HA tratará da colocação do cluster no modo de manutenção e depois fará as alterações necessárias antes de reiniciar os serviços do BeeGFS ou do cluster para aplicar a nova configuração. Como as reinicializações completas de nós geralmente não são necessárias fora da implantação inicial, a reinstalação do Ansible geralmente é considerada um procedimento "seguro", mas é sempre recomendada durante períodos de manutenção ou horas extras, caso os serviços do BeeGFS precisem ser reiniciados. Essas reinicializações geralmente não devem causar erros de aplicativo, mas podem prejudicar o desempenho (o que alguns aplicativos podem lidar melhor que outros).

A reexecução do Ansible também é uma opção quando você deseja retornar todo o cluster a um estado totalmente ideal e pode, em alguns casos, ser capaz de recuperar o estado do cluster com mais facilidade do que o uso de PCs. Especialmente durante uma emergência em que o cluster está inativo por algum motivo, uma vez que todos os nós estão voltando a executar o Ansible pode recuperar o cluster de forma mais rápida e confiável do que tentar usar PCs.

Examine o estado do cluster

Use PCs para visualizar o estado do cluster.

Visão geral

Executar `pcs status` a partir de qualquer um dos nós de cluster é a maneira mais fácil de ver o estado geral do cluster e o status de cada recurso (como os serviços BeeGFS e suas dependências). Esta seção percorre o que você encontrará na saída do `pcs status` comando.

Compreender a saída de `pcs status`

Execute `pcs status` em qualquer nó de cluster onde os serviços de cluster (Pacemaker e Corosync) são iniciados. A parte superior da saída irá mostrar-lhe um resumo do 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
```

A seção abaixo lista os nós no cluster:

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

Isso indica notavelmente todos os nós que estão em standby ou offline. Os nós em modo de espera ainda estão participando do cluster, mas marcados como não qualificados para executar recursos. Os nós que estão offline indicam que os serviços de cluster não estão sendo executados nesse nó, seja devido a ser parado manualmente ou porque o nó foi reinicializado/encerrado.

> Quando os nós são iniciados pela primeira vez, os serviços de cluster serão interrompidos e precisam ser iniciados manualmente para evitar falhas acidentais de recursos para um nó que não seja saudável.

Se os nós estiverem em standby ou offline devido a um motivo não administrativo (por exemplo, uma falha), o texto adicional será exibido ao lado do estado do nó entre parênteses. Por exemplo, se o esgrima estiver desativado e um recurso encontrar uma falha, você verá `Node <HOSTNAME>: standby (on-fail)`. Outro

estado possível é `Node <HOSTNAME>: UNCLEAN (offline)`, que será visto brevemente como um nó está sendo cercado, mas persistirá se o fencing falhar, indicando que o cluster não pode confirmar o estado do nó (isso pode impedir que os recursos comecem em outros nós).

A próxima seção mostra uma lista de todos os recursos no cluster e seus estados:

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

Semelhante aos nós, o texto adicional será exibido ao lado do estado do recurso entre parênteses se houver algum problema com o recurso. Por exemplo, se o pacemaker solicitar uma parada de recurso e ele não for concluído dentro do tempo alocado, o pacemaker tentará cercar o nó. Se o esgrima estiver desativado ou a operação de esgrima falhar, o estado do recurso será `FAILED <HOSTNAME> (blocked)` e o pacemaker não poderá iniciá-lo em um nó diferente.

Vale a pena notar que os clusters de HA do BeeGFS utilizam vários agentes de recursos OCF personalizados otimizados pelo BeeGFS. Em particular, o monitor BeeGFS é responsável por acionar um failover quando os recursos do BeeGFS em um nó específico não estiverem disponíveis.

Reconfigure o cluster de HA e o BeeGFS

Use o Ansible para reconfigurar o cluster.

Visão geral

De um modo geral, a reconfiguração de qualquer aspecto do cluster BeeGFS HA deve ser feita atualizando seu inventário do Ansible e executando novamente `ansible-playbook` o comando. Isso inclui atualização de alertas, alteração da configuração de cercas permanentes ou ajuste da configuração do serviço BeeGFS. Estes são ajustados usando o `group_vars/ha_cluster.yml` arquivo e uma lista completa de opções pode ser encontrada "Especifique a Configuração do nó de ficheiro Comum"na seção.

Consulte abaixo para obter detalhes adicionais sobre opções de configuração selecionadas que os administradores devem estar cientes ao executar a manutenção ou a manutenção do cluster.

Como desativar e ativar o Esgrima

O esgrima é ativado/exigido por padrão ao configurar o cluster. Em alguns casos, pode ser desejável desativar temporariamente o esgrima para garantir que os nós não sejam acidentalmente desligados ao executar determinadas operações de manutenção (como atualizar o sistema operacional). Embora isso possa ser desativado manualmente, há compensações que os administradores devem estar cientes.

Opção 1: Desativar cercas usando o Ansible (recomendado).

Quando a vedação é desativada usando o Ansible, a ação on-fail do monitor BeeGFS é alterada de "cerca"

para "espera". Isso significa que, se o monitor BeeGFS detetar uma falha, ele tentará colocar o nó em espera e fazer o failover de todos os serviços BeeGFS. Fora da solução de problemas/testes ativos, isso geralmente é mais desejável do que a opção 2. A desvantagem é que se um recurso não parar no nó original, ele será bloqueado de começar em outro lugar (é por isso que o esgrima é normalmente necessário para clusters de produção).

1. No inventário do Ansible, `groups_vars/ha_cluster.yml` adicione a seguinte configuração:

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

2. Execute novamente o manual de estratégia do Ansible para aplicar as alterações ao cluster.

Opção 2: Desativar a vedação manualmente.

Em alguns casos, você pode desativar temporariamente o esgrima sem executar novamente o Ansible, talvez para facilitar a solução de problemas ou o teste do cluster.

> Nessa configuração, se o monitor BeeGFS detetar uma falha, o cluster tentará interromper o grupo de recursos correspondente. Ele NÃO acionará um failover completo ou tentará reiniciar ou mover o grupo de recursos afetado para outro host. Para recuperar, solucione quaisquer problemas em seguida, execute `pcs resource cleanup` ou coloque manualmente o nó em espera.

Passos:

1. Para determinar se a vedação (stonith) está globalmente ativada ou desativada, execute: `pcs property show stonith-enabled`
2. Para desativar a execução de esgrima: `pcs property set stonith-enabled=false`
3. Para ativar a execução de esgrima: `pcs property set stonith-enabled=true`

> Esta configuração será substituída na próxima vez que você executar o manual do Ansible.

Atualizar os componentes do cluster HA

Atualizar os serviços BeeGFS

Use o Ansible para atualizar a versão do BeeGFS em execução no seu cluster de alta disponibilidade.

Visão geral

BeeGFS segue um `major.minor.patch` esquema de controle de versão. As funções do BeeGFS HA Ansible são fornecidas para cada versão com suporte `major.minor` (por exemplo, `beegfs_ha_7_2` e `beegfs_ha_7_3`). Cada função de HA é fixada à versão de patch BeeGFS mais recente disponível no momento do lançamento da coleção Ansible.

O Ansible deve ser usado para todas as atualizações do BeeGFS, incluindo a migração entre versões principais, secundárias e de correção do BeeGFS. Para atualizar o BeeGFS, você precisará primeiro atualizar

a coleção Ansible do BeeGFS, o que também trará as correções e melhorias mais recentes para a automação de implantação/gerenciamento e o cluster de alta disponibilidade subjacente. Mesmo após atualizar para a versão mais recente da coleção, o BeeGFS não será atualizado até que `ansible-playbook` seja executado com o `-e "beegfs_ha_force_upgrade=true"` definido. Para detalhes adicionais sobre cada atualização, consulte a "Documentação do BeeGFS Upgrade" da sua versão atual.

> Se você estiver atualizando para o BeeGFS v8, consulte o "Atualize para BeeGFS v8" procedimento em vez disso.

Caminhos de atualização testados

Os seguintes caminhos de upgrade foram testados e verificados:

Versão original	Versão de atualização	Multirail	Detalhes
7.2.6	7.3.2	Sim	Atualizando a coleção beegfs de v3,0.1 para v3,1.0, multirail adicionado
7.2.6	7.2.8	Não	Atualizando a coleção beegfs de v3,0.1 para v3,1.0
7.2.8	7.3.1	Sim	Atualização usando beegfs coleção v3,1.0, multi-rail adicionado
7.3.1	7.3.2	Sim	Atualize usando a coleção beegfs v3,1.0
7.3.2	7.4.1	Sim	Atualize usando a coleção beegfs v3,2.0
7.4.1	7.4.2	Sim	Atualize usando a coleção beegfs v3,2.0
7.4.2	7.4.6	Sim	Atualize usando a coleção beegfs v3,2.0
7.4.6	8,0	Sim	Atualize usando as instruções no "Atualize para BeeGFS v8" procedimento.
7.4.6	8,1	Sim	Atualize usando as instruções no "Atualize para BeeGFS v8" procedimento.
7.4.6	8,2	Sim	Atualize usando as instruções no "Atualize para BeeGFS v8" procedimento.

Etapas de atualização do BeeGFS

As seções a seguir fornecem etapas para atualizar a coleção BeeGFS Ansible e o próprio BeeGFS. Preste atenção especial a qualquer passo extra para atualizar as versões BeeGFS Major ou menor.

Passo 1: Atualize a coleção BeeGFS

Para atualizações de coleção com acesso ao "Ansible Galaxy", execute o seguinte comando:

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

Para atualizações de coleção offline, faça o download da coleção "Ansible Galaxy"clicando no desejado `Install Version` e,em seguida `Download tarball`, . Transfira o tarball para o nó de controle do Ansible e execute o seguinte comando.

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

Consulte "Instalando coleções" para obter mais informações.

Etapa 2: Atualize o inventário do Ansible

Faça as atualizações necessárias ou desejadas nos arquivos de inventário do Ansible do seu cluster. Consulte a seção Notas de atualização da versão abaixo para detalhes sobre os requisitos específicos da sua atualização. Consulte a seção "Visão geral do Ansible Inventory" para informações gerais sobre como configurar o inventário BeeGFS HA.

Etapa 3: Atualizar o manual do Ansible (somente ao atualizar versões principais ou secundárias)

Se você estiver se movendo entre versões maiores ou menores, no `playbook.yml` arquivo usado para implantar e manter o cluster, atualize o nome da `beegfs_ha_<VERSION>` função para refletir a versão desejada. Por exemplo, se você quiser implantar o BeeGFS 7,4, isso `beegfs_ha_7_4` seria :

