部署BeeGFS文件系统

BeeGFS on NetApp with E-Series Storage

NetApp
January 27, 2026

目录

部署BeeGFS文件系统

Ansible攻略手册概述

使用Ansible部署和管理BeeGFS HA集群。

概述

前面几节介绍了构建表示BeeGFS HA集群的Ansible清单所需的步骤。本节介绍由NetApp开发的用于部署和管理集群的Ansible自动化功能。

Ansible：关键概念

在继续操作之前、熟悉几个关键的Ansible概念会很有帮助：

- 根据Ansible清单执行的任务在称为*攻略手册*的内容中进行了定义。
  - Ansible中的大多数任务都设计为*幂等*、这意味着可以多次运行这些任务、以验证所需的配置/状态是否仍然适用、而不会造成中断或进行不必要的更新。
- Ansible中最小的执行单位是*模块*。
  - 典型的攻略手册使用多个模块。
    - 示例：下载软件包、更新配置文件、启动/启用服务。
  - NetApp分发模块以自动执行NetApp E系列系统。
- 更好地将复杂的自动化作为一个角色进行打包。
  - 基本上是分发可重复使用的攻略手册的标准格式。
  - NetApp为Linux主机和BeeGFS文件系统分发角色。

BeeGFS HA Role for Ansible：关键概念

在NetApp上部署和管理每个版本的BeeGFS所需的所有自动化功能均作为Ansible角色打包、并作为的一部分进行分发 "适用于BeeGFS的NetApp E系列Ansible资料集"：

- 可以将此角色视为BeeGFS的*安装程序*和现代*部署/管理*引擎之间的某个位置。
  - 将现代基础架构应用为代码实践和理念、以简化任何规模的存储基础架构管理。
  - 与此"Kubespray"项目允许用户部署/维护整个Kubirnetes分发版以实现横向扩展计算基础架构的方式类似。
- 此角色是NetApp用于打包、分发和维护基于NetApp的BeeGFS解决方案的*软件定义*格式。
  - 无需分发整个Linux分发版或大型映像、即可努力打造"类似设备的"体验。
  - 包括NetApp编写的符合Open Cluster Framework (OCF)的集群资源代理、用于自定义BeeGFS目标、IP地址和监控、从而实现智能Pacemaker/BeeGFS集成。
- 此角色不仅仅是部署"自动化"、还用于管理整个文件系统生命周期、包括：
  - 应用按服务或集群范围的配置更改和更新。

- 解决硬件问题后自动执行集群修复和恢复。
- 通过对BeeGFS和NetApp卷进行广泛测试来设置默认值、简化性能调整。
- 验证并更正配置偏差。

NetApp还为提供了Ansible角色 "BeeGFS客户端"、可选择用于安装BeeGFS并将文件系统挂载到计算/GPU/登录节点。

部署BeeGFS HA集群

使用攻略手册指定部署BeeGFS HA集群时应运行的任务。

概述

本节介绍如何组合用于在NetApp上部署/管理BeeGFS的标准攻略手册。

步骤

创建Ansible攻略手册

创建文件 `playbook.yml` 并按如下所示进行填充：

1. 首先定义一组任务(通常称为 "播放")、并且只能在NetApp E系列块节点上运行。我们会使用暂停任务在运行安装之前进行提示(以避免意外运行攻略手册)、然后导入 `nar_santricity_management` 角色。此角色负责应用中定义的任何常规系统配置 `group_vars/eseries_storage_systems.yml` 或个人 `host_vars/<BLOCK NODE>.yml` 文件。

```
- hosts: eseries_storage_systems
  gather_facts: false
  collections:
    - netapp_eseries.santricity
  tasks:
    - name: Verify before proceeding.
      pause:
        prompt: "Are you ready to proceed with running the BeeGFS HA
role? Depending on the size of the deployment and network performance
between the Ansible control node and BeeGFS file and block nodes this
can take awhile (10+ minutes) to complete."
    - name: Configure NetApp E-Series block nodes.
      import_role:
        name: nar_santricity_management
```

2. 定义对所有文件和块节点运行的播放：

```
- hosts: all
  any_errors_fatal: true
  gather_facts: false
  collections:
    - netapp_eseries.beegfs
```

