



Defina el sistema de archivos BeeGFS

BeeGFS on NetApp with E-Series Storage

NetApp
March 21, 2024

Tabla de contenidos

- Defina el sistema de archivos BeeGFS 1
 - Descripción general del inventario de Ansible 1
 - Planifique el sistema de archivos 1
 - Defina los nodos de archivo y bloque 3
 - Defina los servicios BeeGFS 20
 - Asigne servicios BeeGFS a los nodos de archivo 25

Defina el sistema de archivos BeeGFS

Descripción general del inventario de Ansible

El inventario de Ansible es un conjunto de archivos de configuración que definen el clúster de alta disponibilidad de BeeGFS deseado.

Descripción general

Se recomienda seguir las prácticas de Ansible estándar para organizar el "inventario", incluido el uso de "subdirectorios/archivos" en lugar de almacenar todo el inventario en un archivo.

El inventario de Ansible para un único clúster de alta disponibilidad de BeeGFS está organizado de la siguiente forma:

[Descripción general del inventario de Ansible]



Dado que un único sistema de archivos BeeGFS puede abarcar varios clústeres de alta disponibilidad, es posible que las instalaciones de gran tamaño tengan varios inventarios de Ansible. Por lo general, no se recomienda intentar definir varios clústeres de alta disponibilidad como un único inventario de Ansible para evitar problemas.

Pasos

1. En el nodo de control de Ansible, cree un directorio vacío que contendrá el inventario de Ansible para el clúster de BeeGFS que desea implementar.
 - a. Si su sistema de archivos contendrá en algún momento varios clústeres de alta disponibilidad, se recomienda crear primero un directorio para el sistema de archivos y, a continuación, subdirectorios para el inventario que represente cada clúster de alta disponibilidad. Por ejemplo:

```
beegfs_file_system_1/  
  beegfs_cluster_1/  
  beegfs_cluster_2/  
  beegfs_cluster_N/
```

2. En el directorio que contiene el inventario del clúster de alta disponibilidad que desea implementar, cree dos directorios `group_vars` y `host_vars` y dos archivos `inventory.yml` y `playbook.yml`.

En las siguientes secciones se describe la definición del contenido de cada uno de estos archivos.

Planifique el sistema de archivos

Planifique la puesta en marcha del sistema de archivos antes de crear el inventario de Ansible.

Descripción general

Antes de implementar el sistema de archivos, debe definir qué direcciones IP, puertos y otra configuración serán necesarias para todos los nodos de archivos, nodos de bloques y servicios BeeGFS que se ejecuten en el clúster. Aunque la configuración exacta variará en función de la arquitectura del clúster, esta sección define las prácticas recomendadas y los pasos a seguir que son aplicables.

Pasos

1. Si utiliza un protocolo de almacenamiento basado en IP (como Iser, iSCSI, NVMe/IB o NVMe/roce) para conectar nodos de archivos a nodos de bloques, rellene la siguiente hoja de datos para cada elemento básico. Cada conexión directa en un único bloque de creación debería tener una subred única y no debería haber superposición con subredes utilizadas para la conectividad cliente-servidor.

Nodo de archivo	Puerto IB	Dirección IP	Nodo de bloques	Puerto IB	IP física	Virtual IP (solo para EF600 con HDR IB)
<HOSTNAME >	<PORT>	<IP/SUBNET >	<HOSTNAME >	<PORT>	<IP/SUBNET >	<IP/SUBNET >



Si los nodos de archivo y bloque de cada bloque de creación están conectados directamente, a menudo puede reutilizar las mismas IP/esquema para varios bloques de creación.

2. Independientemente de si utiliza InfiniBand o RDMA sobre Ethernet convergente (roce) para la red de almacenamiento, rellene la siguiente hoja de datos para determinar los rangos de IP que se usarán para los servicios de clúster de alta disponibilidad, los servicios de archivos BeeGFS y los clientes para comunicarse:

Específico	Puerto InfiniBand	Dirección IP o rango
IP(s) de clúster de BeeGFS	<INTERFACE(s)>	<RANGE>
Gestión de BeeGFS	<INTERFACE(s)>	<IP(s)>
Metadatos de BeeGFS	<INTERFACE(s)>	<RANGE>
Almacenamiento de BeeGFS	<INTERFACE(s)>	<RANGE>
Clientes BeeGFS	<INTERFACE(s)>	<RANGE>

- a. Si utiliza una sola subred IP, solo será necesaria una hoja de datos; de lo contrario, rellene también una hoja de cálculo para la segunda subred.
3. En función de lo anterior, para cada bloque de creación del clúster, rellene la siguiente hoja de trabajo que define qué servicios de BeeGFS se ejecutará. Para cada servicio, especifique los nodos de archivo preferidos/secundarios, el puerto de red, las IP flotantes, la asignación de zonas NUMA (si es necesario) y qué nodos de bloque se usarán para sus destinos. Consulte las siguientes directrices al rellenar la hoja de trabajo:
 - a. Especifique los servicios BeeGFS como cualquiera de los dos `mgmt_<ID>.yaml`, `meta_<ID>.yaml`, o `storage_<ID>.yaml` Donde ID representa un número único en todos los servicios BeeGFS de ese tipo en este sistema de archivos. Esta convención simplificará la referencia a esta hoja de trabajo en secciones posteriores mientras crea archivos para configurar cada servicio.

- b. Los puertos para los servicios BeeGFS sólo deben ser únicos en un bloque de construcción en particular. Asegúrese de que los servicios con el mismo número de puerto no pueden ejecutarse nunca en el mismo nodo de archivo para evitar conflictos de puertos.
- c. Si los servicios necesarios pueden utilizar volúmenes de más de un nodo de bloque o un pool de almacenamiento (y no todos los volúmenes deben ser propiedad de la misma controladora). Múltiples servicios también pueden compartir la misma configuración de nodo de bloque o pool de almacenamiento (se definirán los volúmenes individuales en una sección posterior).

Servicio BeeGFS (nombre de archivo)	Nodos de archivos	Puerto	IP flotantes	Zona NUMA	Nodo de bloques	Del banco de almacenamiento	Controladora propietaria
<SERVICE TYPE>_<ID>.yml	<PREFERRED FILE NODE> <SECONDARY FILE NODE(s)>	<PORT>	<INTERFACE>:<IP/SUBNET> <INTERFACE>:<IP/SUBNET>	<NUMA NODE/ZONE>	<BLOCK NODE>	<STORAGE POOL/VOLUME GROUP>	<A OR B>

Para obtener más información sobre las convenciones estándar, las mejores prácticas y las hojas de trabajo de ejemplo rellenas, consulte la ["mejores prácticas"](#) y.. ["Defina los bloques de creación de BeeGFS"](#) Secciones de BeeGFS en Arquitectura verificada de NetApp.

Defina los nodos de archivo y bloque

Configurar nodos de archivos individuales

Especifique la configuración de los nodos de archivos individuales con variables de host (host_var).

Descripción general

Esta sección recorre la relleno de un `host_vars/<FILE_NODE_HOSTNAME>.yml` archivo para cada nodo de archivo del clúster. Estos archivos sólo deben contener una configuración exclusiva de un nodo de archivo concreto. Esto incluye normalmente:

- Definición de la IP o el nombre de host que Ansible debe usar para conectarse al nodo.
- Configurar interfaces adicionales e IP de clúster utilizadas para los servicios de clúster de alta disponibilidad (Pacemaker y Corosync) para comunicarse con otros nodos de archivo. De forma predeterminada, estos servicios utilizan la misma red que la interfaz de gestión, pero deberían estar disponibles interfaces adicionales para la redundancia. La práctica común es definir IP adicionales en la red de almacenamiento, lo que evita la necesidad de un clúster o una red de gestión adicionales.
 - El rendimiento de cualquier red utilizada para la comunicación del clúster no es crítico en cuanto al rendimiento del sistema de archivos. Con la configuración predeterminada del clúster, por lo general una red de al menos 1 Gbps ofrecerá un rendimiento suficiente para las operaciones del clúster, como la sincronización de los estados de nodo y la coordinación de los cambios de estado de los recursos del clúster. Las redes lentas/ocupadas pueden hacer que los cambios en el estado de los recursos tarden más de lo habitual y, en casos extremos, podrían resultar en que los nodos se expulsen del clúster si no pueden enviar latidos en un período de tiempo razonable.

