准备主机（Linux）

StorageGRID software

NetApp
July 23, 2026

目录

准备主机（ Linux ）

StorageGRID 如何更改主机范围的设置

在裸机系统上，StorageGRID 会对主机范围 `sysctl`的设置进行一些更改。

"Linux"是指 RHEL、Ubuntu 或 Debian 部署。有关受支持版本的列表，请参见 "NetApp 互操作性矩阵工具 (IMT)"。

此时将进行以下更改：

```
# Recommended Cassandra setting: CASSANDRA-3563, CASSANDRA-13008, DataStax
documentation
vm.max_map_count = 1048575

# core file customization
# Note: for cores generated by binaries running inside containers, this
# path is interpreted relative to the container filesystem namespace.
# External cores will go nowhere, unless /var/local/core also exists on
# the host.
kernel.core_pattern = /var/local/core/%e.core.%p

# Set the kernel minimum free memory to the greater of the current value
or
# 512MiB if the host has 48GiB or less of RAM or 1.83GiB if the host has
more than 48GiB of RTAM
vm.min_free_kbytes = 524288

# Enforce current default swappiness value to ensure the VM system has
some
# flexibility to garbage collect behind anonymous mappings. Bump
watermark_scale_factor
# to help avoid OOM conditions in the kernel during memory allocation
bursts. Bump
# dirty_ratio to 90 because we explicitly fsync data that needs to be
persistent, and
# so do not require the dirty_ratio safety net. A low dirty_ratio combined
with a large
# working set (nr_active_pages) can cause us to enter synchronous I/O mode
unnecessarily,
# with deleterious effects on performance.
vm.swappiness = 60
vm.watermark_scale_factor = 200
vm.dirty_ratio = 90
```

```
# Turn off slow start after idle
net.ipv4.tcp_slow_start_after_idle = 0

# Tune TCP window settings to improve throughput
net.core.rmem_max = 8388608
net.core.wmem_max = 8388608
net.ipv4.tcp_rmem = 4096 524288 8388608
net.ipv4.tcp_wmem = 4096 262144 8388608
net.core.netdev_max_backlog = 2500

# Turn on MTU probing
net.ipv4.tcp_mtu_probing = 1

# Be more liberal with firewall connection tracking
net.ipv4.netfilter.ip_conntrack_tcp_be_liberal = 1

# Reduce TCP keepalive time to reasonable levels to terminate dead
connections
net.ipv4.tcp_keepalive_time = 270
net.ipv4.tcp_keepalive_probes = 3
net.ipv4.tcp_keepalive_intvl = 30

# Increase the ARP cache size to tolerate being in a /16 subnet
net.ipv4.neigh.default.gc_thresh1 = 8192
net.ipv4.neigh.default.gc_thresh2 = 32768
net.ipv4.neigh.default.gc_thresh3 = 65536
net.ipv6.neigh.default.gc_thresh1 = 8192
net.ipv6.neigh.default.gc_thresh2 = 32768
net.ipv6.neigh.default.gc_thresh3 = 65536

# Disable IP forwarding, we are not a router
net.ipv4.ip_forward = 0

# Follow security best practices for ignoring broadcast ping requests
net.ipv4.icmp_echo_ignore_broadcasts = 1

# Increase the pending connection and accept backlog to handle larger
connection bursts.
net.core.somaxconn=4096
net.ipv4.tcp_max_syn_backlog=4096
```