3. 在此角色中、我们可以选择定义一组"预任务"、这些任务应在部署HA集群之前运行。这对于验证/安装Python等任何前提条件非常有用。我们还可以注入任何飞行前检查、例如验证是否支持提供的Ansible标记：

```
  pre_tasks:
    - name: Ensure a supported version of Python is available on all
file nodes.
      block:
        - name: Check if python is installed.
          failed_when: false
          changed_when: false
          raw: python --version
          register: python_version

        - name: Check if python3 is installed.
          raw: python3 --version
          failed_when: false
          changed_when: false
          register: python3_version
          when: 'python_version["rc"] != 0 or (python_version["stdout"]
| regex_replace("Python ", "")) is not version("3.0", ">=")'

        - name: Install python3 if needed.
          raw: |
            id=$(grep "^ID=" /etc/*release* | cut -d= -f 2 | tr -d '"')
            case $id in
              ubuntu) sudo apt install python3 ;;
              rhel|centos) sudo yum -y install python3 ;;
              sles) sudo zypper install python3 ;;
            esac
          args:
            executable: /bin/bash
          register: python3_install
          when: python_version['rc'] != 0 and python3_version['rc'] != 0
          become: true

        - name: Create a symbolic link to python from python3.
          raw: ln -s /usr/bin/python3 /usr/bin/python
          become: true
```

```
          when: python_version['rc'] != 0
      when: inventory_hostname not in
groups[beegfs_ha_ansible_storage_group]

    - name: Verify any provided tags are supported.
      fail:
        msg: "{{ item }} tag is not a supported BeeGFS HA tag. Rerun
your playbook command with --list-tags to see all valid playbook tags."
      when: 'item not in ["all", "storage", "beegfs_ha",
"beegfs_ha_package", "beegfs_ha_configure",
"beegfs_ha_configure_resource", "beegfs_ha_performance_tuning",
"beegfs_ha_backup", "beegfs_ha_client"]'
      loop: "{{ ansible_run_tags }}"
```

4. 最后、此Play会为您要部署的BeeGFS版本导入BeeGFS HA角色：

```
  tasks:
    - name: Verify the BeeGFS HA cluster is properly deployed.
      import_role:
        name: beegfs_ha_7_4 # Alternatively specify: beegfs_ha_7_3.
```

> 每个受支持的BeeGFS主要版本都会保留一个BeeGFS HA角色。这样、用户可以选择何时升级主要/次要版本。目前(`beegfs_7_3`(`beegfs_7_2`支持BeeGFS 7.3.x或BeeGFS 7.2.x。默认情况下、这两个角色将在发布时部署最新的BeeGFS修补程序版本、但用户可以选择覆盖此修补程序、并根据需要部署最新的修补程序。"升级指南"有关详细信息、请参见最新版本。

5. 可选：如果要定义其他任务、请记住这些任务是否应定向到 `all` 主机(包括E系列存储系统)或仅文件节点。如果需要、可使用定义一个专门针对文件节点的新Play - `hosts: ha_cluster`。

单击 "此处" 有关完整攻略手册文件的示例。

安装NetApp Ansible Collections

Ansible和所有依赖关系的BeeGFS收集将在上进行维护 "Ansible Galax河"。在Ansible控制节点上、运行以下命令以安装最新版本：

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

虽然通常不建议这样做、但也可以安装此集合的特定版本：

```
ansible-galaxy collection install netapp_eseries.beegfs:
==<MAJOR>.<MINOR>.<PATCH>
```

运行攻略手册

位于包含的Ansible控制节点上的目录中 `inventory.yml` 和 `playbook.yml` 文件、请按如下所示运行攻略手册：

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

根据集群的大小、初始部署可能需要20分钟以上的时间。如果部署因任何原因失败、只需更正任何问题(例如布线不当、节点未启动等)、然后重新启动Ansible攻略手册即可。

指定时"通用文件节点配置"，如果选择默认选项让Ansound自动管理基于连接的身份验证，则 `connAuthFile` 可在 `<playbook_dir>/files/beegfs/<sysMgmtdHost>_connAuthFile`(默认情况下)中找到用作共享密钥的。任何需要访问文件系统的客户端都需要使用此共享密钥。如果使用配置客户端，则会自动处理此"BeeGFS客户端角色"问题。