- Configurar las interfaces utilizadas para conectarse a los nodos de bloques sobre el protocolo deseado (por ejemplo, iSCSI/lser, NVMe/IB, NVMe/roce, FCP, etc.)

Pasos

Hacer referencia al esquema de direccionamiento IP definido en "[Planifique el sistema de archivos](#)" para cada nodo de archivo del clúster, cree un archivo `host_vars/<FILE_NODE_HOSTNAME>/yml` y rellénarlo de la siguiente manera:

1. En la parte superior, especifique la IP o el nombre de host que Ansible debe usar a SSH del nodo y gestione:

```
ansible_host: "<MANAGEMENT_IP>"
```

2. Configure las IP adicionales que se puedan usar para el tráfico del clúster:

- a. Si el tipo de red es "[InfiniBand \(uso de IPoIB\)](#)":

```
eseries_ipoib_interfaces:  
- name: <INTERFACE> # Example: ib0 or ilb  
  address: <IP/SUBNET> # Example: 100.127.100.1/16  
- name: <INTERFACE> # Additional interfaces as needed.  
  address: <IP/SUBNET>
```

- b. Si el tipo de red es "[RDMA sobre Ethernet convergente \(roce\)](#)":

```
eseries_roce_interfaces:  
- name: <INTERFACE> # Example: eth0.  
  address: <IP/SUBNET> # Example: 100.127.100.1/16  
- name: <INTERFACE> # Additional interfaces as needed.  
  address: <IP/SUBNET>
```

- c. Si el tipo de red es "[Ethernet \(solo TCP, sin RDMA\)](#)":

```
eseries_ip_interfaces:  
- name: <INTERFACE> # Example: eth0.  
  address: <IP/SUBNET> # Example: 100.127.100.1/16  
- name: <INTERFACE> # Additional interfaces as needed.  
  address: <IP/SUBNET>
```

3. Indique qué IP se deben utilizar para el tráfico del clúster con las IP preferidas más alta:

```
beegfs_ha_cluster_node_ips:
- <MANAGEMENT_IP> # Including the management IP is typically but not
  required.
- <IP_ADDRESS> # Ex: 100.127.100.1
- <IP_ADDRESS> # Additional IPs as needed.
```



Los IPS configurados en el paso dos no se utilizarán como IP de clúster a menos que estén incluidos en el `beegfs_ha_cluster_node_ips` lista. Esto le permite configurar IP/interfaces adicionales con Ansible que pueden utilizarse para otros fines si así lo desea.

4. Si el nodo de archivo tiene que comunicarse con los nodos de bloque a través de un protocolo basado en IP, se deberán configurar las IP en la interfaz adecuada y con todos los paquetes necesarios para instalar y configurar ese protocolo.

- a. Si se utiliza "ISCSI":

```
eseries_iscsi_interfaces:
- name: <INTERFACE> # Example: eth0.
  address: <IP/SUBNET> # Example: 100.127.100.1/16
```

- b. Si se utiliza "Iser":

```
eseries_ib_iser_interfaces:
- name: <INTERFACE> # Example: ib0.
  address: <IP/SUBNET> # Example: 100.127.100.1/16
  configure: true # If the file node is directly connected to the
  block node set to true to setup OpenSM.
```

- c. Si se utiliza "NVMe/IB":

```
eseries_nvme_ib_interfaces:
- name: <INTERFACE> # Example: ib0.
  address: <IP/SUBNET> # Example: 100.127.100.1/16
  configure: true # If the file node is directly connected to the
  block node set to true to setup OpenSM.
```

- d. Si se utiliza "NVMe/roce":

```
eseries_nvme_roce_interfaces:
- name: <INTERFACE> # Example: eth0.
  address: <IP/SUBNET> # Example: 100.127.100.1/16
```

e. Otros protocolos:

- i. Si se utiliza "NVMe/FC", no es necesario configurar interfaces individuales. La implementación del clúster BeeGFS detectará automáticamente los requisitos de protocolo e instalará/configurará según sea necesario. Si utiliza una estructura para conectar nodos de archivos y bloques, asegúrese de que los switches se dividen correctamente siguiendo las prácticas recomendadas de NetApp y del proveedor del switch.
- ii. El uso de FCP o SAS no requiere la instalación ni la configuración de software adicional. Si utiliza FCP, asegúrese de que los switches se dividen correctamente a continuación "NetApp" y las prácticas recomendadas de su proveedor del switch.
- iii. No se recomienda el uso de SRP IB en este momento. Utilice NVMe/IB o Iser en función de lo que admita su nodo de bloque E-Series.

Haga clic en ["aquí"](#) para obtener un ejemplo de un archivo de inventario completo que representa un solo nodo de archivo.

Avanzado: Alternar los adaptadores VPI NVIDIA ConnectX entre Ethernet y modo InfiniBand

Los adaptadores NVIDIA ConnectX-Virtual Protocol Interconnect® (VPI) admiten InfiniBand y Ethernet como capa de transporte. El cambio entre modos no se negocia automáticamente y se debe configurar mediante el `mstconfig` herramienta incluida en `mstflint`, un paquete de código abierto que forma parte de ["Herramientas Mellanox Firmare \(MFT\)"](https://docs.nvidia.com/networking/display/MFTV4133/MFT+Supported+Configurations+and+Parameters). El cambio del modo de los adaptadores solo debe realizarse una vez. Esto se puede realizar manualmente o incluso en el inventario de Ansible como parte de las interfaces configuradas con el `eseries-[ib|ib_iser|ipoib|nvme_ib|nvme_roce|roce]_interfaces:` sección del inventario, para que se compruebe/aplique automáticamente.

Por ejemplo, para cambiar una interfaz actual en modo InfiniBand a Ethernet, se puede utilizar para roce:

1. Especifique para cada interfaz que desee configurar `mstconfig` como una asignación (o diccionario) que especifica `LINK_TYPE_P<N>` donde `<N>` Viene determinado por el número de puerto de HCA de la interfaz. La `<N>` el valor se puede determinar ejecutando `grep PCI_SLOT_NAME /sys/class/net/<INTERFACE_NAME>/device/uevent` Y agregando 1 al último número desde el nombre de la ranura PCI y convirtiendo a decimal.
 - a. Por ejemplo dado `PCI_SLOT_NAME=0000:2f:00.2` (`2 + 1` → puerto HCA 3) → `LINK_TYPE_P3: eth:`

```
eseries_roce_interfaces:
- name: <INTERFACE>
  address: <IP/SUBNET>
  mstconfig:
    LINK_TYPE_P3: eth
```

Para obtener información adicional, consulte ["Documentación de la colección de hosts E-Series de NetApp"](#) para el tipo/protocolo de interfaz que utiliza.

Configure nodos de bloques individuales

Especifique la configuración de los nodos de bloque individuales con variables de host (host_var).

Descripción general

Esta sección recorre la relleno de un `host_vars/<BLOCK_NODE_HOSTNAME>.yaml` archivo de cada nodo de bloque del clúster. Estos archivos sólo deben contener una configuración exclusiva de un nodo de bloque determinado. Esto incluye normalmente:

- El nombre del sistema (como se muestra en System Manager).
- La URL de HTTPS para una de las controladoras (se utiliza para gestionar el sistema mediante su API DE REST).
- Qué nodos de archivo de protocolo de almacenamiento utilizan para conectarse a este nodo de bloque.
- Configurar los puertos de tarjeta de interfaz del host (HIC), como las direcciones IP (si son necesarias).