在 **Linux** 网格主机上安装 **StorageGRID**

您必须在所有 Linux 网格主机上安装 StorageGRID。有关受支持版本的列表，请使用 NetApp 互操作性矩阵工具。

开始之前

确保您的操作系统满足 StorageGRID 的最低内核版本要求，如下所示。使用命令 `uname -r`获取操作系统的内核版本，或咨询操作系统供应商。

"Linux"是指 RHEL、Ubuntu 或 Debian 部署。有关受支持版本的列表，请参见 "NetApp 互操作性矩阵工具 (IMT)"。

RHEL

RHEL 版本	最低内核版本	内核包名称
8.8 （已弃用）	4.18.0-477.10.1.el8_8.x86_64	kernel-4.18.0-477.10.1.el8_8.x86_64
8.10	4.18.0-553.el8_10.x86_64	kernel-4.18.0-553.el8_10.x86_64
9.0（已弃用）	5.14.0-70.22.1.el9_0.x86_64	kernel-5.14.0-70.22.1.el9_0.x86_64
9.2 （已弃用）	5.14.0-284.11.1.el9_2.x86_64	kernel-5.14.0-284.11.1.el9_2.x86_64
9.4	5.14.0-427.18.1.el9_4.x86_64	kernel-5.14.0-427.18.1.el9_4.x86_64
9.6	5.14.0-570.18.1.el9_6.x86_64	kernel-5.14.0-570.18.1.el9_6.x86_64

Ubuntu

注意： 对 Ubuntu 版本 18.04 和 20.04 的支持已弃用，将在未来版本中删除。

Ubuntu 版本	最低内核版本	内核包名称
22.04.1	5.15.0-47-generic	linux-image-5.15.0-47-generic/jammy-updates,jammy-security,now 5.15.0-47.51
24.04	6.8.0-31-generic	linux-image-6.8.0-31-generic/noble, 现在 6.8.0-31.31

Debian

注意： 对 Debian 版本 11 的支持已被弃用，并将在未来版本中删除。

Debian 版本	最低内核版本	内核包名称
11 (已弃用)	5.10.0-18-amd64	linux-image-5.10.0-18-amd64/stable,now 5.10.150-1
12	6.1.0-9-amd64	linux-image-6.1.0-9-amd64/stable,now 6.1.27-1

步骤

1. 根据分销商的说明或您的标准程序，在所有物理或虚拟网格主机上安装 Linux。

   请勿安装任何图形桌面环境。

   - 如果在安装 RHEL 时使用标准 Linux 安装程序，请选择"compute node"软件配置（如果有）或"minimal install"基本环境。
   - 安装 Ubuntu 时，您必须选择 标准系统实用程序。建议选择 **OpenSSH server** 以启用对 Ubuntu 主机的 ssh 访问。所有其他选项均可保持清除状态。

2. 确保所有主机都可以访问软件包存储库，包括 RHEL 的 Extras 频道。
3. 如果启用了 swap：
   a. 运行以下命令： `$ sudo swapoff --all`
   b. 从 `/etc/fstab`中删除所有交换条目以保留设置。

      如果不能完全禁用交换，可能会严重降低性能。

了解 Ubuntu 和 Debian 上的 AppArmor 配置文件用于 StorageGRID

如果您在自行部署的 Ubuntu 环境中运行并使用 AppArmor 强制访问控制系统，则与基本系统上安装的软件包相关联的 AppArmor 配置文件可能会被随 StorageGRID 一起安装的相应软件包阻止。

默认情况下，将为您在基本操作系统上安装的软件包安装 AppArmor 配置文件。当您从 StorageGRID 系统容器运行这些软件包时，AppArmor 配置文件将被阻止。DHCP、MySQL、NTP 和 tcdump 基本软件包与 AppArmor 冲突，其他基本软件包也可能发生冲突。

处理AppArmor配置文件有两种选择：

- 禁用基本系统上安装的与 StorageGRID 系统容器中的软件包重叠的软件包的单个配置文件。禁用单个配置文件时，StorageGRID 日志文件中会显示一个条目，指示 AppArmor 已启用。

  使用以下命令：

  ```
  sudo ln -s /etc/apparmor.d/<profile.name> /etc/apparmor.d/disable/
  sudo apparmor_parser -R /etc/apparmor.d/<profile.name>
  ```

  示例：

  ```
  sudo ln -s /etc/apparmor.d/bin.ping /etc/apparmor.d/disable/
  sudo apparmor_parser -R /etc/apparmor.d/bin.ping
  ```

- 完全禁用 AppArmor。对于 Ubuntu 9.10 或更高版本，请按照 Ubuntu 在线社区中的说明进行操作： "禁用 AppArmor"。在较新的 Ubuntu 版本上可能无法完全禁用 AppArmor。

