



定义 **BeeGFS** 服务

BeeGFS on NetApp with E-Series Storage

NetApp

January 27, 2026

目录

- 定义BeeGFS服务 1
 - 定义BeeGFS管理服务 1
 - 概述 1
 - 步骤 1
 - 定义BeeGFS元数据服务 2
 - 概述 2
 - 步骤 3
 - 定义BeeGFS存储服务 4
 - 概述 4
 - 步骤 4

定义BeeGFS服务

定义BeeGFS管理服务

BeeGFS服务使用组变量(group_vars)进行配置。

概述

本节将介绍如何定义BeeGFS管理服务。对于特定文件系统、HA集群中只应存在一个此类型的服务。配置此服务包括定义：

- 服务类型(管理)。
- 定义应仅适用于此BeeGFS服务的任何配置。
- 配置可访问此服务的一个或多个浮动IP (逻辑接口)。
- 指定卷存储此服务数据的位置/方式(BeeGFS管理目标)。

步骤

创建一个新文件 `group\_vars/mgmt.yml` 并引用该["规划文件系统"](#)部分、按如下所示填充该文件：

1. 指示此文件表示BeeGFS管理服务的配置：

```
beegfs_service: management
```

2. 定义应仅应用于此BeeGFS服务的任何配置。管理服务通常不需要执行此操作、除非您需要启用配额、无论您的任何受支持的配置参数如何 beegfs-mgmt.conf 可以包括在内。请注意、以下参数是自动/在其他位置配置的、不应在此处指定： storeMgmtDirectory, connAuthFile, connDisableAuthentication, connInterfacesFile, 和 connNetFilterFile。

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

3. 配置一个或多个浮动IP、供其他服务和客户端连接到此服务时使用(此操作将自动设置BeeGFS connInterfacesFile 选项)：

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

4. 或者、指定一个或多个允许用于传出通信的IP子网(此操作将自动设置BeeGFS connNetFilterFile 选项)：

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

5. 根据以下准则指定此服务将在其中存储数据的BeeGFS管理目标：

- 同一存储池或卷组名称可用于多个BeeGFS服务/目标、只需确保使用相同的即可 name, raid_level, criteria_*, 和 common_* 每个服务的配置(为每个服务列出的卷应不同)。
- 卷大小应指定为存储池/卷组的百分比、并且使用特定存储池/卷组的所有服务/卷的总数不应超过100。注意使用SSD时、建议在卷组中保留一些可用空间、以最大程度地提高SSD性能和使用寿命(单击可["此处"](#)了解更多信息)。
- 单击 ["此处"](#) 以获取可用于的配置选项的完整列表 eseries_storage_pool_configuration。请注意一些选项、例如 state, host, host_type, workload_name, 和 workload_metadata 和卷名称将自动生成、不应在此指定。

```
beegfs_targets:
  <BLOCK_NODE>: # The name of the block node as found in the Ansible
inventory. Ex: netapp_01
  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
```

单击 ["此处"](#) 有关表示BeeGFS管理服务的完整清单文件的示例。

定义BeeGFS元数据服务

BeeGFS服务使用组变量(group_vars)进行配置。

概述

本节将介绍如何定义BeeGFS元数据服务。对于特定文件系统、HA集群中应至少存在一个此类型的服务。配置此服务包括定义：

- 服务类型(元数据)。
- 定义应仅适用于此BeeGFS服务的任何配置。
- 配置可访问此服务的一个或多个浮动IP (逻辑接口)。
- 指定卷存储此服务数据的位置/方式(BeeGFS元数据目标)。

步骤

参考“[规划文件系统](#)”部分、`group\_vars/meta\_<ID>.yaml`为集群中的每个元数据服务创建一个位于的文件、然后按如下所示填充它们：

1. 指示此文件表示BeeGFS元数据服务的配置：

```
beegfs_service: metadata
```

2. 定义应仅应用于此BeeGFS服务的任何配置。您必须至少指定所需的TCP和UDP端口、无论这些端口来自哪个受支持的配置参数 `beegfs-meta.conf` 也可以包括在内。请注意、以下参数是自动/在其他位置配置的、不应在此处指定： `sysMgmtHost`， `storeMetaDirectory`， `connAuthFile`， `connDisableAuthentication`， `connInterfacesFile`， 和 `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. 配置一个或多个浮动IP、供其他服务和客户端连接到此服务时使用(此操作将自动设置BeeGFS `connInterfacesFile` 选项)：

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

4. 或者、指定一个或多个允许用于传出通信的IP子网(此操作将自动设置BeeGFS `connNetFilterFile` 选项)：

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

5. 根据以下准则指定此服务将在其中存储数据的BeeGFS元数据目标(此操作也会自动配置) `storeMetaDirectory` 选项)：

- a. 同一存储池或卷组名称可用于多个BeeGFS服务/目标、只需确保使用相同的即可 `name`， `raid_level`， `criteria_*`， 和 `common_*` 每个服务的配置(为每个服务列出的卷应不同)。
- b. 卷大小应指定为存储池/卷组的百分比、并且使用特定存储池/卷组的所有服务/卷的总数不应超过100。注意使用SSD时、建议在卷组中保留一些可用空间、以最大程度地提高SSD性能和使用寿命(单击可[此处](#)了解更多详细信息)。
- c. 单击 [此处](#) 以获取可用于的配置选项的完整列表 `eseries_storage_pool_configuration`。请注意一些选项、例如 `state`， `host`， `host_type`， `workload_name`， 和 `workload_metadata` 和卷