Pasos

Hacer referencia al esquema de direccionamiento IP definido en ["Planifique el sistema de archivos"](#) para cada nodo de bloque del clúster, cree un archivo `host_vars/<BLOCK_NODE_HOSTNAME>.yaml` y relleno de la siguiente manera:

1. En la parte superior, especifique el nombre del sistema y la URL de HTTPS para una de las controladoras:

```
eseries_system_name: <SYSTEM_NAME>
eseries_system_api_url:
https://<MANAGEMENT_HOSTNAME_OR_IP>:8443/devmgr/v2/
```

2. Seleccione la ["protocolo"](#) los nodos de archivo utilizarán para conectarse a este nodo de bloque:

- a. Protocolos compatibles: auto, iscsi, fc, sas, ib_srp, ib_iser, nvme_ib, nvme_fc, nvme_roce.

```
eseries_initiator_protocol: <PROTOCOL>
```

3. Según el protocolo en uso, los puertos HIC pueden necesitar una configuración adicional. Si es necesario, se debe definir la configuración de puertos de HIC para que la entrada superior de la configuración de cada controladora se corresponda con el puerto físico más a la izquierda de cada controladora y el puerto inferior al puerto que se encuentra en el extremo derecho. Todos los puertos requieren una configuración válida incluso si no están en uso actualmente.



Consulte también la siguiente sección si utiliza InfiniBand HDR (200 GB) o roce de 200 GB con nodos de bloque EF600.

- a. Para iSCSI:

```

eseries_controller_iscsi_port:
  controller_a:          # Ordered list of controller A channel
definition.
  - state:              # Whether the port should be enabled.
Choices: enabled, disabled
  config_method:       # Port configuration method Choices: static,
dhcp
  address:             # Port IPv4 address
  gateway:            # Port IPv4 gateway
  subnet_mask:        # Port IPv4 subnet_mask
  mtu:                # Port IPv4 mtu
  - (...)             # Additional ports as needed.
  controller_b:       # Ordered list of controller B channel
definition.
  - (...)             # Same as controller A but for controller B

# Alternatively the following common port configuration can be
defined for all ports and omitted above:
eseries_controller_iscsi_port_state: enabled          # Generally
specifies whether a controller port definition should be applied
Choices: enabled, disabled
eseries_controller_iscsi_port_config_method: dhcp    # General port
configuration method definition for both controllers. Choices:
static, dhcp
eseries_controller_iscsi_port_gateway:                # General port
IPv4 gateway for both controllers.
eseries_controller_iscsi_port_subnet_mask:           # General port
IPv4 subnet mask for both controllers.
eseries_controller_iscsi_port_mtu: 9000             # General port
maximum transfer units (MTU) for both controllers. Any value greater
than 1500 (bytes).

```

b. Para Iser:

```

eseries_controller_ib_iser_port:
  controller_a:        # Ordered list of controller A channel address
definition.
  -                   # Port IPv4 address for channel 1
  - (...)             # So on and so forth
  controller_b:       # Ordered list of controller B channel address
definition.

```

c. Para NVMe/IB:

```

eseries_controller_nvme_ib_port:
  controller_a:      # Ordered list of controller A channel address
definition.
  -                 # Port IPv4 address for channel 1
  - (...)           # So on and so forth
  controller_b:     # Ordered list of controller B channel address
definition.

```

d. Para NVMe/roce:

```

eseries_controller_nvme_roce_port:
  controller_a:      # Ordered list of controller A channel
definition.
  - state:           # Whether the port should be enabled.
  config_method:    # Port configuration method Choices: static,
dhcp
  address:           # Port IPv4 address
  subnet_mask:      # Port IPv4 subnet_mask
  gateway:           # Port IPv4 gateway
  mtu:               # Port IPv4 mtu
  speed:             # Port IPv4 speed
  controller_b:     # Ordered list of controller B channel
definition.
  - (...)           # Same as controller A but for controller B

# Alternatively the following common port configuration can be
defined for all ports and omitted above:
eseries_controller_nvme_roce_port_state: enabled          # Generally
specifies whether a controller port definition should be applied
Choices: enabled, disabled
eseries_controller_nvme_roce_port_config_method: dhcp     # General
port configuration method definition for both controllers. Choices:
static, dhcp
eseries_controller_nvme_roce_port_gateway:                # General
port IPv4 gateway for both controllers.
eseries_controller_nvme_roce_port_subnet_mask:           # General
port IPv4 subnet mask for both controllers.
eseries_controller_nvme_roce_port_mtu: 4200              # General
port maximum transfer units (MTU). Any value greater than 1500
(bytes).
eseries_controller_nvme_roce_port_speed: auto            # General
interface speed. Value must be a supported speed or auto for
automatically negotiating the speed with the port.

```

e. Los protocolos FC y SAS no requieren configuración adicional. SRP no se recomienda correctamente.

Para obtener opciones adicionales para configurar los puertos HIC y los protocolos de host, incluida la capacidad de configurar CHAP iSCSI, consulte la "[documentación](#)" Incluido con la colección SANtricity. Tenga en cuenta que al implementar BeeGFS, el pool de almacenamiento, la configuración de volumen y otros aspectos del aprovisionamiento del almacenamiento se configurarán en otra parte y no se deberán definir en este archivo.

Haga clic en "[aquí](#)" para obtener un ejemplo de un archivo de inventario completo que representa un solo nodo de bloque.

Mediante InfiniBand HDR (200 GB) o roce de 200 GB con nodos de bloque de EF600 de NetApp:

Para utilizar InfiniBand HDR (200 GB) con EF600, se debe configurar una segunda IP "virtual" para cada puerto físico. Este es un ejemplo de la forma correcta de configurar una cabina EF600 equipada con la tarjeta InfiniBand HDR de doble puerto:

```
eseries_controller_nvme_ib_port:
  controller_a:
    - 192.168.1.101 # Port 2a (physical)
    - 192.168.2.101 # Port 2a (virtual)
    - 192.168.1.100 # Port 2b (physical)
    - 192.168.2.100 # Port 2b (virtual)
  controller_b:
    - 192.168.3.101 # Port 2a (physical)
    - 192.168.4.101 # Port 2a (virtual)
    - 192.168.3.100 # Port 2b (physical)
    - 192.168.4.100 # Port 2b (virtual)
```

Especifique la configuración de nodos de archivos comunes

Especifique la configuración de nodos de archivos comunes mediante variables de grupo (group_var).

Descripción general

La configuración que debería Apple para todos los nodos de archivo se define en group_vars/ha_cluster.yml. Normalmente incluye:

- Información detallada sobre cómo conectarse e iniciar sesión en cada nodo de archivo.
- Configuración de red común.
- Si se permiten reinicios automáticos.
- Cómo deben configurarse los estados de firewall y selinux.
- Configuración de clústeres, incluidas las alertas y las cercas.
- Ajuste del rendimiento.
- Configuración de servicio de BeeGFS común.



Las opciones establecidas en este archivo también pueden definirse en nodos de archivos individuales; por ejemplo, si se están usando modelos de hardware mixtos o tiene contraseñas diferentes para cada nodo. La configuración en nodos de archivo individuales tendrá prioridad sobre la configuración de este archivo.

Pasos

Cree el archivo `group_vars/ha_cluster.yml` y relleno de la siguiente manera:

1. Indique cómo debe autenticarse el nodo de Ansible Control con los hosts remotos:

```
ansible_ssh_user: root
ansible_become_password: <PASSWORD>
```



Especialmente en entornos de producción, no almacene contraseñas en texto sin formato. En su lugar, use Ansible Vault (consulte "[Cifrado de contenido con Ansible Vault](#)") o el `--ask-become-pass` al ejecutar el libro de estrategia. Si la `ansible_ssh_user` es ya el usuario raíz, puede omitir de forma opcional el `ansible_become_password`.