  禁用 AppArmor 后，任何指示 AppArmor 已启用的条目都不会出现在 StorageGRID 日志文件中。

为 Linux 部署配置 StorageGRID 主机网络

在主机上完成 Linux 安装后，您可能需要执行一些额外的配置，以便在每台主机上准备一组网络接口，这些接口适合映射到稍后将部署的 StorageGRID 节点。

"Linux"是指 RHEL、Ubuntu 或 Debian 部署。有关受支持版本的列表，请参见 "NetApp 互操作性矩阵工具 (IMT)"。

开始之前

- 您已审阅 "StorageGRID 网络准则"。
- 您已查看有关 "节点容器迁移要求" 的信息。
- 如果您使用的是虚拟主机，请在配置主机网络之前阅读MAC 地址克隆的注意事项和建议。

如果使用 VM 作为主机，则应选择 VMXNET 3 作为虚拟网络适配器。VMware E1000 网络适配器导致某些 Linux 发行版上部署的 StorageGRID 容器出现连接问题。

关于此任务

网格节点必须能够访问网格网络，以及可选的管理和客户端网络。您可以通过创建将主机的物理接口与每个网格节点的虚拟接口相关联的映射来提供此访问。创建主机接口时，请使用友好名称以方便跨所有主机的部署，并启用迁移。

主机和一个或多个节点之间可以共享相同的接口。例如，您可能对主机访问和节点管理网络访问使用相同的接口，以方便主机和节点的维护。虽然主机和单个节点之间可以共享相同的接口，但所有节点必须具有不同的 IP 地址。IP 地址不能在节点之间或主机与任何节点之间共享。

您可以使用相同的主机网络接口为主机上的所有 StorageGRID 节点提供网格网络接口；您可以为每个节点使用不同的主机网络接口；或者您也可以采用介于两者之间的方式。但是，您通常不会为单个节点同时提供网格网络接口和管理网络接口使用相同的主机网络接口，也不会将某一主机网络接口同时用作一个节点的网格网络接口和另一个节点的客户端网络接口。

您可以通过多种方式完成此任务。例如，如果您的主机是虚拟机，并且您要为每个主机部署一个或两个 StorageGRID 节点，则可以在虚拟机监控程序中创建正确数量的网络接口，并使用 1 对 1 映射。如果要在裸机主机上部署多个节点以供生产使用，则可以利用 Linux 网络堆栈对 VLAN 和 LACP 的支持来实现容错和带宽共享。以下部分提供了这两个示例的详细方法。您无需使用这些示例中的任何一个；您可以使用任何符合您需求的方法。

请勿直接使用绑定或桥接设备作为容器网络接口。这样做可能会导致在容器命名空间中将 MACVLAN 与绑定和桥接设备结合使用时，因内核问题而无法启动节点。请改用非绑定设备，例如 VLAN 或虚拟以太网 (veth) 对。在节点配置文件中将此设备指定为网络接口。

MAC 地址克隆的注意事项和建议

MAC 地址克隆会导致容器使用主机的 MAC 地址，而主机使用您指定的地址或随机生成的地址的 MAC 地址。您应该使用 MAC 地址克隆，以避免使用混杂模式网络配置。

启用MAC克隆

在某些环境中，可以通过 MAC 地址克隆来增强安全性，因为它使您能够为管理网络、网格网络和客户端网络使用专用的虚拟 NIC。通过让容器使用主机上专用 NIC 的 MAC 地址，您可以避免使用混杂模式网络配置。

> MAC 地址克隆旨在与虚拟服务器安装配合使用，可能无法在所有物理设备配置中正常运行。

> 如果由于 MAC 克隆目标接口繁忙而无法启动节点，则可能需要在启动节点之前将链路设置为"down"。此外，当链路处于启动状态时，虚拟环境可能会阻止网络接口上的 MAC 克隆。如果节点由于接口繁忙而无法设置 MAC 地址并启动，则在启动节点之前将链路设置为"down"可能会解决此问题。

默认情况下，MAC 地址克隆处于禁用状态，必须通过节点配置密钥进行设置。您应在安装 StorageGRID 时启用此功能。

每个网络都有一个密钥：

- `ADMIN_NETWORK_TARGET_TYPE_INTERFACE_CLONE_MAC`
- `GRID_NETWORK_TARGET_TYPE_INTERFACE_CLONE_MAC`
- `CLIENT_NETWORK_TARGET_TYPE_INTERFACE_CLONE_MAC`