部署BeeGFS客户端

也可以使用Ansible配置BeeGFS客户端并挂载文件系统。

概述

要访问BeeGFS文件系统、需要在需要挂载文件系统的每个节点上安装和配置BeeGFS客户端。本节介绍如何使用可用执行这些任务 "Ansible角色"。

步骤

创建客户端清单文件

1. 如果需要、请从Ansible控制节点向要配置为BeeGFS客户端的每个主机设置无密码SSH：

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

2. 下 `host_vars/` 下、为名为的每个BeeGFS客户端创建一个文件 `<HOSTNAME>.yml` 使用以下内容、在占位符文本中填写适用于您环境的正确信息：

   ```
   # BeeGFS Client
   ansible_host: <MANAGEMENT_IP>
   ```

3. 如果要使用NetApp E系列主机集合的角色配置InfiniBand或以太网接口、以便客户端连接到BeeGFS文件节点、也可以包括以下选项之一：

   a. 网络类型为 "InfiniBand (使用IPoIB)"：

```
eseries_ipoib_interfaces:
- name: <INTERFACE>  # Example: ib0 or i1b
  address: <IP/SUBNET> # Example: 100.127.100.1/16
- name: <INTERFACE>  # Additional interfaces as needed.
  address: <IP/SUBNET>
```

b. 网络类型为 "基于融合以太网的RDMA (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. 网络类型为 "以太网(仅限TCP、无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>
```

4. 创建新文件 `client_inventory.yml` 并指定Ansible应用于连接到每个客户端的用户、Ansible应用于权限升级的密码(这需要 `ansible_ssh_user` 为root用户或具有sudo权限)：

```
# BeeGFS client inventory.
all:
  vars:
    ansible_ssh_user: <USER>
    ansible_become_password: <PASSWORD>
```

> 请勿以纯文本格式存储密码。请改用Ansible Vault (请参见) "Ansible文档" 使用Ansible Vault对内容进行加密)或使用 `--ask-become-pass` 运行攻略手册时的选项。

5. 在中 `client_inventory.yml` 文件中、列出应在下配置为BeeGFS客户端的所有主机 `beegfs_clients` 组、然后参考实时注释、取消注释在系统上构建BeeGFS客户端内核模块所需的任何其他配置：

```
children:
    # Ansible group representing all BeeGFS clients:
    beegfs_clients:
      hosts:
        <CLIENT HOSTNAME>:
        # Additional clients as needed.

      vars:
        # OPTION 1: If you're using the NVIDIA OFED drivers and they are
already installed:
        #eseries_ib_skip: True # Skip installing inbox drivers when
using the IPoIB role.
        #beegfs_client_ofed_enable: True
        #beegfs_client_ofed_include_path:
"/usr/src/ofa_kernel/default/include"

        # OPTION 2: If you're using inbox IB/RDMA drivers and they are
already installed:
        #eseries_ib_skip: True # Skip installing inbox drivers when
using the IPoIB role.

        # OPTION 3: If you want to use inbox IB/RDMA drivers and need
them installed/configured.
        #eseries_ib_skip: False # Default value.
        #beegfs_client_ofed_enable: False # Default value.
```

使用NVIDIA OFED驱动程序时、请确保beegfs_client_OFED_include_path指向适用于您的Linux安装的正确"header include path"。有关详细信息，请参见的BeeGFS文档 "RDMA支持"。

6. 在中 `client_inventory.yml` 文件中、列出要在先前定义的任何下挂载的BeeGFS文件系统 `vars`：

```
        beegfs_client_mounts:
          - sysMgmtdHost: <IP ADDRESS>  # Primary IP of the BeeGFS
management service.
            mount_point: /mnt/beegfs    # Path to mount BeeGFS on the
client.
            connInterfaces:
              - <INTERFACE> # Example: ibs4f1
              - <INTERFACE>
            beegfs_client_config:
              # Maximum number of simultaneous connections to the same
node.
              connMaxInternodeNum: 128 # BeeGFS Client Default: 12
              # Allocates the number of buffers for transferring IO.
              connRDMABufNum: 36 # BeeGFS Client Default: 70
              # Size of each allocated RDMA buffer
              connRDMABufSize: 65536 # BeeGFS Client Default: 8192
              # Required when using the BeeGFS client with the shared-
disk HA solution.
              # This does require BeeGFS targets be mounted in the
default "sync" mode.
              # See the documentation included with the BeeGFS client
role for full details.
              sysSessionChecksEnabled: false
        # Specify additional file system mounts for this or other file
systems.
```

7. 自BeeGFS 7.2.7和7.3.1起"连接身份验证"、必须配置或显式禁用。根据您在指定时选择配置基于连接的身份验证"通用文件节点配置"的方式，您可能需要调整客户端配置：

   a. 默认情况下、HA集群部署将自动配置连接身份验证并生成 `connauthfile` 该位置将放置/维护在位于的Ansible控制节点上 `<INVENTORY>/files/beegfs/<sysMgmtdHost>_connAuthFile`。默认情况下、BeeGFS客户端角色设置为读取此文件或将其分发到中定义的客户端 `client_inventory.yml`、不需要执行其他操作。

      i. 有关高级选项、请参阅随附的默认值完整列表 "BeeGFS客户端角色"。

   b. 如果选择使用指定自定义密钥 `beegfs_ha_conn_auth_secret` 在中指定 `client_inventory.yml` 文件：