2. Si está configurando IP estáticas en interfaces ethernet o InfiniBand (por ejemplo, IP de clúster) y varias interfaces se encuentran en la misma subred IP (por ejemplo, si `ib0` está usando `192.168.1.10/24` e `ib1` está usando `192.168.1.11/24`), Para que el soporte multihost funcione correctamente, se deben configurar reglas y tablas de enrutamiento IP adicionales. Sólo tiene que activar el enlace de configuración de la interfaz de red proporcionado de la siguiente forma:

```
eseries_ip_default_hook_templates:
- 99-multihoming.j2
```

3. Al poner en marcha el clúster, según el protocolo de almacenamiento, puede que sean necesarios el reinicio de los nodos para facilitar la detección de dispositivos de bloques remotos (volúmenes de E-Series) o aplicar otros aspectos de la configuración. De forma predeterminada, los nodos se preguntará antes de reiniciar, pero puede permitir que los nodos se reinicien automáticamente especificando lo siguiente:

```
eseries_common_allow_host_reboot: true
```

- a. De forma predeterminada, después de un reinicio, para asegurarse de que los dispositivos de bloque y otros servicios estén listos, Ansible esperará hasta el sistema `default.target` se alcanza antes de continuar con la implementación. En algunos casos, cuando se utiliza NVMe/IB, es posible que este no sea el tiempo suficiente para inicializar, detectar y conectarse a dispositivos remotos. Esto puede provocar que la implementación automatizada continúe prematuramente y falle. Para evitar esto cuando se usa NVMe/IB, también se debe definir lo siguiente:

```
eseries_common_reboot_test_command: "! systemctl status
eseries_nvme_ib.service || systemctl --state=exited | grep
eseries_nvme_ib.service"
```

4. Para comunicarse con los servicios de clúster de BeeGFS y ha se necesitan varios puertos de firewall. A menos que desee configurar manualmente el firewall (no se recomienda), especifique lo siguiente para crear zonas de firewall necesarias y abrir puertos automáticamente:

```
beegfs_ha_firewall_configure: True
```

5. En este momento, SELinux no es compatible y se recomienda que el estado se configure como desactivado para evitar conflictos (especialmente cuando RDMA está en uso). Establezca lo siguiente para asegurarse de que SELinux esté desactivado:

```
eseries_beegfs_ha_disable_selinux: True
eseries_selinux_state: disabled
```

6. Configure la autenticación de modo que los nodos de archivo puedan comunicarse, ajustando los valores predeterminados según sea necesario según las directivas de su organización:

```
beegfs_ha_cluster_name: hacluster # BeeGFS HA cluster
name.
beegfs_ha_cluster_username: hacluster # BeeGFS HA cluster
username.
beegfs_ha_cluster_password: hapassword # BeeGFS HA cluster
username's password.
beegfs_ha_cluster_password_sha512_salt: randomSalt # BeeGFS HA cluster
username's password salt.
```

7. Según la "[Planifique el sistema de archivos](#)" Especifique la IP de administración de BeeGFS para este sistema de archivos:

```
beegfs_ha_mgmtd_floating_ip: <IP ADDRESS>
```



Aunque aparentemente redundante, `beegfs_ha_mgmtd_floating_ip` Es importante cuando escala el sistema de archivos BeeGFS más allá de un único clúster de alta disponibilidad. Los clústeres de alta disponibilidad posteriores se ponen en marcha sin un servicio de gestión de BeeGFS adicional y se señalan en el servicio de gestión proporcionado por el primer clúster.

8. Habilite las alertas de correo electrónico si lo desea:

```

beegfs_ha_enable_alerts: True
# E-mail recipient list for notifications when BeeGFS HA resources
change or fail.
beegfs_ha_alert_email_list: ["<EMAIL>"]
# This dictionary is used to configure postfix service
(/etc/postfix/main.cf) which is required to set email alerts.
beegfs_ha_alert_conf_ha_group_options:
    # This parameter specifies the local internet domain name. This is
optional when the cluster nodes have fully qualified hostnames (i.e.
host.example.com)
    mydomain: <MY_DOMAIN>
beegfs_ha_alert_verbosity: 3
# 1) high-level node activity
# 3) high-level node activity + fencing action information + resources
(filter on X-monitor)
# 5) high-level node activity + fencing action information + resources

```

9. Se recomienda encarecidamente habilitar la delimitación; de lo contrario, se puede bloquear que los servicios se inicien en nodos secundarios cuando se produzca un error en el nodo principal.

a. Active la delimitación de forma global especificando lo siguiente:

```

beegfs_ha_cluster_crm_config_options:
    stonith-enabled: True

```

i. Nota: Cualquier soporte "[propiedad del clúster](#)" también se puede especificar aquí si es necesario. Normalmente no es necesario ajustar estos ajustes, ya que el rol de ha de BeeGFS se incluye con una serie de pruebas bien probadas "[valores predeterminados](#)".

b. A continuación, seleccione y configure un agente de cercado:

i. OPCIÓN 1: Para habilitar la cercado mediante unidades de distribución de energía (PDU) APC:

```

beegfs_ha_fencing_agents:
    fence_apc:
        - ipaddr: <PDU_IP_ADDRESS>
          login: <PDU_USERNAME>
          passwd: <PDU_PASSWORD>
          pcmk_host_map:
            "<HOSTNAME>:<PDU_PORT>,<PDU_PORT>;<HOSTNAME>:<PDU_PORT>,<PDU_PORT>"

```

ii. OPCIÓN 2: Para habilitar la esgrima mediante las API Redfish proporcionadas por Lenovo XCC (y otros BMCs):

```

redfish: &redfish
  username: <BMC_USERNAME>
  password: <BMC_PASSWORD>
  ssl_insecure: 1 # If a valid SSL certificate is not available
specify "1".

beegfs_ha_fencing_agents:
  fence_redfish:
    - pcmk_host_list: <HOSTNAME>
      ip: <BMC_IP>
      <<: *redfish
    - pcmk_host_list: <HOSTNAME>
      ip: <BMC_IP>
      <<: *redfish

```

iii. Para obtener más información sobre la configuración de otros agentes de cercado, consulte ["Documentación de redhat"](#).

10. El rol de ha de BeeGFS puede aplicar muchos parámetros de ajuste diferentes para ayudar a optimizar aún más el rendimiento. Entre ellos se incluyen la optimización de la utilización de la memoria del núcleo y la E/S del dispositivo en bloque, entre otros parámetros. La función se entrega con un conjunto razonable de ["valores predeterminados"](#) En función de la prueba con nodos de bloques E-Series de NetApp, pero de forma predeterminada, no se aplican a menos que se especifique:

```
beegfs_ha_enable_performance_tuning: True
```

- a. Si es necesario, también especifique aquí cualquier cambio en el ajuste del rendimiento predeterminado. Vea el completo ["parámetros de ajuste del rendimiento"](#) para obtener detalles adicionales.
11. Para garantizar que las direcciones IP flotantes (a veces conocidas como interfaces lógicas) utilizadas para los servicios BeeGFS puedan conmutar por error entre nodos de archivos, todas las interfaces de red deben tener un nombre coherente. De forma predeterminada, el kernel genera nombres de interfaz de red, lo cual no garantiza la generación de nombres coherentes, incluso en modelos de servidor idénticos con adaptadores de red instalados en las mismas ranuras PCIe. Esto también es útil cuando se crean inventarios antes de que el equipo se despliegue y se conozcan los nombres de las interfaces generadas. Para garantizar nombres de dispositivos coherentes, basados en un diagrama de bloque del servidor o. `lshw -class network -businfo` Output, especifique la asignación de dirección PCIe a interfaz lógica deseada del siguiente modo:

- a. Para interfaces de red InfiniBand (IPoIB):

```

eseries_ipoib_udev_rules:
  "<PCIe ADDRESS>": <NAME> # Ex: 0000:41:00.0: i1a

```

- b. Para interfaces de red Ethernet:


```
eseries_ip_udev_rules:
  "<PCIe ADDRESS>": <NAME> # Ex: 0000:41:00.0: e1a
```



Para evitar conflictos cuando se cambia el nombre de las interfaces (evitando que se le cambie el nombre), no debe utilizar ningún nombre predeterminado potencial como eth0, ens9f0, ib0 o ibs4f0. Una convención de nomenclatura común consiste en usar "e" o "i" para Ethernet o InfiniBand, seguido del número de ranura PCIe y una letra para indicar el puerto. Por ejemplo, el segundo puerto de un adaptador InfiniBand instalado en la ranura 3 sería: I3b.