将密钥设置为"true"将导致容器使用主机 NIC 的 MAC 地址。此外，主机将使用指定容器网络的 MAC 地址。默认情况下，容器地址是随机生成的地址，但如果使用 `_NETWORK_MAC`节点配置密钥设置了地址，则将使用该地址。主机和容器将始终具有不同的 MAC 地址。

> 在虚拟主机上启用 MAC 克隆而不同时在虚拟机监控程序上启用混杂模式，可能会导致使用主机接口的 Linux 主机网络停止工作。

MAC克隆使用案例

使用MAC克隆时需要考虑两种用例：

- 未启用 MAC 克隆：当节点配置文件中的 `_CLONE_MAC`密钥未设置或设置为 "false" 时，主机将使用主机 NIC MAC，除非在 `_NETWORK_MAC`密钥中指定了 MAC，否则容器将使用 StorageGRID 生成的 MAC。如果在 `_NETWORK_MAC`密钥中设置了地址，则容器将使用 `_NETWORK_MAC`密钥中指定的地址。这种密钥配置需要使用混杂模式。
- 启用 MAC 克隆：当节点配置文件中的 `_CLONE_MAC`密钥设置为"true"时，容器使用主机 NIC MAC，主机使用 StorageGRID 生成的 MAC，除非在 `_NETWORK_MAC`密钥中指定了 MAC。如果在 `_NETWORK_MAC`密钥中设置了地址，则主机使用指定的地址而不是生成的地址。在此密钥配置中，不应使用混杂模式。

如果您不想使用 MAC 地址克隆，而是希望允许所有接口接收和传输虚拟机管理程序所分配地址以外的 MAC 地址数据，请确保虚拟交换机和端口组级别的安全属性针对混杂模式、MAC 地址更改和伪造传输均设置为 **Accept**。虚拟交换机上设置的值可被端口组级别的值覆盖，因此请确保两处的设置相同。

要启用 MAC 克隆，请参见 "节点配置文件创建说明"。

MAC克隆示例

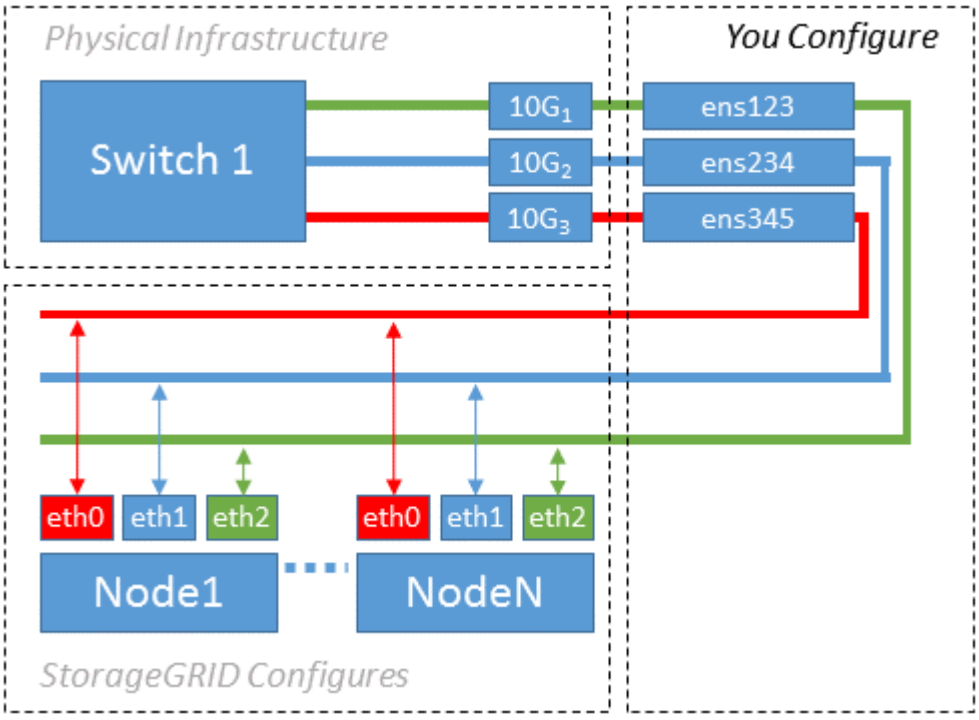
以下示例展示了主机 MAC 地址为 11:22:33:44:55:66 的接口 ens256 启用 MAC 克隆，以及节点配置文件中的相关密钥：