```
- 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
```

Para obter mais detalhes sobre o conteúdo deste arquivo de manual de estratégia, consulte "Implante o cluster BeeGFS HA"a seção.

Passo 4: Execute a atualização BeeGFS

Para aplicar a atualização BeeGFS:

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

Nos bastidores, o papel BeeGFS HA vai lidar com:

- Verifique se o cluster está no estado ideal com cada serviço BeeGFS localizado no nó preferido.
- Coloque o cluster no modo de manutenção.
- Atualize os componentes do cluster HA (se necessário).
- Atualize cada nó de arquivo, um de cada vez, da seguinte forma:

    - Coloque-a em standby e faça failover de seus serviços para o nó secundário.
    - Atualize os pacotes BeeGFS.
    - Serviços de retorno.
- Mova o cluster para fora do modo de manutenção.

Notas de atualização da versão

Atualização do BeeGFS versão 7.2.6 ou 7.3.0

Alterações na autenticação baseada em conexão

BeeGFS versão 7.3.2 e posteriores exigem que a autenticação baseada em conexão seja configurada. Os serviços não serão iniciados sem:

- Especificando um `connAuthFile`, ou
- Configuração `connDisableAuthentication=true` no arquivo de configuração do serviço.

É altamente recomendável habilitar autenticação baseada em conexão para segurança. Veja "Autenticação baseada em conexão BeeGFS" para mais informações.

As `beegfs_ha*`funções geram e distribuem automaticamente o arquivo de autenticação para:

- Todos os nós de arquivo no cluster
- O nó de controle Ansible em `<playbook_directory>/files/beegfs/<beegfs_mgmt_ip_address>_connAuthFile`

A `beegfs_client`função detectará e aplicará automaticamente este arquivo aos clientes quando estiver presente.

> Se você não utilizou o `beegfs_client` papel para configurar os clientes, será necessário distribuir manualmente o arquivo de autenticação para cada cliente e configurar a configuração `connAuthFile` no arquivo `beegfs-client.conf`. Ao atualizar de uma versão do BeeGFS sem autenticação baseada em conexão, os clientes perderão o acesso, a menos que você desative a autenticação baseada em conexão durante a atualização, configurando `beegfs_ha_conn_auth_enabled: false` em `group_vars/ha_cluster.yml` (não recomendado).

Para obter detalhes adicionais e opções de configuração alternativas, consulte a etapa de configuração de autenticação de conexão na seção "Especifique a Configuração do nó de ficheiro Comum".

Atualize para BeeGFS v8

Siga estes passos para atualizar seu cluster BeeGFS HA da versão 7.4.6 para BeeGFS v8.

Visão geral

O BeeGFS v8 introduz diversas mudanças significativas que exigem configuração adicional antes da atualização do BeeGFS v7. Este documento orienta você na preparação do seu cluster para os novos requisitos do BeeGFS v8 e, em seguida, na atualização para o BeeGFS v8.

Antes de atualizar para o BeeGFS v8, certifique-se de que seu sistema esteja executando pelo menos o BeeGFS 7.4.6. Qualquer cluster que execute uma versão anterior ao BeeGFS 7.4.6 deve primeiro "Atualize para a versão 7.4.6" antes de prosseguir com este procedimento de atualização para o BeeGFS v8.

Principais alterações no BeeGFS v8

BeeGFS v8 introduz as seguintes alterações principais:

- **Aplicação de licença:** O BeeGFS v8 requer uma licença para usar recursos premium, como pools de armazenamento, destinos de armazenamento remoto, BeeOND e mais. Adquira uma licença válida para o seu cluster BeeGFS antes de atualizar. Se necessário, você pode obter uma licença de avaliação temporária do BeeGFS v8 em "Portal de Licenças BeeGFS".
- **Migração de banco de dados do serviço de gerenciamento:** Para habilitar a configuração com o novo formato baseado em TOML no BeeGFS v8, você deve migrar manualmente o banco de dados do serviço de gerenciamento do BeeGFS v7 para o formato atualizado do BeeGFS v8.
- **Criptografia TLS:** O BeeGFS v8 introduz TLS para comunicação segura entre serviços. Você precisará gerar e distribuir certificados TLS para o serviço de gerenciamento do BeeGFS e o `beegfs` utilitário de linha de comando como parte da atualização.

Para mais detalhes e alterações adicionais no BeeGFS 8, consulte o "Guia de atualização do BeeGFS v8.0.0".

A atualização para BeeGFS v8 requer a indisponibilidade do cluster. Além disso, clientes BeeGFS v7 não podem se conectar a clusters BeeGFS v8. Coordene cuidadosamente o cronograma de atualização entre o cluster e os clientes para minimizar o impacto nas operações.

Prepare seu cluster BeeGFS para a atualização

Antes de iniciar a atualização, prepare cuidadosamente seu ambiente para garantir uma transição tranquila e minimizar o tempo de inatividade.

1. Certifique-se de que seu cluster esteja em bom estado, com todos os serviços BeeGFS em execução nos nós preferenciais. A partir de um nó de arquivo executando os serviços BeeGFS, verifique se todos os recursos Pacemaker estão em execução nos nós preferenciais:

   ```
   pcs status
   ```

2. Registre e faça backup da configuração do seu cluster.
   a. Consulte o "Documentação de backup do BeeGFS" para obter instruções sobre como fazer backup da configuração do seu cluster.
   b. Faça backup do diretório de dados de gerenciamento existente:

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

   c. Execute os seguintes comandos a partir de um cliente beegfs e salve a saída para referência:

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

d. Se estiver usando espelhamento, colete informações detalhadas sobre o estado:

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

3. Prepare seus clientes para períodos de inatividade e pare os `beegfs-client` serviços. Para cada cliente, execute:

```
systemctl stop beegfs-client
```

4. Para cada cluster Pacemaker, desative STONITH. Isso permitirá validar a integridade do cluster após a atualização sem acionar reinicializações desnecessárias dos nós.

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

5. Para todos os clusters Pacemaker no namespace BeeGFS, use PCS para parar o cluster:

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

Atualize os pacotes BeeGFS

Em todos os nós de arquivos do cluster, adicione o repositório do pacote BeeGFS v8 para sua distribuição Linux. As instruções para usar os repositórios oficiais do BeeGFS podem ser encontradas em "Página de download do BeeGFS". Caso contrário, configure seu repositório espelho local do beegfs de acordo.

Os passos a seguir descrevem o uso do repositório oficial BeeGFS 8.2 em nós de arquivo RHEL 9. Execute os seguintes passos em todos os nós de arquivo do cluster:

1. Importe a chave GPG do BeeGFS:

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

2. Importe o repositório BeeGFS:

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

Remova quaisquer repositórios BeeGFS previamente configurados para evitar conflitos com o novo repositório BeeGFS v8.

3. Limpe o cache do seu gerenciador de pacotes:

```
dnf clean all
```

4. Em todos os nós de arquivo, atualize os pacotes BeeGFS para BeeGFS 8.2.

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

Em um cluster padrão, o `beegfs-mgmtd` pacote será atualizado apenas nos dois primeiros nós de arquivo.

Atualize o banco de dados de gerenciamento

Em um dos nós de arquivos que executam o serviço de gerenciamento BeeGFS, realize as seguintes etapas para migrar o banco de dados de gerenciamento do BeeGFS v7 para o v8.

1. Liste todos os dispositivos NVMe e filtre pelo destino de gerenciamento:

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

   a. Observe o caminho do dispositivo a partir da saída.

   b. Monte o dispositivo de destino de gerenciamento no ponto de montagem de destino de gerenciamento existente (substitua `/dev/nvmeXnY` pelo caminho do seu dispositivo):

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

2. Importe seus dados de gerenciamento do BeeGFS 7 para o novo formato de banco de dados executando:

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

   Saída esperada:

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

A importação automática pode não ser bem-sucedida em todos os casos devido aos requisitos de validação mais rigorosos no BeeGFS v8. Por exemplo, se os destinos forem atribuídos a pools de armazenamento inexistentes, a importação falhará. Se a migração de banco de dados falhar, não prossiga com a atualização. Entre em contato com o suporte da NetApp para obter assistência na resolução dos problemas de migração de banco de dados. Como solução provisória, você pode fazer o downgrade dos pacotes do BeeGFS v8 e continuar executando o BeeGFS v7 enquanto o problema é resolvido.

3. Mova o arquivo SQLite gerado para o ponto de montagem do serviço de gerenciamento:

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

4. Mova o arquivo gerado `beegfs-mgmtd.toml` para o ponto de montagem do serviço de gerenciamento:

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

Preparar o `beegfs-mgmtd.toml` arquivo de configuração será feito após a conclusão das etapas de licenciamento e configuração TLS nas próximas seções.

Configurar licenciamento

1. Instale os pacotes de licença do beegfs em todos os nós que executam o serviço de gerenciamento do beegfs. Isso normalmente corresponde aos dois primeiros nós do cluster:

```
dnf install libbeegfs-license
```

2. Faça o download do seu arquivo de licença BeeGFS v8 para os nós de gerenciamento e coloque-o em:

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

Configurar criptografia TLS

O BeeGFS v8 requer criptografia TLS para comunicação segura entre os serviços de gerenciamento e os clientes. Existem três opções para configurar a criptografia TLS nas comunicações de rede entre os serviços de gerenciamento e os serviços de cliente. O método recomendado e mais seguro é usar certificados assinados por uma Autoridade Certificadora confiável. Como alternativa, você pode criar sua própria CA local para assinar certificados para o seu cluster BeeGFS. Para ambientes onde a criptografia não é necessária ou para solução de problemas, o TLS pode ser desativado completamente, embora isso seja desaconselhado, pois expõe informações confidenciais à rede.

Antes de prosseguir, siga as instruções no "Configurar criptografia TLS para BeeGFS 8" guia para configurar a criptografia TLS em seu ambiente.

Atualizar a configuração do serviço de gerenciamento

Prepare o arquivo de configuração do serviço de gerenciamento BeeGFS v8 transferindo manualmente as configurações do seu arquivo de configuração BeeGFS v7 para o arquivo `/mnt/mgmt_tgt_mgmt01/mgmt_config/beegfs-mgmtd.toml`.

1. No nó de gerenciamento com o alvo de gerenciamento montado, faça referência ao `/mnt/mgmt_tgt_mgmt01/mgmt_config/beegfs-mgmtd.conf` arquivo de serviço de gerenciamento do BeeGFS 7 e transfira todas as configurações para o `/mnt/mgmt_tgt_mgmt01/mgmt_config/beegfs-mgmtd.toml` arquivo. Para uma configuração básica, seu `beegfs-mgmtd.toml` pode ser semelhante ao seguinte:

   ```
   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']
   ```

   Ajuste todos os caminhos conforme necessário para corresponder ao seu ambiente e à configuração TLS.

2. Em cada nó de arquivo que executa serviços de gerenciamento, modifique o arquivo de serviço do systemd para apontar para o novo local do arquivo de configuração.

   ```
   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. Recarregar systemd:

      ```
      systemctl daemon-reload
      ```

3. Para cada nó de arquivo que executa serviços de gerenciamento, abra a porta 8010 para a comunicação gRPC do serviço de gerenciamento.

   a. Adicione a porta 8010/tcp à zona beegfs:

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

b. Recarregue o firewall para aplicar a alteração:

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

Atualize o script de monitor do BeeGFS

O script OCF do Pacemaker `beegfs-monitor` precisa ser atualizado para oferecer suporte ao novo formato de configuração TOML e ao gerenciamento de serviços do systemd. Atualize o script em um nó do cluster, depois copie o script atualizado para todos os outros nós.

1. Crie um backup do script atual:

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

2. Atualize o caminho do arquivo de configuração de gerenciamento de `.conf` para `.toml`:

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

Alternativamente, localize manualmente o seguinte bloco no script:

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

E substitua por:

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

3. Atualize as `get_interfaces()` e `get_subnet_ips()` funções para suportar a configuração TOML:

a. Abra o script em um editor de texto:

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

b. Localize as duas funções: `get_interfaces()` e `get_subnet_ips()`.

c. Exclua ambas as funções inteiras, começando em `get_interfaces()` até o final de `get_subnet_ips()`.

d. Copie e cole as seguintes funções atualizadas em seus respectivos lugares:

```
# 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. Salve e saia do editor de texto.

f. Execute o seguinte comando para verificar se há erros de sintaxe no script antes de prosseguir. A ausência de saída indica que o script está sintaticamente correto.

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

4. Copie o script OCF atualizado `beegfs-monitor` para todos os outros nós do cluster para garantir a consistência:

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

Traga o cluster de volta à ativa

1. Após concluir todas as etapas de upgrade anteriores, coloque o cluster online novamente iniciando os serviços BeeGFS em todos os nós.

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

2. Verifique se o `beegfs-mgmtd` serviço foi iniciado com sucesso:

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

O resultado esperado inclui linhas como:

```
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 erros aparecerem nos logs do diário, revise os caminhos do arquivo de configuração de gerenciamento e certifique-se de que todos os valores foram transferidos corretamente do arquivo de configuração do BeeGFS 7.

3. Execute `pcs status` e verifique se o cluster está íntegro e se os serviços foram iniciados nos nós preferenciais.
4. Depois que o cluster for verificado como íntegro, reative o STONITH:

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

5. Prossiga para a próxima seção para atualizar os clientes BeeGFS no cluster e verificar a integridade do cluster BeeGFS.

Atualizar clientes BeeGFS

Após atualizar com sucesso seu cluster para BeeGFS v8, você também deve atualizar todos os clientes BeeGFS.

Os passos a seguir descrevem o processo de upgrade dos clientes BeeGFS em um sistema baseado em Ubuntu.

1. Caso ainda não o tenha feito, pare o serviço do cliente BeeGFS:

```
systemctl stop beegfs-client
```

2. Adicione o repositório do pacote BeeGFS v8 para sua distribuição Linux. Instruções para usar os repositórios oficiais do BeeGFS podem ser encontradas em "^Página de download do BeeGFS". Caso contrário, configure seu repositório local BeeGFS de acordo.