Si va a utilizar un modelo de nodo de archivos verificado, haga clic en ["aquí"](#) Asignaciones de puerto lógico a dirección PCIe de ejemplo.

12. Opcionalmente, especifique la configuración que debe aplicarse a todos los servicios de BeeGFS del clúster. Se pueden encontrar los valores de configuración predeterminados ["aquí"](#), y la configuración por servicio se especifica en otra parte:

- a. Servicio de gestión de BeeGFS:

```
beegfs_ha_beegfs_mgmt_d_conf_ha_group_options:
  <OPTION>: <VALUE>
```

- b. Servicios de metadatos BeeGFS:

```
beegfs_ha_beegfs_meta_conf_ha_group_options:
  <OPTION>: <VALUE>
```

- c. Servicios de almacenamiento de BeeGFS:

```
beegfs_ha_beegfs_storage_conf_ha_group_options:
  <OPTION>: <VALUE>
```

13. A partir de BeeGFS 7.2.7 y 7.3.1 ["autenticación de conexión"](#) se debe configurar o deshabilitar explícitamente. Hay algunas formas de configurar esto con la puesta en marcha basada en Ansible:

- a. De forma predeterminada, la implementación configurará automáticamente la autenticación de conexión y generará un `connauthfile` Se distribuirá a todos los nodos de archivos y se utilizará con los servicios BeeGFS. Este archivo también se colocará/mantendrá en el nodo de control Ansible en `<INVENTORY>/files/beegfs/<sysMgmtHost>_connAuthFile` donde se debe mantener (de forma segura) para reutilizarlo con clientes que necesiten acceder a este sistema de archivos.
 - i. Para generar una nueva clave, especifique `-e "beegfs_ha_conn_auth_force_new=True` Al ejecutar el libro de estrategia de Ansible. Nota esto se ignora si un `beegfs_ha_conn_auth_secret` está definido.
 - ii. Para obtener información sobre las opciones avanzadas, consulte la lista completa de valores predeterminados que se incluyen con la ["Rol de BeeGFS ha"](#).

b. Se puede utilizar un secreto personalizado definiendo lo siguiente en `ha_cluster.yml`:

```
beegfs_ha_conn_auth_secret: <SECRET>
```

c. La autenticación de conexión se puede deshabilitar completamente (NO se recomienda):

```
beegfs_ha_conn_auth_enabled: false
```

Haga clic en ["aquí"](#) para obtener un ejemplo de un archivo de inventario completo que representa la configuración común de nodos de archivos.

Usar InfiniBand HDR (200 GB) con nodos de bloque de EF600 de NetApp:

Para utilizar InfiniBand HDR (200 GB) con EF600, el administrador de subredes debe admitir la virtualización. Si los nodos de archivos y bloques se conectan mediante un switch, deberá habilitarse en el administrador de subredes de la estructura general.

Si los nodos de bloque y archivo se conectan directamente mediante InfiniBand, una instancia de `opensm` debe configurarse en cada nodo de archivo para cada interfaz conectada directamente a un nodo de bloque. Esto se realiza especificando `configure: true` cuando ["configurar las interfaces de almacenamiento del nodo de archivo"](#).

Actualmente, la versión de la bandeja de entrada de `opensm` El suministro con distribuciones de Linux compatibles no admite la virtualización. En su lugar, es necesario instalar y configurar la versión de `opensm` De Mellanox OpenFabrics Enterprise Distribution (OFED). A pesar de que todavía se admite la puesta en marcha con Ansible, se requieren algunos pasos adicionales:

1. Con `curl` o la herramienta que desee, descargue los paquetes para la versión de OpenSM que se indica en el ["requisitos tecnológicos"](#) Sección del sitio web de Mellanox al `<INVENTORY>/packages/` directorio. Por ejemplo:

```
curl -o packages/opensm-libs-5.9.0.MLNX20210617.c9f2ade-0.1.54103.x86_64.rpm https://linux.mellanox.com/public/repo/mlnx_ofed/5.4-1.0.3.0/rhel8.4/x86_64/opensm-libs-5.9.0.MLNX20210617.c9f2ade-0.1.54103.x86_64.rpm

curl -o packages/opensm-5.9.0.MLNX20210617.c9f2ade-0.1.54103.x86_64.rpm https://linux.mellanox.com/public/repo/mlnx_ofed/5.4-1.0.3.0/rhel8.4/x86_64/opensm-5.9.0.MLNX20210617.c9f2ade-0.1.54103.x86_64.rpm
```

2. Inferior `group_vars/ha_cluster.yml` defina la siguiente configuración:

```

### OpenSM package and configuration information
eseries_ib_opensm_allow_upgrades: true
eseries_ib_opensm_skip_package_validation: true
eseries_ib_opensm_rhel_packages: []
eseries_ib_opensm_custom_packages:
  install:
    - files:
      add:
        "packages/opensm-libs-5.9.0.MLNX20210617.c9f2ade-
0.1.54103.x86_64.rpm": "/tmp/"
        "packages/opensm-5.9.0.MLNX20210617.c9f2ade-
0.1.54103.x86_64.rpm": "/tmp/"
    - packages:
      add:
        - /tmp/opensm-5.9.0.MLNX20210617.c9f2ade-0.1.54103.x86_64.rpm
        - /tmp/opensm-libs-5.9.0.MLNX20210617.c9f2ade-
0.1.54103.x86_64.rpm
  uninstall:
    - packages:
      remove:
        - opensm
        - opensm-libs
      files:
      remove:
        - /tmp/opensm-5.9.0.MLNX20210617.c9f2ade-0.1.54103.x86_64.rpm
        - /tmp/opensm-libs-5.9.0.MLNX20210617.c9f2ade-
0.1.54103.x86_64.rpm

eseries_ib_opensm_options:
  virt_enabled: "2"

```

Especifique la configuración de nodo de bloque común

Especifique la configuración de nodos de bloque común con las variables de grupo (group_var).

Descripción general

La configuración que debería Apple a todos los nodos de bloque se define en group_vars/eseries_storage_systems.yml. Normalmente incluye:

- Detalles sobre cómo el nodo de control Ansible debe conectarse a los sistemas de almacenamiento E-Series que se utilizan como nodos de bloques.
- Las versiones de firmware, NVSRAM y de unidad que deben ejecutar los nodos.
- Configuración global, que incluye la configuración de caché, la configuración de hosts y la configuración

de cómo deben provisionarse los volúmenes.



Las opciones establecidas en este archivo también pueden definirse en nodos de bloques individuales; por ejemplo, si se están utilizando modelos de hardware mixtos o tiene contraseñas diferentes para cada nodo. La configuración en nodos de bloque individuales tendrá prioridad sobre la configuración de este archivo.

Pasos

Cree el archivo `group_vars/eseries_storage_systems.yml` y relleno de la siguiente manera:

1. Ansible no utiliza SSH para conectarse a los nodos de bloques y, en su lugar, utiliza API DE REST. Para lograrlo, debemos establecer:

```
ansible_connection: local
```

2. Especifique el nombre de usuario y la contraseña para gestionar cada nodo. El nombre de usuario puede omitirse opcionalmente (y, de forma predeterminada, `admin`), si no es posible especificar cualquier cuenta con privilegios de administrador. Especifique también si los certificados SSL deben verificarse o ignorarse:

```
eseries_system_username: admin
eseries_system_password: <PASSWORD>
eseries_validate_certs: false
```



No se recomienda enumerar las contraseñas en texto sin formato. Use el almacén de Ansible o proporcione el `eseries_system_password` Al ejecutar Ansible con distribuidores de valor añadido de `--extra`.