名称将自动生成、不应在此指定。

```
beegfs_targets:
  <BLOCK_NODE>: # The name of the block node as found in the Ansible
inventory. Ex: netapp_01
  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
```

单击 ["此处"](#) 有关表示BeeGFS元数据服务的完整清单文件的示例。

定义BeeGFS存储服务

BeeGFS服务使用组变量(group_vars)进行配置。

概述

本节将介绍如何定义BeeGFS存储服务。对于特定文件系统、HA集群中应至少存在一个此类型的服务。配置此服务包括定义：

- 服务类型(存储)。
- 定义应仅适用于此BeeGFS服务的任何配置。
- 配置可访问此服务的一个或多个浮动IP (逻辑接口)。
- 指定卷存储此服务数据的位置/方式(BeeGFS存储目标)。

步骤

参考["规划文件系统"](#)部分、`group\_vars/stor\_<ID>.yaml`为集群中的每个存储服务创建一个文件、然后按如下所示填充它们：

1. 指示此文件表示BeeGFS存储服务的配置：

```
beegfs_service: storage
```

2. 定义应仅应用于此BeeGFS服务的任何配置。您必须至少指定所需的TCP和UDP端口、无论这些端口来自哪个受支持的配置参数 `beegfs-storage.conf` 也可以包括在内。请注意、以下参数是自动/在其他位置配置的、不应在此处指定：`sysMgmdHost`，`storeStorageDirectory`，`connAuthFile`，

connDisableAuthentication, connInterfacesFile, 和 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. 配置一个或多个浮动IP、供其他服务和客户端连接到此服务时使用(此操作将自动设置BeeGFS connInterfacesFile 选项):

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

4. 或者、指定一个或多个允许用于传出通信的IP子网(此操作将自动设置BeeGFS connNetFilterFile 选项):

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

5. 根据以下准则指定此服务要存储数据的BeeGFS存储目标(此操作也会自动配置) storeStorageDirectory 选项):
 - a. 同一存储池或卷组名称可用于多个BeeGFS服务/目标、只需确保使用相同的即可 name, raid_level, criteria_*, 和 common_* 每个服务的配置(为每个服务列出的卷应不同)。
 - b. 卷大小应指定为存储池/卷组的百分比、并且使用特定存储池/卷组的所有服务/卷的总数不应超过100。注意使用SSD时、建议在卷组中保留一些可用空间、以最大程度地提高SSD性能和使用寿命(单击可["此处"](#)了解更多详细信息)。
 - c. 单击 ["此处"](#) 以获取可用于的配置选项的完整列表 eseries_storage_pool_configuration。请注意一些选项、例如 state, host, host_type, workload_name, 和 workload_metadata 和卷名称将自动生成、不应在此指定。

```

beegfs_targets:
  <BLOCK_NODE>: # The name of the block node as found in the Ansible
inventory. Ex: netapp_01
  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

```

单击 ["此处"](#) 有关表示BeeGFS存储服务的完整清单文件的示例。

版权信息

版权所有 © 2026 NetApp, Inc.。保留所有权利。中国印刷。未经版权所有者事先书面许可，本文档中受版权保护的任何部分不得以任何形式或通过任何手段（图片、电子或机械方式，包括影印、录音、录像或存储在电子检索系统中）进行复制。

从受版权保护的 NetApp 资料派生的软件受以下许可和免责声明的约束：

本软件由 NetApp 按“原样”提供，不含任何明示或暗示担保，包括但不限于适销性以及针对特定用途的适用性的隐含担保，特此声明不承担任何责任。在任何情况下，对于因使用本软件而以任何方式造成的任何直接性、间接性、偶然性、特殊性、惩罚性或后果性损失（包括但不限于购买替代商品或服务；使用、数据或利润方面的损失；或者业务中断），无论原因如何以及基于何种责任理论，无论出于合同、严格责任或侵权行为（包括疏忽或其他行为），NetApp 均不承担责任，即使已被告知存在上述损失的可能性。

NetApp 保留在不另行通知的情况下随时对本文档所述的任何产品进行更改的权利。除非 NetApp 以书面形式明确同意，否则 NetApp 不承担因使用本文档所述产品而产生的任何责任或义务。使用或购买本产品不表示获得 NetApp 的任何专利权、商标权或任何其他知识产权许可。

本手册中描述的产品可能受一项或多项美国专利、外国专利或正在申请的专利的保护。

有限权利说明：政府使用、复制或公开本文档受 DFARS 252.227-7013（2014 年 2 月）和 FAR 52.227-19（2007 年 12 月）中“技术数据权利 — 非商用”条款第 (b)(3) 条规定的限制条件的约束。

本文档中所含数据与商业产品和/或商业服务（定义见 FAR 2.101）相关，属于 NetApp, Inc. 的专有信息。根据本协议提供的所有 NetApp 技术数据和计算机软件具有商业性质，并完全由私人出资开发。美国政府对这些数据的使用权具有非排他性、全球性、受限且不可撤销的许可，该许可既不可转让，也不可再许可，但仅限在与交付数据所依据的美国政府合同有关且受合同支持的情况下使用。除本文档规定的情形外，未经 NetApp, Inc. 事先书面批准，不得使用、披露、复制、修改、操作或显示这些数据。美国政府对国防部的授权仅限于 DFARS 的第 252.227-7015(b)（2014 年 2 月）条款中明确的权利。

商标信息

NetApp、NetApp 标识和 <http://www.netapp.com/TM> 上所列的商标是 NetApp, Inc. 的商标。其他公司和产品名称可能是其各自所有者的商标。