- `ADMIN_NETWORK_TARGET = ens256`
- `ADMIN_NETWORK_MAC = b2:9c:02:c2:27:10`
- `ADMIN_NETWORK_TARGET_TYPE_INTERFACE_CLONE_MAC = true`

结果：ens256 的主机 MAC 为 b2:9c:02:c2:27:10，Admin Network MAC 为 11:22:33:44:55:66

示例 1：1 对 1 映射到物理或虚拟 NIC

示例 1 描述了一个简单的物理接口映射，只需极少或无需主机端配置。



Linux 操作系统在安装或引导期间 `ensXYZ`，或在热添加接口时自动创建接口。除了确保将接口设置为在启动后自动启动外，无需进行任何配置。您必须确定哪个 ensXYZ 对应于哪个 StorageGRID 网络（网格、管理员或客户端），以便稍后在配置过程中提供正确的映射。

请注意，图中显示了多个 StorageGRID 节点；但是，通常会对单节点虚拟机使用此配置。

如果交换机 1 是物理交换机，则应将连接到接口 10G1 至 10G3 的端口配置为接入模式，并将其放置在相应的

VLAN 上。

示例 2：承载 VLAN 的 LACP 绑定

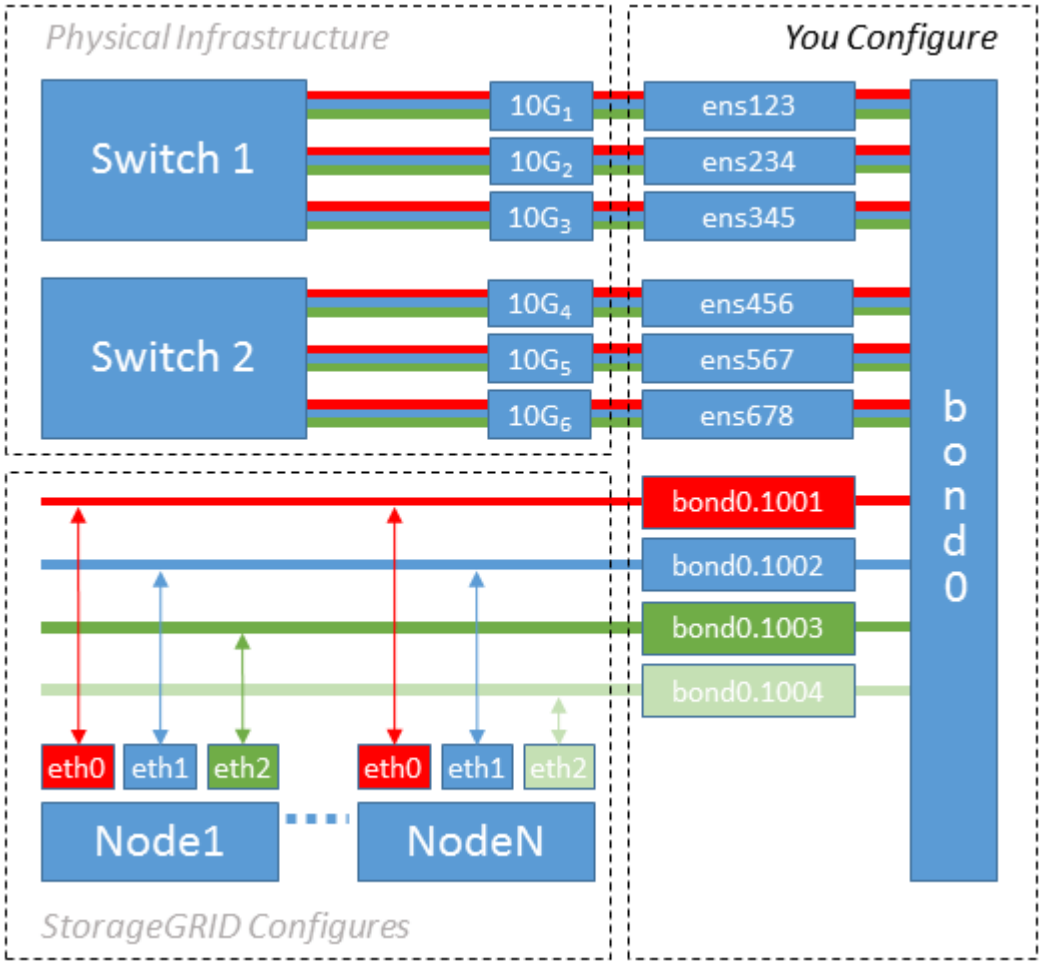
示例 2 假设您熟悉绑定网络接口以及在使用的 Linux 发行版上创建 VLAN 接口。