3. Opcionalmente, especifique qué firmware de la controladora, NVSRAM y firmware de la unidad se debe instalar en los nodos. Deberá descargarse en el `packages/` directorio antes de ejecutar Ansible. El firmware de la controladora E-Series y NVSRAM se pueden descargar ["aquí"](#) y el firmware de la unidad ["aquí"](#):

```
eseries_firmware_firmware: "packages/<FILENAME>.dlp" # Ex.
"packages/RCB_11.70.2_6000_61b1131d.dlp"
eseries_firmware_nvram: "packages/<FILENAME>.dlp" # Ex.
"packages/N6000-872834-D06.dlp"
eseries_drive_firmware_firmware_list:
  - "packages/<FILENAME>.dlp"
  # Additional firmware versions as needed.
eseries_drive_firmware_upgrade_drives_online: true # Recommended unless
BeeGFS hasn't been deployed yet, as it will disrupt host access if set
to "false".
```



Si se especifica esta configuración, Ansible actualizará automáticamente todo el firmware, incluido el reinicio de las controladoras (si es necesario) sin ningún aviso adicional. Se espera que esto no sea disruptivo para la I/O del host de BeeGFS, pero podría provocar un descenso temporal del rendimiento.

4. Ajuste los valores predeterminados de configuración global del sistema. BeeGFS en NetApp suele recomendar las opciones y valores que se incluyen en esta lista, pero se pueden ajustar en caso necesario:

```
eseries_system_cache_block_size: 32768
eseries_system_cache_flush_threshold: 80
eseries_system_default_host_type: linux dm-mp
eseries_system_autoload_balance: disabled
eseries_system_host_connectivity_reporting: disabled
eseries_system_controller_shelf_id: 99 # Required by default.
```

5. Configure las opciones predeterminadas de aprovisionamiento de volúmenes globales. BeeGFS en NetApp suele recomendar las opciones y valores que se incluyen en esta lista, pero se pueden ajustar en caso necesario:

```
eseries_volume_size_unit: pct # Required by default. This allows volume
capacities to be specified as a percentage, simplifying putting together
the inventory.
eseries_volume_read_cache_enable: true
eseries_volume_read_ahead_enable: false
eseries_volume_write_cache_enable: true
eseries_volume_write_cache_mirror_enable: true
eseries_volume_cache_without_batteries: false
```

6. Si es necesario, ajuste el orden en el que Ansible seleccionará las unidades para los pools de almacenamiento y los grupos de volúmenes, teniendo en cuenta las siguientes prácticas recomendadas:
 - a. Enumere cualquier unidad (potencialmente menor) que se deben usar para los volúmenes de metadatos o gestión primero, y los volúmenes de almacenamiento en último lugar.
 - b. Asegúrese de equilibrar el orden de selección de las unidades en los canales de unidad disponibles según los modelos de bandeja de discos/compartimento de unidades. Por ejemplo, con EF600 y sin expansiones, las unidades 0-11 están en el canal de unidades 1 y las unidades 12-23 están en el canal de unidades. Por lo tanto, una estrategia para equilibrar la selección de conducción es seleccionar `disk shelf:drive 99:0, 99:23, 99:1, 99:22, etc.` En el caso de que haya más de un compartimento, el primer dígito representa el ID de bandeja de unidades.

```
# Optimal/recommended order for the EF600 (no expansion):
eseries_storage_pool_usable_drives:
"99:0,99:23,99:1,99:22,99:2,99:21,99:3,99:20,99:4,99:19,99:5,99:18,99
:6,99:17,99:7,99:16,99:8,99:15,99:9,99:14,99:10,99:13,99:11,99:12"
```

Haga clic en ["aquí"](#) para obtener un ejemplo de un archivo de inventario completo que representa la configuración común de nodos de bloques.

Defina los servicios BeeGFS

Defina el servicio de gestión de BeeGFS

Los servicios BeeGFS se configuran mediante variables de grupo (Group_var).

Descripción general

En esta sección se describe la definición del servicio de gestión de BeeGFS. En los clústeres de alta disponibilidad solo debe haber un servicio de este tipo para un sistema de archivos concreto. La configuración de este servicio incluye la definición:

- El tipo de servicio (gestión).
- Definir cualquier configuración que sólo se debe aplicar a este servicio BeeGFS.
- Configuración de una o varias IP flotantes (interfaces lógicas) en las que se puede acceder a este servicio.
- Especificar dónde y cómo debe almacenar un volumen datos para este servicio (el objetivo de gestión de BeeGFS).

Pasos

Cree un archivo nuevo `group_vars/mgmt.yml` y haciendo referencia a la ["Planifique el sistema de archivos"](#) sección rellenarla de la siguiente forma:

1. Indique que este archivo representa la configuración de un servicio de administración de BeeGFS:

```
beegfs_service: management
```

2. Defina cualquier configuración que se deba aplicar sólo a este servicio BeeGFS. Esto no suele ser necesario para el servicio de gestión a menos que necesite habilitar cuotas, sin embargo, con cualquier parámetro de configuración admitido de `beegfs-mgmt.conf` se puede incluir. Nota los siguientes parámetros se configuran automáticamente u otros lugares y no se deben especificar aquí: `storeMgmtDirectory`, `connAuthFile`, `connDisableAuthentication`, `connInterfacesFile`, y `connNetFilterFile`.

```
beegfs_ha_beegfs_mgmt_conf_resource_group_options:  
  <beegfs-mgmt.conf:key>:<beegfs-mgmt.conf:value>
```

3. Configure uno o varios IP flotantes que utilizarán otros servicios y clientes para conectarse a este servicio (esto establecerá automáticamente BeeGFS `connInterfacesFile` opción):

```
floating_ips:
  - <INTERFACE>:<IP/SUBNET> # Primary interface. Ex.
i1b:100.127.101.0/16
  - <INTERFACE>:<IP/SUBNET> # Secondary interface(s) as needed.
```

4. Opcionalmente, especifique una o varias subredes IP permitidas que se pueden utilizar para la comunicación saliente (esto establecerá automáticamente BeeGFS `connNetFilterFile` opción):

```
filter_ip_ranges:
  - <SUBNET>/<MASK> # Ex. 192.168.10.0/24
```

5. Especifique el objetivo de gestión de BeeGFS en el que este servicio almacenará datos de acuerdo con las siguientes directrices:
- Se puede utilizar el mismo nombre de pool de almacenamiento o grupo de volúmenes para varios servicios/objetivos de BeeGFS; asegúrese de utilizar el mismo `name`, `raid_level`, `criteria_*`, y `common_*` la configuración de cada uno (los volúmenes enumerados para cada servicio deben ser diferentes).
 - Los tamaños de los volúmenes se deben especificar como un porcentaje del pool de almacenamiento/grupo de volúmenes y el total no debe ser superior a 100 en todos los servicios/volúmenes que utilizan un pool de almacenamiento/grupo de volúmenes en particular. Nota cuando se usan SSD, se recomienda dejar espacio libre en el grupo de volúmenes para maximizar el rendimiento de SSD y la vida útil (haga clic en ["aquí"](#) para obtener más detalles).
 - Haga clic en ["aquí"](#) para obtener una lista completa de las opciones de configuración disponibles para `eseries_storage_pool_configuration`. Tenga en cuenta algunas opciones como `state`, `host`, `host_type`, `workload_name`, y `workload_metadata` y los nombres de volúmenes se generan automáticamente y no se deben especificar aquí.

```
beegfs_targets:
  <BLOCK_NODE>: # The name of the block node as found in the Ansible
inventory. Ex: ictad22a01
  eseries_storage_pool_configuration:
    - name: <NAME> # Ex: beegfs_m1_m2_m5_m6
      raid_level: <LEVEL> # One of: raid1, raid5, raid6, raidDiskPool
      criteria_drive_count: <DRIVE COUNT> # Ex. 4
      common_volume_configuration:
        segment_size_kb: <SEGMENT SIZE> # Ex. 128
      volumes:
        - size: <PERCENT> # Percent of the pool or volume group to
allocate to this volume. Ex. 1
          owning_controller: <CONTROLLER> # One of: A, B
```

Haga clic en ["aquí"](#) Para obtener un ejemplo de un archivo de inventario completo que representa un servicio de administración de BeeGFS.