   Os passos a seguir utilizam o repositório oficial BeeGFS 8.2 em um sistema baseado em Ubuntu:

3. Importe a chave GPG do BeeGFS:

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

4. Baixe o arquivo do repositório:

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

   Remova quaisquer repositórios BeeGFS previamente configurados para evitar conflitos com o novo repositório BeeGFS v8.

5. Atualize os pacotes do cliente BeeGFS:

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

6. Configure o TLS para o cliente. O TLS é necessário para usar a CLI do BeeGFS. Consulte o "Configurar criptografia TLS para BeeGFS 8" procedimento para configurar o TLS no cliente.
7. Inicie o serviço cliente BeeGFS:

```
systemctl start beegfs-client
```

Verifique a atualização

Após concluir a atualização para BeeGFS v8, execute os seguintes comandos para verificar se a atualização foi bem-sucedida.

1. Verifique se o inode raiz pertence ao mesmo nó de metadados de antes. Isso deve acontecer automaticamente se você utilizou a funcionalidade `import-from-v7` no serviço de gerenciamento:

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

2. Verifique se todos os nós e alvos estão online e em bom estado:

```
beegfs health check
```

> Se a verificação de "Capacidade Disponível" alertar que os destinos têm pouco espaço livre, você pode ajustar os limites do "pool de capacidade" definidos no arquivo `beegfs-mgmtd.toml` para que se adequem melhor ao seu ambiente.

Atualize os pacotes Pacemaker e Corosync em um cluster HA

Siga estas etapas para atualizar os pacotes Pacemaker e Corosync em um cluster HA.

Visão geral

A atualização da Pacemaker e do Corosync garante que o cluster se beneficie de novos recursos, patches de segurança e melhorias de desempenho.

Abordagem de atualização

Há duas abordagens recomendadas para atualizar um cluster: Uma atualização contínua ou um desligamento completo do cluster. Cada abordagem tem suas próprias vantagens e desvantagens. O procedimento de atualização pode variar dependendo da versão de lançamento do pacemaker. Consulte a documentação do ClusterLabs "Atualizando um cluster de pacemaker"para determinar qual abordagem usar. Antes de seguir uma abordagem de atualização, verifique se:

- Os novos pacotes de pacemaker e Corosync são compatíveis com a solução BeeGFS da NetApp.
- Existem backups válidos para o sistema de arquivos BeeGFS e a configuração do cluster do pacemaker.
- O cluster está em um estado saudável.

Atualização progressiva

Esse método envolve remover cada nó do cluster, atualizá-lo e reintroduzi-lo no cluster até que todos os nós executem a nova versão. Essa abordagem mantém o cluster operacional, ideal para clusters de HA maiores, mas corre o risco de executar versões mistas durante o processo. Essa abordagem deve ser evitada em um

cluster de dois nós.

1. Confirme se o cluster está no estado ideal, com cada serviço BeeGFS sendo executado no nó preferido. "Examine o estado do cluster"Consulte para obter detalhes.
2. Para que o nó seja atualizado, coloque-o no modo de espera para drenar (ou mover) todos os serviços do BeeGFS:

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

3. Verifique se os serviços do nó foram drenados executando:

```
pcs status
```

   Certifique-se de que nenhum serviço é comunicado como `Started` no nó em espera.

   Dependendo do tamanho do cluster, os serviços podem levar segundos ou minutos para serem movidos para o nó irmão. Se um serviço BeeGFS não iniciar no nó irmão, consulte o "Guias de solução de problemas".

4. Encerre o cluster no nó:

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

5. Atualize os pacotes Pacemaker, Corosync e PCs no nó:

   Os comandos do gerenciador de pacotes variam de acordo com o sistema operacional. Os comandos a seguir são para sistemas que executam RHEL 8 e posteriores.

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

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

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

6. Inicie os serviços de cluster do pacemaker no nó:

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

7. Se o `pcs` pacote foi atualizado, reautentique o nó com o cluster:

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

8. Verifique se a configuração do pacemaker ainda é válida com a `crm_verify` ferramenta.

   Isso só precisa ser verificado uma vez durante a atualização do cluster.

```
crm_verify -L -V
```

9. Retire o nó do modo de espera:

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

10. Realocar todos os serviços BeeGFS de volta para o nó preferido:

```
pcs resource relocate run
```

11. Repita as etapas anteriores para cada nó no cluster até que todos os nós estejam executando as versões de pacemaker, Corosync e PCs desejadas.
12. Finalmente, execute `pcs status` e verifique se o cluster está saudável e os `Current DC` relatórios da versão desejada do pacemaker.

   Se o `Current DC` relatório 'versão em caixa', um nó no cluster ainda está em execução com a versão anterior do pacemaker e precisa ser atualizado. Se qualquer nó atualizado não conseguir ingressar novamente no cluster ou se os recursos não forem iniciados, verifique os logs do cluster e consulte as notas de versão do pacemaker ou os guias do usuário para obter problemas de atualização conhecidos.

Encerramento completo do cluster

Nesta abordagem, todos os nós e recursos do cluster são desligados, os nós são atualizados e, em seguida, o cluster é reiniciado. Essa abordagem é necessária se as versões Pacemaker e Corosync não suportarem uma configuração de versão mista.

1. Confirme se o cluster está no estado ideal, com cada serviço BeeGFS sendo executado no nó preferido. "Examine o estado do cluster"Consulte para obter detalhes.
2. Encerre o software do cluster (Pacemaker e Corosync) em todos os nós.

   Dependendo do tamanho do cluster, pode levar segundos ou minutos para que todo o cluster pare.

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

3. Uma vez que os serviços de cluster sejam desativados em todos os nós, atualize os pacotes Pacemaker, Corosync e PCs em cada nó de acordo com suas necessidades.

> Os comandos do gerenciador de pacotes variam de acordo com o sistema operacional. Os comandos a seguir são para sistemas que executam RHEL 8 e posteriores.

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

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

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

4. Depois de atualizar todos os nós, inicie o software de cluster em todos os nós:

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

5. Se o `pcs` pacote foi atualizado, reautentique cada nó no cluster:

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

6. Finalmente, execute `pcs status` e verifique se o cluster está saudável e os `Current DC` relatórios da versão correta do pacemaker.

> Se o `Current DC` relatório 'versão em caixa', um nó no cluster ainda está em execução com a versão anterior do pacemaker e precisa ser atualizado.

Atualize o firmware do adaptador do nó de arquivo

Siga estas etapas para atualizar os adaptadores ConnectX-7 do nó do arquivo para o firmware mais recente.

Visão geral

A atualização do firmware do adaptador ConnectX-7 pode ser necessária para suportar um novo driver MLNX_OFED, habilitar novos recursos ou corrigir bugs. Este guia usará o utilitário do NVIDIA `mlxfwmanager` para atualizações de adaptadores devido à sua facilidade de uso e eficiência.

Considerações sobre a atualização

Este guia aborda duas abordagens para atualizar o firmware do adaptador ConnectX-7: Uma atualização contínua e uma atualização de cluster de dois nós. Escolha a abordagem de atualização apropriada de acordo com o tamanho do cluster. Antes de executar atualizações de firmware, verifique se:

- Um driver MLNX_OFED suportado está instalado, consulte o "requisitos de tecnologia".
- Existem backups válidos para o sistema de arquivos BeeGFS e a configuração do cluster do pacemaker.
- O cluster está em um estado saudável.

Preparação da atualização de firmware

Recomenda-se usar o utilitário do NVIDIA `mlxfwmanager` para atualizar o firmware do adaptador de um nó, que é fornecido com o driver MLNX_OFED da NVIDIA. Antes de iniciar as atualizações, baixe a imagem de firmware do adaptador "Site de suporte da NVIDIA"e armazene-a em cada nó de arquivo.

Para adaptadores Lenovo ConnectX-7, use a `mlxfwmanager_LES` ferramenta, que está disponível na página do NVIDIA"Firmware OEM".

Abordagem de atualização progressiva

Essa abordagem é recomendada para qualquer cluster de HA com mais de dois nós. Essa abordagem envolve a atualização do firmware do adaptador em um nó de arquivo de cada vez, permitindo que o cluster de HA mantenha solicitações de manutenção, embora seja recomendável evitar a manutenção de e/S durante esse período.