```
beegfs_ha_conn_auth_secret: <SECRET>
```

   c. 如果选择完全禁用基于连接的身份验证 `beegfs_ha_conn_auth_enabled`、在中指定 `client_inventory.yml` 文件：

```
beegfs_ha_conn_auth_enabled: false
```

有关支持的参数的完整列表和其他详细信息、请参见 "完整的BeeGFS客户端文档"。有关客户端清单的完整示例、请单击 "此处"。

创建**BeeGFS**客户端攻略手册文件

1. 创建新文件 `client_playbook.yml`

```
# BeeGFS client playbook.
- hosts: beegfs_clients
  any_errors_fatal: true
  gather_facts: true
  collections:
    - netapp_eseries.beegfs
    - netapp_eseries.host
  tasks:
```

2. 可选：如果要使用NetApp E系列主机集合的角色配置客户端连接到BeeGFS文件系统的接口、请导入与要配置的接口类型对应的角色：

   a. 如果您使用的是使用InfiniBand (IPoIB)：

```
    - name: Ensure IPoIB is configured
      import_role:
        name: ipoib
```

   b. 如果您使用的是基于融合以太网的RDMA (RoCE)：

```
    - name: Ensure IPoIB is configured
      import_role:
        name: roce
```

   c. 如果您使用的是以太网(仅限TCP、无RDMA)：

```
    - name: Ensure IPoIB is configured
      import_role:
        name: ip
```

3. 最后、导入BeeGFS客户端角色以安装客户端软件并设置文件系统挂载：

```
    # REQUIRED: Install the BeeGFS client and mount the BeeGFS file
system.
    - name: Verify the BeeGFS clients are configured.
      import_role:
        name: beegfs_client
```

有关客户端攻略手册的完整示例、请单击 "此处"。

运行**BeeGFS**客户端攻略手册

要安装/构建客户端并挂载BeeGFS、请运行以下命令：

```
ansible-playbook -i client_inventory.yml client_playbook.yml
```

验证BeeGFS部署

在将系统投入生产之前、请验证文件系统部署。

概述

在将BeeGFS文件系统投入生产之前、请执行一些验证检查。

步骤

1. 登录到任何客户端并运行以下命令、以确保所有预期节点均存在/可访问、并且未报告任何不一致或其他问题：

```
beegfs-fsck --checkfs
```

2. 关闭整个集群、然后重新启动它。从任何文件节点运行以下命令：

```
pcs cluster stop --all # Stop the cluster on all file nodes.
pcs cluster start --all  # Start the cluster on all file nodes.
pcs status # Verify all nodes and services are started and no failures
are reported (the command may need to be reran a few times to allow time
for all services to start).
```

3. 将每个节点置于备用状态、并验证BeeGFS服务是否能够故障转移到二级节点。要登录到任何文件节点并运行以下命令：

```
pcs status # Verify the cluster is healthy at the start.
pcs node standby <FILE NODE HOSTNAME> # Place the node under test in
standby.
pcs status # Verify services are started on a secondary node and no
failures are reported.
pcs node unstandby <FILE NODE HOSTNAME> # Take the node under test out
of standby.
pcs status # Verify the file node is back online and no failures are
reported.
pcs resource relocate run # Move all services back to their preferred
nodes.
pcs status # Verify services have moved back to the preferred node.
```

4. 使用IOR和MDTest等性能基准测试工具验证文件系统性能是否满足预期。有关BeeGFS的常见测试和参数示例、请参见"设计验证"经验证的NetApp架构上的BeeGFS一节。

应根据为特定站点/安装定义的验收标准执行其他测试。

版权信息

商标信息