Defina el servicio de metadatos BeeGFS

Los servicios BeeGFS se configuran mediante variables de grupo (Group_var).

Descripción general

En esta sección se describe la definición del servicio de metadatos de BeeGFS. Al menos debe haber un servicio de este tipo en los clústeres de alta disponibilidad para un sistema de archivos determinado. La configuración de este servicio incluye la definición:

- El tipo de servicio (metadatos).
- Definir cualquier configuración que sólo se debe aplicar a este servicio BeeGFS.
- Configuración de una o varias IP flotantes (interfaces lógicas) en las que se puede acceder a este servicio.
- Especificar dónde y cómo debe almacenar un volumen datos para este servicio (el objetivo de metadatos BeeGFS).

Pasos

Haciendo referencia a "[Planifique el sistema de archivos](#)" cree un archivo en `group_vars/meta_<ID>.yaml` para cada servicio de metadatos del clúster y rellene los siguientes pasos:

1. Indique que este archivo representa la configuración de un servicio de metadatos BeeGFS:

```
beegfs_service: metadata
```

2. Defina cualquier configuración que se deba aplicar sólo a este servicio BeeGFS. Al mínimo debe especificar el puerto TCP y UDP deseado, sin embargo, cualquier parámetro de configuración compatible desde `beegfs-meta.conf` también se puede incluir. Nota los siguientes parámetros se configuran automáticamente u otros lugares y no se deben especificar aquí: `sysMgmtHost`, `storeMetaDirectory`, `connAuthFile`, `connDisableAuthentication`, `connInterfacesFile`, y `connNetFilterFile`.

```
beegfs_ha_beegfs_meta_conf_resource_group_options:  
  connMetaPortTCP: <TCP PORT>  
  connMetaPortUDP: <UDP PORT>  
  tuneBindToNumaZone: <NUMA ZONE> # Recommended if using file nodes with  
  multiple CPU sockets.
```

3. Configure uno o varios IP flotantes que utilizarán otros servicios y clientes para conectarse a este servicio (esto establecerá automáticamente BeeGFS `connInterfacesFile` opción):

```
floating_ips:  
  - <INTERFACE>:<IP/SUBNET> # Primary interface. Ex.  
  i1b:100.127.101.1/16  
  - <INTERFACE>:<IP/SUBNET> # Secondary interface(s) as needed.
```


4. Opcionalmente, especifique una o varias subredes IP permitidas que se pueden utilizar para la comunicación saliente (esto establecerá automáticamente BeeGFS `connNetFilterFile` opción):

```
filter_ip_ranges:
- <SUBNET>/<MASK> # Ex. 192.168.10.0/24
```

5. Especifique el destino de metadatos BeeGFS en el que este servicio almacenará datos de acuerdo con las siguientes directrices (también configurará automáticamente el `storeMetaDirectory` opción):

- Se puede utilizar el mismo nombre de pool de almacenamiento o grupo de volúmenes para varios servicios/objetivos de BeeGFS; asegúrese de utilizar el mismo `name`, `raid_level`, `criteria_*`, y `common_*` la configuración de cada uno (los volúmenes enumerados para cada servicio deben ser diferentes).
- Los tamaños de los volúmenes se deben especificar como un porcentaje del pool de almacenamiento/grupo de volúmenes y el total no debe ser superior a 100 en todos los servicios/volúmenes que utilizan un pool de almacenamiento/grupo de volúmenes en particular. Nota cuando se usan SSD, se recomienda dejar espacio libre en el grupo de volúmenes para maximizar el rendimiento de SSD y la vida útil (haga clic en ["aquí"](#) para obtener más detalles).
- Haga clic en ["aquí"](#) para obtener una lista completa de las opciones de configuración disponibles para `eseries_storage_pool_configuration`. Tenga en cuenta algunas opciones como `state`, `host`, `host_type`, `workload_name`, y `workload_metadata` y los nombres de volúmenes se generan automáticamente y no se deben especificar aquí.

```
beegfs_targets:
  <BLOCK_NODE>: # The name of the block node as found in the Ansible
inventory. Ex: ictad22a01
  eseries_storage_pool_configuration:
    - name: <NAME> # Ex: beegfs_m1_m2_m5_m6
      raid_level: <LEVEL> # One of: raid1, raid5, raid6, raidDiskPool
      criteria_drive_count: <DRIVE COUNT> # Ex. 4
      common_volume_configuration:
        segment_size_kb: <SEGMENT SIZE> # Ex. 128
      volumes:
        - size: <PERCENT> # Percent of the pool or volume group to
allocate to this volume. Ex. 1
          owning_controller: <CONTROLLER> # One of: A, B
```

Haga clic en ["aquí"](#) Para obtener un ejemplo de un archivo de inventario completo que representa un servicio de metadatos BeeGFS.

Defina el servicio de almacenamiento BeeGFS

Los servicios BeeGFS se configuran mediante variables de grupo (`Group_var`).

Descripción general

En esta sección se describe la definición del servicio de almacenamiento de BeeGFS. Al menos debe haber

un servicio de este tipo en los clústeres de alta disponibilidad para un sistema de archivos determinado. La configuración de este servicio incluye la definición:

- El tipo de servicio (almacenamiento).
- Definir cualquier configuración que sólo se debe aplicar a este servicio BeeGFS.
- Configuración de una o varias IP flotantes (interfaces lógicas) en las que se puede acceder a este servicio.
- Especificar dónde/cómo deben ser los volúmenes para almacenar datos para este servicio (los destinos de almacenamiento BeeGFS).

Pasos

Haciendo referencia a "[Planifique el sistema de archivos](#)" cree un archivo en `group_vars/stor_<ID>.yaml` para cada servicio de almacenamiento del clúster y rellene los siguientes pasos:

1. Indique este archivo que representa la configuración de un servicio de almacenamiento BeeGFS:

```
beegfs_service: storage
```

2. Defina cualquier configuración que se deba aplicar sólo a este servicio BeeGFS. Al mínimo debe especificar el puerto TCP y UDP deseado, sin embargo, cualquier parámetro de configuración compatible desde `beegfs-storage.conf` también se puede incluir. Nota los siguientes parámetros se configuran automáticamente u otros lugares y no se deben especificar aquí: `sysMgmtHost`, `storeStorageDirectory`, `connAuthFile`, `connDisableAuthentication`, `connInterfacesFile`, y `connNetFilterFile`.

```
beegfs_ha_beegfs_storage_conf_resource_group_options:  
  connStoragePortTCP: <TCP PORT>  
  connStoragePortUDP: <UDP PORT>  
  tuneBindToNumaZone: <NUMA ZONE> # Recommended if using file nodes with  
  multiple CPU sockets.
```

3. Configure uno o varios IP flotantes que utilizarán otros servicios y clientes para conectarse a este servicio (esto establecerá automáticamente BeeGFS `connInterfacesFile` opción):

```
floating_ips:  
  - <INTERFACE>:<IP/SUBNET> # Primary interface. Ex.  
  i1b:100.127.101.1/16  
  - <INTERFACE>:<IP/SUBNET> # Secondary interface(s) as needed.
```

4. Opcionalmente, especifique una o varias subredes IP permitidas que se pueden utilizar para la comunicación saliente (esto establecerá automáticamente BeeGFS `connNetFilterFile` opción):

```
filter_ip_ranges:  
  - <SUBNET>/<MASK> # Ex. 192.168.10.0/24
```