1. Confirme se o cluster está no estado ideal, com cada serviço BeeGFS sendo executado no nó preferido. "Examine o estado do cluster"Consulte para obter detalhes.
2. Escolha um nó de arquivo a ser atualizado e coloque-o no modo de espera, que drena (ou move) todos os serviços BeeGFS desse nó:

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

3. Verifique se os serviços do nó foram drenados executando:

   ```
   pcs status
   ```

   Verifique se nenhum serviço está relatando como `Started` no nó em espera.

   

   Dependendo do tamanho do cluster, os serviços do BeeGFS podem levar segundos ou minutos para o nó irmão. Se um serviço BeeGFS não iniciar no nó irmão, consulte o "Guias de solução de problemas".

4. Atualize o firmware do adaptador usando `mlxfwmanager`o .

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

   Observe o `PCI Device Name` para cada adaptador que recebe atualizações de firmware.

5. Redefina cada adaptador usando o `mlxfwreset` utilitário para aplicar o novo firmware.

Algumas atualizações de firmware podem exigir uma reinicialização para aplicar a atualização. "Limitações de mlxfwreset do NVIDIA"Consulte para obter orientação. Se for necessária uma reinicialização, execute uma reinicialização em vez de redefinir os adaptadores.

a. Pare o serviço opensm:

```
systemctl stop opensm
```

b. Execute o comando a seguir para cada `PCI Device Name` observado anteriormente.

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

c. Inicie o serviço opensm:

```
systemctl start opensm
```

d. Reinicie o `eseries_nvme_ib.service` .

```
systemctl restart eseries_nvme_ib.service
```

e. Verifique se os volumes do array de armazenamento da Série E estão presentes.

```
multipath -ll
```

1. Execute `ibstat` e verifique se todos os adaptadores estão sendo executados na versão de firmware desejada:

```
ibstat
```

2. Inicie os serviços de cluster do pacemaker no nó:

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

3. Retire o nó do modo de espera:

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

4. Realocar todos os serviços BeeGFS de volta para o nó preferido:

```
pcs resource relocate run
```

Repita estas etapas para cada nó de arquivo no cluster até que todos os adaptadores tenham sido atualizados.

Abordagem de atualização de cluster de dois nós

Essa abordagem é recomendada para clusters de HA com apenas dois nós. Essa abordagem é semelhante a uma atualização contínua, mas inclui etapas adicionais para evitar o tempo de inatividade do serviço quando os serviços de cluster de um nó são interrompidos.

1. Confirme se o cluster está no estado ideal, com cada serviço BeeGFS sendo executado no nó preferido. "Examine o estado do cluster"Consulte para obter detalhes.
2. Escolha um nó de arquivo a ser atualizado e coloque o nó no modo de espera, que drena (ou move) todos os serviços BeeGFS desse nó:

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

3. Verifique se os recursos do nó foram drenados executando:

   ```
   pcs status
   ```

   Verifique se nenhum serviço está relatando como `Started` no nó em espera.

   > Dependendo do tamanho do cluster, os serviços BeeGFS podem levar segundos ou minutos para serem reportados como `Started` no nó secundário. Se um serviço BeeGFS não for iniciado, consulte o "Guias de solução de problemas".

4. Coloque o cluster no modo de manutenção.

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

5. Atualize o firmware do adaptador usando `mlxfwmanager`o .

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

   Observe o `PCI Device Name` para cada adaptador que recebe atualizações de firmware.

6. Redefina cada adaptador usando o `mlxfwreset` utilitário para aplicar o novo firmware.

Algumas atualizações de firmware podem exigir uma reinicialização para aplicar a atualização. "Limitações de mlxfwreset do NVIDIA"Consulte para obter orientação. Se for necessária uma reinicialização, execute uma reinicialização em vez de redefinir os adaptadores.

a. Pare o serviço opensm:

```
systemctl stop opensm
```

b. Execute o comando a seguir para cada `PCI Device Name` observado anteriormente.

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

c. Inicie o serviço opensm:

```
systemctl start opensm
```

7. Execute `ibstat` e verifique se todos os adaptadores estão sendo executados na versão de firmware desejada:

```
ibstat
```

8. Inicie os serviços de cluster do pacemaker no nó:

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

9. Retire o nó do modo de espera:

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

10. Retire o cluster do modo de manutenção.

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

11. Realocar todos os serviços BeeGFS de volta para o nó preferido:

```
pcs resource relocate run
```

Repita estas etapas para cada nó de arquivo no cluster até que todos os adaptadores tenham sido

atualizados.

Atualizar o storage array do e-Series

Siga estas etapas para atualizar os componentes do storage array e-Series do cluster HA.

Visão geral

Manter os storage arrays NetApp e-Series atualizados do seu cluster de HA com o firmware mais recente garante desempenho ideal e segurança aprimorada. As atualizações de firmware para a matriz de armazenamento são aplicadas através de ficheiros de firmware da unidade, NVSRAM e SANtricity os.

> Embora os storage arrays possam ser atualizados on-line com o cluster de HA, é recomendável colocar o cluster no modo de manutenção para todas as atualizações.

Bloquear etapas de atualização do nó

As etapas a seguir descrevem como atualizar o firmware dos storages de armazenamento usando a `Netapp_Eseries.Santricity` coleção Ansible. Antes de prosseguir, reveja o "Considerações sobre a atualização" para atualizar os sistemas e-Series.

> A atualização para o SANtricity os 11,80 ou versões posteriores só é possível a partir de 11.71.5P1. O storage array deve primeiro ser atualizado para 11.70.5P1 antes de aplicar novas atualizações.

1. Verifique se o nó de controle do Ansible está usando a coleção mais recente do SANtricity Ansible.
   - Para atualizações de coleção com acesso ao "Ansible Galaxy", execute o seguinte comando:

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

   - Para atualizações off-line, baixe o tarball de coleta de "Ansible Galaxy", transfira-o para o nó de controle e execute:

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

     Consulte "Instalando coleções" para obter mais informações.

2. Obtenha o firmware mais recente para a sua matriz de armazenamento e unidades.
   a. Transfira os ficheiros de firmware.
      - **SANtricity os e NVSRAM:** navegue até o "Site de suporte da NetApp" e faça o download da versão mais recente do SANtricity os e NVSRAM para seu modelo de storage array.
      - **Drive firmware:** navegue até o "Site de firmware de disco e-Series" e faça o download do firmware mais recente para cada um dos modelos de unidade da matriz de armazenamento.
   b. Armazene os arquivos de firmware da unidade, NVSRAM e do sistema operacional SANtricity no diretório do nó de controle do Ansible `<inventory_directory>/packages`.

3. Se necessário, atualize os arquivos de inventário do Ansible do cluster para incluir todos os storage arrays (nós de bloco) que exigem atualizações. Para obter orientação, consulte "Visão geral do Ansible Inventory"a secção.
4. Garantir que o cluster esteja no estado ideal com cada serviço BeeGFS no nó de sua preferência. "Examine o estado do cluster"Consulte para obter detalhes.
5. Coloque o cluster no modo de manutenção seguindo as instruções na "Coloque o cluster no modo de manutenção".
6. Crie um novo manual do Ansible chamado `update_block_node_playbook.yml`. Preencha o manual com o seguinte conteúdo, substituindo as versões do SANtricity os, NVSRAM e firmware da unidade para o caminho de atualização desejado:

```
- 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. Para iniciar as atualizações, execute o seguinte comando a partir do nó de controle do Ansible:

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

8. Depois que o manual de estratégia for concluído, verifique se cada storage array está no estado ideal.
9. Mova o cluster para fora do modo de manutenção e valide que o cluster está no estado ideal. Cada serviço BeeGFS está no nó de sua preferência.

Manutenção e manutenção

Serviços de failover e failback

Movimentação de serviços do BeeGFS entre nós de cluster.

Visão geral

Os serviços BeeGFS podem fazer o failover entre nós no cluster para garantir que os clientes possam

continuar acessando o sistema de arquivos em caso de falha ou que você precise executar a manutenção planejada. Esta seção descreve várias maneiras pelas quais os administradores podem curar o cluster depois de se recuperar de uma falha ou mover serviços manualmente entre nós.

Passos

Failover e failback

Failover (planejado)

Geralmente, quando você precisa colocar um único nó de arquivo off-line para manutenção, você vai querer mover (ou drenar) todos os serviços BeeGFS desse nó. Isso pode ser feito colocando primeiro o nó em standby:

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

Depois de verificar o uso `pcs status` de todos os recursos terem sido reiniciados no nó de arquivo alternativo, você pode encerrar ou fazer outras alterações no nó conforme necessário.

Failback (após um failover planejado)

Quando você estiver pronto para restaurar os serviços BeeGFS para o nó preferido, primeiro execute `pcs status` e verifique na "Lista de nós" o status está em espera. Se o nó foi reinicializado, ele será exibido offline até que você coloque os serviços de cluster online:

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

Quando o nó estiver online, retire-o do modo de espera com:

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

Por último, reposicione todos os serviços BeeGFS de volta aos seus nós preferidos com:

```
pcs resource relocate run
```

Failback (após um failover não planejado)

Se um nó apresentar uma falha de hardware ou outra, o cluster de HA deve reagir automaticamente e mover seus serviços para um nó íntegro, fornecendo tempo para os administradores tomarem as medidas corretivas. Antes de prosseguir, consulte "solução de problemas"a seção para determinar a causa do failover e resolver quaisquer problemas pendentes. Depois que o nó estiver ligado novamente e saudável, você poderá prosseguir com o failback.

Quando um nó é inicializado após uma reinicialização não planejada (ou planejada), os serviços de cluster não são configurados para serem iniciados automaticamente, então você primeiro precisará colocar o nó on-line com:

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

Em seguida, limpe todas as falhas de recursos e redefina o histórico de esgrima do nó:

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

Verifique se `pcs status` o nó está on-line e saudável. Por padrão, os serviços BeeGFS não irão fazer o failback automaticamente para evitar mover acidentalmente recursos de volta para um nó que não está saudável. Quando estiver pronto, retorne todos os recursos no cluster de volta aos nós preferidos com:

```
pcs resource relocate run
```

Movendo serviços individuais BeeGFS para nós de arquivos alternativos

Mova permanentemente um serviço BeeGFS para um novo nó de arquivo

Se você quiser alterar permanentemente o nó de arquivo preferido para um serviço BeeGFS individual, ajuste o inventário do Ansible para que o nó preferido seja listado primeiro e execute novamente o manual de estratégia do Ansible.

Por exemplo, neste arquivo de exemplo `inventory.yml`, beegfs_01 é o nó de arquivo preferido para executar o serviço de gerenciamento BeeGFS:

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

Reverter a ordem faria com que os serviços de gerenciamento fossem preferidos no beegfs_02:

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

Mova temporariamente um serviço BeeGFS para um nó de arquivo alternativo

Geralmente, se um nó estiver em manutenção, você vai querer usar as [etapas de failover e failback] para afastar todos os serviços desse nó.

Se, por algum motivo, você precisar mover um serviço individual para uma execução diferente do nó de arquivo:

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

Não especifique recursos individuais ou o grupo de recursos. Sempre especifique o nome do monitor para o serviço BeeGFS que deseja realocar. Por exemplo, para mover o serviço de gerenciamento BeeGFS para beegfs_02 execute: `pcs resource move mgmt-monitor beegfs_02`. Esse processo pode ser repetido para afastar um ou mais serviços de seus nós preferidos. Verifique se o uso `pcs status` dos serviços foi relocado/iniciado no novo nó.

Para mover um serviço BeeGFS de volta para o nó preferido, primeiro limpe as restrições de recursos temporários (repetindo essa etapa conforme necessário para vários serviços):

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

Então, quando estiver pronto para realmente mover o(s) serviço(s) de volta para o(s) nó(s) preferido(s) executado(s):

```
pcs resource relocate run
```

Observação esse comando irá realocar quaisquer serviços que não tenham mais restrições de recursos temporários que não estejam localizados em seus nós preferidos.

Coloque o cluster no modo de manutenção

Impedir que o cluster de HA reaja acidentalmente às alterações pretendidas no ambiente.

Visão geral

Colocar o cluster no modo de manutenção desativa todo o monitoramento de recursos e impede que o pacemaker mova ou gerencie recursos no cluster. Todos os recursos permanecerão em execução em seus nós originais, independentemente de haver uma condição de falha temporária que os impeça de serem acessíveis. Os cenários em que isso é recomendado/útil incluem:

- Manutenção de rede que pode interromper temporariamente as conexões entre nós de arquivo e serviços BeeGFS.
- Atualizações do nó de bloco.
- File Node sistema operacional, kernel ou outras atualizações de pacote.

Geralmente, a única razão para colocar manualmente o cluster no modo de manutenção é impedir que ele reaja a alterações externas no ambiente. Se um nó individual no cluster exigir reparo físico, não use o modo de manutenção e simplesmente coloque esse nó em espera seguindo o procedimento acima. Observe que a reexecução do Ansible coloca o cluster automaticamente em modo de manutenção, facilitando a manutenção de software, incluindo atualizações e alterações de configuração.

Passos

Para verificar se o cluster está no modo de manutenção, execute:

```
pcs property config
```

A `maintenance-mode` propriedade não aparecerá se o cluster estiver operando normalmente. Se o cluster estiver no modo de manutenção, a propriedade será reportada como `true`. Para ativar a execução do modo de manutenção:

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

Você pode verificar executando o status dos PCs e garantindo que todos os recursos mostrem "(unmanaged)". Para retirar o cluster do modo de manutenção, execute:

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

Pare e inicie o cluster

Parar e iniciar o cluster HA com simplicidade.

Visão geral

Esta seção descreve como encerrar e reiniciar o cluster BeeGFS com simplicidade. Exemplos de cenários onde isso pode ser necessário incluem manutenção elétrica ou migração entre data centers ou racks.

Passos

Se, por algum motivo, você precisar parar todo o cluster BeeGFS e encerrar todos os serviços serão executados:

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

Também é possível parar o cluster em nós individuais (que farão failover automático de serviços para outro nó), embora seja recomendável colocar primeiro o nó em standby (consulte "failover"a seção):

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

Para iniciar os serviços e recursos do cluster em todos os nós executados:

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

Ou inicie serviços em um nó específico com:

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

Nesse momento, execute `pcs status` e verifique se o cluster e os serviços BeeGFS começam em todos os nós, e os serviços estão sendo executados nos nós que você espera.

> Dependendo do tamanho do cluster, pode levar segundos ou minutos para que todo o cluster pare, ou mostre como iniciado em `pcs status`. se `pcs cluster <COMMAND>` travar por mais de cinco minutos, antes de executar "Ctrl C" para cancelar o comando, faça login em cada nó do cluster e use `pcs status` para ver se os serviços de cluster (Corosync/Pacemaker) ainda estão em execução nesse nó. De qualquer nó em que o cluster ainda esteja ativo, você pode verificar quais recursos estão bloqueando o cluster. Aborde manualmente o problema e o comando deve ser concluído ou pode ser executado novamente para parar quaisquer serviços restantes.

Substituir nós de arquivo

Substituindo um nó de arquivo se o servidor original estiver com defeito.

Visão geral

Esta é uma visão geral das etapas necessárias para substituir um nó de arquivo no cluster. Essas etapas presumem que o nó do arquivo falhou devido a um problema de hardware e foi substituído por um novo nó de arquivo idêntico.

Passos:

1. Substitua fisicamente o nó de arquivo e restaure todo o cabeamento para o nó de bloco e rede de armazenamento.
2. Reinstale o sistema operacional no nó de arquivo, incluindo a adição de assinaturas Red Hat.
3. Configurar o gerenciamento e a rede BMC no nó de arquivo.
4. Atualize o inventário do Ansible se o nome de host, IP, mapeamentos de interface PCIe para lógica ou qualquer outra coisa mudou sobre o novo nó de arquivo. Geralmente, isso não é necessário se o nó foi substituído por hardware de servidor idêntico e você estiver usando a configuração de rede original.
   a. Por exemplo, se o nome do host mudou, crie (ou renomeie) o arquivo de inventário do nó (`host_vars/<NEW_NODE>.yml`) e, em seguida, no arquivo de inventário do Ansible (`inventory.yml`), substitua o nome do nó antigo pelo novo nome do nó:

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

5. De um dos outros nós no cluster, remova o nó antigo: `pcs cluster node remove <HOSTNAME>`.

   NÃO PROSSIGA ANTES DE EXECUTAR ESTE PASSO.

6. No nó de controle do Ansible:
   a. Remova a chave SSH antiga com:

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

   b. Configure o SSH sem senha para o nó Substituir por:

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

7. Execute novamente o manual de estratégia do Ansible para configurar o nó e adicioná-lo ao cluster:

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

8. Neste ponto, execute `pcs status` e verifique se o nó substituído está listado e executando serviços.

Expanda ou diminua o cluster

Adicione ou remova blocos de construção do cluster.

Visão geral

Esta seção documenta várias considerações e opções para ajustar o tamanho do cluster BeeGFS HA. Normalmente, o tamanho do cluster é ajustado adicionando ou removendo componentes básicos, que geralmente são dois nós de arquivo configurados como um par de HA. Também é possível adicionar ou remover nós de arquivo individuais (ou outros tipos de nós de cluster), se necessário.

Adicionando um Building Block ao cluster

Considerações

Aumentar o cluster adicionando componentes básicos adicionais é um processo simples. Antes de começar, lembre-se das restrições relativas ao número mínimo e máximo de nós de cluster em cada cluster de HA individual e determine se você deve adicionar nós ao cluster de HA existente ou criar um novo cluster de HA. Normalmente, cada componente básico consiste em dois nós de arquivo, mas três nós é o número mínimo de nós por cluster (para estabelecer quorum) e dez é o máximo recomendado (testado). Para cenários avançados, é possível adicionar um único nó "tiebreaker" que não executa nenhum serviço BeeGFS ao implantar um cluster de dois nós. Entre em Contato com o suporte da NetApp se você estiver pensando em tal implantação.

Tenha em mente essas restrições e qualquer crescimento futuro esperado de cluster ao decidir como expandir o cluster. Por exemplo, se você tiver um cluster de seis nós e precisar adicionar mais quatro nós, seria recomendável apenas iniciar um novo cluster de HA.

Lembre-se: Um único sistema de arquivos BeeGFS pode consistir em vários clusters de HA independentes. Isso permite que os sistemas de arquivos continuem dimensionando muito além dos limites recomendados/físicos dos componentes subjacentes do cluster de HA.

Passos

Ao adicionar um componente básico ao cluster, você precisará criar os `host_vars` arquivos para cada um dos novos nós de arquivo e nós de bloco (arrays e-Series). Os nomes desses hosts precisam ser adicionados ao inventário, juntamente com os novos recursos que devem ser criados. Os arquivos correspondentes `group_vars` precisarão ser criados para cada novo recurso. Consulte "use arquiteturas personalizadas"a secção para obter detalhes.

Depois de criar os arquivos corretos, tudo o que é necessário é executar novamente a automação usando o comando:

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

Removendo um Building Block do cluster

Há uma série de considerações a ter em mente quando você precisa aposentar um bloco de construção, por exemplo:

- Quais serviços BeeGFS estão sendo executados nesse componente básico?
- Apenas os nós de arquivo estão se aposentando e os nós de bloco devem ser anexados a novos nós de arquivo?
- Se todo o componente básico estiver sendo aposentado, os dados devem ser movidos para um novo componente básico, dispersos em nós existentes no cluster ou movidos para um novo sistema de arquivos BeeGFS ou outro sistema de storage?
- Isso pode acontecer durante uma interrupção ou deve ser feito sem interrupções?
- O componente básico está ativamente em uso ou contém dados que não estão mais ativos?

Devido aos diversos pontos de partida possíveis e aos estados finais desejados, entre em Contato com o suporte da NetApp para que possamos identificar e ajudar a implementar a melhor estratégia com base em seu ambiente e requisitos.

Solucionar problemas

Solução de problemas de um cluster BeeGFS HA.

Visão geral

Esta seção descreve como investigar e solucionar problemas de várias falhas e outros cenários que podem surgir ao operar um cluster BeeGFS HA.

Guias de solução de problemas

Investigando failovers inesperados

Quando um nó é fechado inesperadamente e seus serviços são movidos para outro nó, a primeira etapa deve ser verificar se o cluster indica falhas de recursos na parte inferior `pcs status`do . Normalmente, nada estará presente se o esgrima for concluído com sucesso e os recursos forem reiniciados em outro nó.

Geralmente, o próximo passo será pesquisar através dos logs do systemd usando `journalctl` em qualquer um dos nós de arquivo restantes (os logs do pacemaker são sincronizados em todos os nós). Se você souber a hora em que ocorreu a falha, você pode iniciar a pesquisa imediatamente antes da falha ocorrer (geralmente, pelo menos dez minutos antes é recomendado):

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

As seções a seguir mostram o texto comum que você pode grep nos logs para restringir ainda mais a investigação.

Passos para investigar/resolver

Passo 1: Verifique se o monitor BeeGFS detetou uma falha:

Se o failover tiver sido acionado pelo monitor BeeGFS, você verá um erro (se não prosseguir para a próxima etapa).

```
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
```

Neste caso, o serviço BeeGFS meta_08 parou por algum motivo. Para continuar a solução de problemas, devemos inicializar o beegfs_02 e revisar os logs do serviço em `/var/log/beegfs-meta-meta_08_tgt_0801.log`. Por exemplo, o serviço BeeGFS pode ter encontrado um erro de aplicação devido a um problema interno ou problema com o nó.

Diferentemente dos logs do pacemaker, os logs dos serviços BeeGFS não são distribuídos para todos os nós do cluster. Para investigar esses tipos de falhas, os logs do nó original onde a falha ocorreu são necessários.

Possíveis problemas que podem ser relatados pelo monitor incluem:

- O(s) alvo(s) não estão acessíveis!
  - Descrição: Indica que os volumes de bloco não estavam acessíveis.
  - Resolução de problemas:
    - Se o serviço também não conseguir iniciar no nó de arquivo alternativo, confirme se o nó de bloco está em bom estado.
    - Verifique se há problemas físicos que impeçam o acesso aos nós de bloco a partir deste nó de arquivo, por exemplo, adaptadores InfiniBand com defeito ou cabos.

- A rede não está acessível!
  - Descrição: Nenhum dos adaptadores usados pelos clientes para se conetar a este serviço BeeGFS estava online.
  - Resolução de problemas:
    - Se vários/todos os nós de arquivo foram afetados, verifique se houve uma falha na rede usada para conetar os clientes BeeGFS e o sistema de arquivos.
    - Verifique se há problemas físicos que impeçam o acesso aos clientes a partir deste nó de arquivo, por exemplo, adaptadores InfiniBand ou cabos defeituosos.
- O serviço BeeGFS não está ativo!
  - Descrição: Um serviço BeeGFS parou inesperadamente.
  - Resolução de problemas:
    - No nó de arquivo que relatou o erro, verifique os logs para o serviço BeeGFS impactado para ver se ele relatou uma falha. Se isso aconteceu, abra um caso com o suporte do NetApp para que a falha possa ser investigada.
    - Se não houver erros relatados no log BeeGFS, verifique os logs do diário para ver se systemd registrou um motivo pelo qual o serviço foi interrompido. Em alguns cenários, o serviço BeeGFS pode não ter tido a chance de Registrar quaisquer mensagens antes do processo ser encerrado (por exemplo, se alguém executou `kill -9 <PID>`).

Etapa 2: Verifique se o nó deixou o cluster inesperadamente

Caso o nó tenha sofrido alguma falha catastrófica de hardware (por exemplo, a placa do sistema morreu) ou tenha ocorrido um problema de pânico do kernel ou de software semelhante, o monitor BeeGFS não reportará um erro. Em vez disso, procure o nome do host e você deve ver mensagens do Pacemaker indicando que o nó foi perdido inesperadamente:

```
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
```

Passo 3: Verifique se o pacemaker foi capaz de cercar o nó

Em todos os cenários, você deve ver o pacemaker tentar cercar o nó para verificar se ele está realmente offline (as mensagens exatas podem variar por causa da esgrima):