关于此任务

示例 2 描述了一种通用的、灵活的、基于 VLAN 的方案，该方案便于在单个主机上的所有节点之间共享所有可用的网络带宽。此示例尤其适用于裸机主机。

要理解此示例，假设每个数据中心的网格、管理和客户端网络有三个独立的子网。这些子网位于单独的 VLAN（1001、1002 和 1003）上，并通过 LACP 绑定中继端口（bond0）呈现给主机。您需要在该绑定上配置三个 VLAN 接口：bond0.1001、bond0.1002 和 bond0.1003。

如果您需要在同一主机上为节点网络提供单独的 VLAN 和子网，您可以在绑定上添加 VLAN 接口并将其映射到主机（在图中显示为 bond0.1004）。



步骤

1. 将用于 StorageGRID 网络连接的所有物理网络接口聚合到一个 LACP 绑定中。

   在每台主机上使用相同的绑定名称，例如 `bond0`。

2. 使用标准 VLAN 接口命名约定，创建使用此绑定作为其关联"物理设备"的 VLAN 接口 `physdev-`

`name.VLAN ID`。

请注意，步骤 1 和 2 需要在端接网络链路另一端的边缘交换机上进行适当的配置。边缘交换机端口还必须聚合到 LACP 端口通道中，配置为中继，并允许通过所有必需的 VLAN。

提供了此每主机网络配置方案的示例接口配置文件。

相关信息

- "Ubuntu 和 Debian 的 /etc/network/interfaces 示例"
- "RHEL 的 /etc/sysconfig/network-scripts 示例"

为 Linux 部署配置 StorageGRID 主机存储

必须为每个 Linux 主机分配块存储卷。

开始之前

您已经查看了以下主题，这些主题提供了完成此任务所需的信息：

- "存储和性能要求"
- "节点容器迁移要求"

> "Linux"是指 RHEL、Ubuntu 或 Debian 部署。有关受支持版本的列表，请参见 "NetApp 互操作性矩阵工具 (IMT)"。

关于此任务

将块存储卷（LUN）分配给主机时，请使用"存储要求"中的表确定以下内容：

- 每个主机所需的卷数量（基于将在该主机上部署的节点的数量和类型）
- 每个卷的存储类别（即系统数据或对象数据）
- 每卷的大小

在主机上部署 StorageGRID 节点时，您将使用此信息以及 Linux 分配给每个物理卷的持久名称。

> 您无需对这些卷进行分区、格式化或装载，只需确保它们对主机可见即可。

> 仅元数据存储节点只需要一个对象数据 LUN。

在编写卷名列表时，请避免使用"原始"特殊设备文件 (`/dev/sdb`。这些文件可能会在主机重新启动时发生变化，这将影响系统的正常运行。如果要使用 iSCSI LUN 和设备映射器多路径，请考虑在 `/dev/mapper`目录中使用多路径别名，特别是当 SAN 拓扑包含到共享存储的冗余网络路径时。或者，您可以使用 `/dev/disk/by-path/`下方系统创建的软链接作为永久设备名称。