5. Especifique los objetivos de almacenamiento de BeeGFS en los que este servicio almacenará datos de acuerdo con las siguientes directrices (también configurará automáticamente el `storeStorageDirectory` opción):
- Se puede utilizar el mismo nombre de pool de almacenamiento o grupo de volúmenes para varios servicios/objetivos de BeeGFS; asegúrese de utilizar el mismo `name`, `raid_level`, `criteria_*`, y `common_*` la configuración de cada uno (los volúmenes enumerados para cada servicio deben ser diferentes).
 - Los tamaños de los volúmenes se deben especificar como un porcentaje del pool de almacenamiento/grupo de volúmenes y el total no debe ser superior a 100 en todos los servicios/volúmenes que utilizan un pool de almacenamiento/grupo de volúmenes en particular. Nota cuando se usan SSD, se recomienda dejar espacio libre en el grupo de volúmenes para maximizar el rendimiento de SSD y la vida útil (haga clic en "[aquí](#)" para obtener más detalles).
 - Haga clic en "[aquí](#)" para obtener una lista completa de las opciones de configuración disponibles para `eseries_storage_pool_configuration`. Tenga en cuenta algunas opciones como `state`, `host`, `host_type`, `workload_name`, y `workload_metadata` y los nombres de volúmenes se generan automáticamente y no se deben especificar aquí.

```
beegfs_targets:
  <BLOCK_NODE>: # The name of the block node as found in the Ansible
inventory. Ex: ictad22a01
  eseries_storage_pool_configuration:
    - name: <NAME> # Ex: beegfs_s1_s2
      raid_level: <LEVEL> # One of: raid1, raid5, raid6,
raidDiskPool
      criteria_drive_count: <DRIVE COUNT> # Ex. 4
      common_volume_configuration:
        segment_size_kb: <SEGMENT SIZE> # Ex. 128
      volumes:
        - size: <PERCENT> # Percent of the pool or volume group to
allocate to this volume. Ex. 1
          owning_controller: <CONTROLLER> # One of: A, B
        # Multiple storage targets are supported / typical:
        - size: <PERCENT> # Percent of the pool or volume group to
allocate to this volume. Ex. 1
          owning_controller: <CONTROLLER> # One of: A, B
```

Haga clic en "[aquí](#)" Para obtener un ejemplo de un archivo de inventario completo que representa un servicio de almacenamiento BeeGFS.

Asigne servicios BeeGFS a los nodos de archivo

Especifique qué nodos de archivo pueden ejecutar cada servicio BeeGFS mediante `inventory.yml` archivo.

Descripción general

En esta sección se explica cómo crear la `inventory.yml` archivo. Esto incluye una lista de todos los nodos de bloque y especificar qué nodos de archivo pueden ejecutar cada servicio BeeGFS.

Pasos

Cree el archivo `inventory.yml` y relleno de la siguiente manera:

1. Desde la parte superior del archivo, cree la estructura de inventario estándar de Ansible:

```
# BeeGFS HA (High_Availability) cluster inventory.
all:
  children:
```

2. Cree un grupo que contenga todos los nodos de bloques que participan en este clúster de alta disponibilidad:

```
# Ansible group representing all block nodes:
eseries_storage_systems:
  hosts:
    <BLOCK NODE HOSTNAME>:
    <BLOCK NODE HOSTNAME>:
    # Additional block nodes as needed.
```

3. Cree un grupo que contendrá todos los servicios BeeGFS del clúster y los nodos de archivo que los ejecutarán:

```
# Ansible group representing all file nodes:
ha_cluster:
  children:
```

4. Para cada servicio BeeGFS del clúster, defina los nodos de archivos preferidos y secundarios que deben ejecutar ese servicio:

```
<SERVICE>: # Ex. "mgmt", "meta_01", or "stor_01".
  hosts:
    <FILE NODE HOSTNAME>:
    <FILE NODE HOSTNAME>:
    # Additional file nodes as needed.
```

Haga clic en ["aquí"](#) para obtener un ejemplo de un archivo de inventario completo.

Información de copyright

Copyright © 2024 NetApp, Inc. Todos los derechos reservados. Imprimido en EE. UU. No se puede reproducir este documento protegido por copyright ni parte del mismo de ninguna forma ni por ningún medio (gráfico, electrónico o mecánico, incluidas fotocopias, grabaciones o almacenamiento en un sistema de recuperación electrónico) sin la autorización previa y por escrito del propietario del copyright.

El software derivado del material de NetApp con copyright está sujeto a la siguiente licencia y exención de responsabilidad:

ESTE SOFTWARE LO PROPORCIONA NETAPP «TAL CUAL» Y SIN NINGUNA GARANTÍA EXPRESA O IMPLÍCITA, INCLUYENDO, SIN LIMITAR, LAS GARANTÍAS IMPLÍCITAS DE COMERCIALIZACIÓN O IDONEIDAD PARA UN FIN CONCRETO, CUYA RESPONSABILIDAD QUEDA EXIMIDA POR EL PRESENTE DOCUMENTO. EN NINGÚN CASO NETAPP SERÁ RESPONSABLE DE NINGÚN DAÑO DIRECTO, INDIRECTO, ESPECIAL, EJEMPLAR O RESULTANTE (INCLUYENDO, ENTRE OTROS, LA OBTENCIÓN DE BIENES O SERVICIOS SUSTITUTIVOS, PÉRDIDA DE USO, DE DATOS O DE BENEFICIOS, O INTERRUPCIÓN DE LA ACTIVIDAD EMPRESARIAL) CUALQUIERA SEA EL MODO EN EL QUE SE PRODUJERON Y LA TEORÍA DE RESPONSABILIDAD QUE SE APLIQUE, YA SEA EN CONTRATO, RESPONSABILIDAD OBJETIVA O AGRAVIO (INCLUIDA LA NEGLIGENCIA U OTRO TIPO), QUE SURJAN DE ALGÚN MODO DEL USO DE ESTE SOFTWARE, INCLUSO SI HUBIEREN SIDO ADVERTIDOS DE LA POSIBILIDAD DE TALES DAÑOS.

NetApp se reserva el derecho de modificar cualquiera de los productos aquí descritos en cualquier momento y sin aviso previo. NetApp no asume ningún tipo de responsabilidad que surja del uso de los productos aquí descritos, excepto aquello expresamente acordado por escrito por parte de NetApp. El uso o adquisición de este producto no lleva implícita ninguna licencia con derechos de patente, de marcas comerciales o cualquier otro derecho de propiedad intelectual de NetApp.

Es posible que el producto que se describe en este manual esté protegido por una o más patentes de EE. UU., patentes extranjeras o solicitudes pendientes.

LEYENDA DE DERECHOS LIMITADOS: el uso, la copia o la divulgación por parte del gobierno están sujetos a las restricciones establecidas en el subpárrafo (b)(3) de los derechos de datos técnicos y productos no comerciales de DFARS 252.227-7013 (FEB de 2014) y FAR 52.227-19 (DIC de 2007).

Los datos aquí contenidos pertenecen a un producto comercial o servicio comercial (como se define en FAR 2.101) y son propiedad de NetApp, Inc. Todos los datos técnicos y el software informático de NetApp que se proporcionan en este Acuerdo tienen una naturaleza comercial y se han desarrollado exclusivamente con fondos privados. El Gobierno de EE. UU. tiene una licencia limitada, irrevocable, no exclusiva, no transferible, no sublicenciable y de alcance mundial para utilizar los Datos en relación con el contrato del Gobierno de los Estados Unidos bajo el cual se proporcionaron los Datos. Excepto que aquí se disponga lo contrario, los Datos no se pueden utilizar, desvelar, reproducir, modificar, interpretar o mostrar sin la previa aprobación por escrito de NetApp, Inc. Los derechos de licencia del Gobierno de los Estados Unidos de América y su Departamento de Defensa se limitan a los derechos identificados en la cláusula 252.227-7015(b) de la sección DFARS (FEB de 2014).

Información de la marca comercial

NETAPP, el logotipo de NETAPP y las marcas que constan en <http://www.netapp.com/TM> son marcas comerciales de NetApp, Inc. El resto de nombres de empresa y de producto pueden ser marcas comerciales de sus respectivos propietarios.