```
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 a ação de esgrima for concluída com sucesso, você verá mensagens como:

```
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 a ação de esgrima falhou por algum motivo, os serviços BeeGFS não poderão reiniciar em outro nó para evitar o risco de corrupção de dados. Isso seria um problema para investigar separadamente, se, por exemplo, o dispositivo de vedação (PDU ou BMC) estivesse inacessível ou mal configurado.

Ações recurso Falha Endereço (encontradas na parte inferior do status PCs)

Se um recurso necessário para executar um serviço BeeGFS falhar, um failover será acionado pelo monitor BeeGFS. Se isso ocorrer, provavelmente não haverá "ações de recurso com falha" listadas na parte inferior do `pcs status` e você deve consultar as etapas sobre como "failback após um failover não planejado".

Caso contrário, geralmente deve haver apenas dois cenários onde você verá "ações de recurso falhadas".

Passos para investigar/resolver

Cenário 1: Um problema temporário ou permanente foi detetado com um agente de esgrima e foi reiniciado ou movido para outro nó.

Alguns agentes de vedação são mais confiáveis do que outros, e cada um implementará seu próprio método de monitoramento para garantir que o dispositivo de vedação esteja pronto. Em particular, o agente de esgrima do redfish foi visto para relatar ações de recursos falhadas como as seguintes, mesmo que ele ainda mostre iniciado:

```
  * 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
```

Não é esperado que um agente de esgrima que relata ações de recursos com falha em um nó específico acione um failover dos serviços BeeGFS executados nesse nó. Ele deve simplesmente ser reiniciado automaticamente no mesmo nó ou em um nó diferente.

Passos para resolver:

1. Se o agente de esgrima se recusar a executar consistentemente em todos ou em um subconjunto de nós, verifique se esses nós são capazes de se conetar ao agente de esgrima e verifique se o agente de esgrima está configurado corretamente no inventário do Ansible.
   a. Por exemplo, se um agente de esgrima de peixe vermelho (BMC) estiver sendo executado no mesmo nó que é responsável por esgrima, e o gerenciamento de SO e IPs BMC estiverem na mesma

interface física, algumas configurações de switch de rede não permitirão a comunicação entre as duas interfaces (para evitar loops de rede). Por padrão, o cluster de HA tentará evitar colocar agentes de vedação no nó que são responsáveis por cercas, mas isso pode acontecer em alguns cenários/configurações.

2. Uma vez que todos os problemas são resolvidos (ou se o problema parecia efêmero), execute `pcs resource cleanup` para redefinir as ações de recursos com falha.

Cenário 2: O monitor BeeGFS detetou um problema e acionou um failover, mas por algum motivo os recursos não puderam ser iniciados em um nó secundário.

Desde que o esgrima esteja habilitado e o recurso não tenha sido bloqueado de parar no nó original (consulte a seção de solução de problemas para "standby (on-fail)"), as razões mais prováveis incluem problemas para iniciar o recurso em um nó secundário porque:

- O nó secundário já estava offline.
- Um problema de configuração físico ou lógico impediu que o secundário acessasse os volumes de bloco usados como destinos BeeGFS.

Passos para resolver:

1. Para cada entrada nas ações de recursos com falha:
   a. Confirme se a ação de recurso falhou foi uma operação de início.
   b. Com base no recurso indicado e no nó especificado nas ações de recurso com falha:
      i. Procure e corrija quaisquer problemas externos que impeçam o nó de iniciar o recurso especificado. Por exemplo, se o endereço IP BeeGFS (IP flutuante) não foi iniciado, verifique se pelo menos uma das interfaces necessárias está conetada/on-line e cabeada ao switch de rede direito. Se um destino BeeGFS (dispositivo de bloco/volume e-Series) falhar, verifique se as conexões físicas com os nós de bloco de back-end estão conetadas conforme o esperado e verifique se os nós de bloco estão íntegros.
   c. Se não houver problemas externos óbvios e você desejar uma causa raiz para esse incidente, é sugerido que você abra um caso com suporte do NetApp para investigar antes de prosseguir, pois as etapas a seguir podem tornar a análise de causa raiz (RCA) desafiadora/impossível.
2. Depois de resolver quaisquer problemas externos:
   a. Comente todos os nós não funcionais do arquivo Ansible inventory.yml e execute novamente o manual completo do Ansible para garantir que toda a configuração lógica esteja configurada corretamente nos nós secundários.
      i. Observação: Não se esqueça de descomentar esses nós e executar novamente o manual de estratégia quando os nós estiverem saudáveis e você estiver pronto para o failback.
   b. Como alternativa, você pode tentar recuperar manualmente o cluster:
      i. Coloque todos os nós offline de volta online usando: `pcs cluster start <HOSTNAME>`
      ii. Limpar todas as ações de recursos com falha usando: `pcs resource cleanup`
      iii. Execute o status dos PCs e verifique se todos os serviços começam conforme esperado.
      iv. Se necessário, execute `pcs resource relocate run` para mover os recursos de volta para o nó preferido (se ele estiver disponível).

Questões comuns

Os serviços BeeGFS não fazem failover ou failback quando solicitados

- Problema provável:* o `pcs resource relocate` comando run foi executado, mas nunca terminou com sucesso.

Como verificar: Executar `pcs constraint --full` e verificar quaisquer restrições de localização com um ID de `pcs-relocate-<RESOURCE>`.

How to resolve: execute `pcs resource relocate clear` e execute novamente `pcs constraint --full` para verificar se as restrições extras são removidas.

Um nó no estado dos PCes mostra "standby (on-fail)" quando a vedação está desativada

Problema provável: o pacemaker não conseguiu confirmar com êxito todos os recursos foram parados no nó que falhou.

Como resolver:

1. Execute `pcs status` e verifique se há recursos que não são "iniciados" ou mostram erros na parte inferior da saída e resolva quaisquer problemas.
2. Para colocar o nó novamente online, execute `pcs resource cleanup --node=<HOSTNAME>`.

Após um failover inesperado, os recursos mostram "Started (on-fail)" no status PCs quando o esgrima está ativado

Problema provável: ocorreu Um problema que desencadeou um failover, mas o pacemaker não conseguiu verificar se o nó estava vedado. Isso pode acontecer porque o esgrima foi mal configurado ou houve um problema com o agente de esgrima (exemplo: O PDU foi desconetado da rede).

Como resolver:

1. Verifique se o nó está realmente desligado.

> Se o nó especificado não estiver realmente desativado, mas executando serviços ou recursos de cluster, ocorrerá corrupção de dados/falha de cluster.

2. Confirme manualmente a vedação com: `pcs stonith confirm <NODE>`

Neste ponto, os serviços devem terminar de falhar e ser reiniciados em outro nó saudável.

Tarefas comuns de resolução de problemas

Reinicie os serviços BeeGFS individuais

Normalmente, se um serviço BeeGFS precisar ser reiniciado (por exemplo, para facilitar uma alteração de configuração), isso deve ser feito atualizando o inventário do Ansible e executando novamente o manual de estratégia. Em alguns cenários, pode ser desejável reiniciar serviços individuais para facilitar a solução de problemas mais rápida, por exemplo, para alterar o nível de log sem precisar esperar que todo o manual de estratégia seja executado.

A menos que quaisquer alterações manuais também sejam adicionadas ao inventário do Ansible, elas serão revertidas na próxima vez que o manual de estratégia do Ansible for executado.

Opção 1: Reinício controlado pelo sistema

Se houver um risco de o serviço BeeGFS não reiniciar corretamente com a nova configuração, primeiro coloque o cluster no modo de manutenção para impedir que o monitor BeeGFS detete que o serviço seja interrompido e acione um failover indesejado:

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

Se necessário, faça alterações na configuração dos serviços em `/mnt/<SERVICE_ID>/_config/beegfs-.conf` (exemplo: `/mnt/meta_01_tgt_0101/metadata_config/beegfs-meta.conf`), em seguida, use systemd para reiniciá-lo:

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

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

Opção 2: Reinício controlado pelo pacemaker

Se você não estiver preocupado com a nova configuração pode fazer com que o serviço pare inesperadamente (por exemplo, simplesmente mudando o nível de log), ou você está em uma janela de manutenção e não está preocupado com o tempo de inatividade, você pode simplesmente reiniciar o monitor BeeGFS para o serviço que deseja reiniciar:

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

Por exemplo, para reiniciar o serviço de gerenciamento BeeGFS: `pcs resource restart mgmt-monitor`

Informações sobre direitos autorais

Informações sobre marcas comerciais