例如：

```
ls -l
$ ls -l /dev/disk/by-path/
total 0
lrwxrwxrwx 1 root root  9 Sep 19 18:53 pci-0000:00:07.1-ata-2 -> ../../sr0
lrwxrwxrwx 1 root root  9 Sep 19 18:53 pci-0000:03:00.0-scsi-0:0:0:0 ->
../../sda
lrwxrwxrwx 1 root root 10 Sep 19 18:53 pci-0000:03:00.0-scsi-0:0:0:0-part1
-> ../../sda1
lrwxrwxrwx 1 root root 10 Sep 19 18:53 pci-0000:03:00.0-scsi-0:0:0:0-part2
-> ../../sda2
lrwxrwxrwx 1 root root  9 Sep 19 18:53 pci-0000:03:00.0-scsi-0:0:1:0 ->
../../sdb
lrwxrwxrwx 1 root root  9 Sep 19 18:53 pci-0000:03:00.0-scsi-0:0:2:0 ->
../../sdc
lrwxrwxrwx 1 root root  9 Sep 19 18:53 pci-0000:03:00.0-scsi-0:0:3:0 ->
../../sdd
```

每个安装的结果会有所不同。

为每个块存储卷分配友好名称，以简化初始 StorageGRID 安装和未来的维护过程。如果要使用设备映射器多路径驱动程序对共享存储卷进行冗余访问，则可以使用 `alias`字段（位于 `/etc/multipath.conf`文件中）。

例如：

```
multipaths {
     multipath {
          wwid 3600a09800059d6df00005df2573c2c30
          alias docker-storage-volume-hostA
     }
     multipath {
          wwid 3600a09800059d6df00005df3573c2c30
          alias sgws-adm1-var-local
     }
     multipath {
          wwid 3600a09800059d6df00005df4573c2c30
          alias sgws-adm1-audit-logs
     }
     multipath {
          wwid 3600a09800059d6df00005df5573c2c30
          alias sgws-adm1-tables
     }
     multipath {
          wwid 3600a09800059d6df00005df6573c2c30
          alias sgws-gw1-var-local
     }
     multipath {
          wwid 3600a09800059d6df00005df7573c2c30
          alias sgws-sn1-var-local
     }
     multipath {
          wwid 3600a09800059d6df00005df7573c2c30
          alias sgws-sn1-rangedb-0
     }
     …
```

以这种方式使用别名字段会使别名在主机上的 `/dev/mapper`目录中显示为块设备，从而允许您在配置或维护操作需要指定块存储卷时指定易于识别和验证的名称。

如果您正在设置共享存储以支持 StorageGRID 节点迁移并使用 Device Mapper 多路径，则可以在所有同地主机上创建和安装通用 `/etc/multipath.conf`。只需确保在每台主机上使用不同的容器引擎存储卷。使用别名并在每个容器引擎存储卷 LUN 的别名中包含目标主机名将使其易于记忆，建议这样做。

对于纯软件部署，使用 Docker 作为容器引擎的支持已弃用。Docker 将在未来版本中替换为另一个容器引擎。

相关信息

- "配置容器引擎存储卷"
- "存储和性能要求"
- "节点容器迁移要求"

在 Linux 上为 StorageGRID 配置容器引擎存储卷

在安装 Docker 或 Podman 容器引擎之前，可能需要格式化存储卷并挂载它。

对于纯软件部署，使用 Docker 作为容器引擎的支持已弃用。Docker 将在未来版本中替换为另一个容器引擎。

"Linux"是指 RHEL、Ubuntu 或 Debian 部署。有关受支持版本的列表，请参见 "NetApp 互操作性矩阵工具 (IMT)"。

关于此任务

如果您计划将根卷用于 Docker 或 Podman 存储卷，并且在包含以下内容的主机分区上有足够的可用空间，则可以跳过这些步骤：

- Podman： `/var/lib/containers`
- Docker： `/var/lib/docker`

步骤

1. 在容器引擎存储卷上创建文件系统：

RHEL

```
sudo mkfs.ext4 container-engine-storage-volume-device
```

Ubuntu或Debian

```
sudo mkfs.ext4 docker-storage-volume-device
```

2. 挂载容器引擎存储卷：

RHEL

- 对于 Docker：

```
sudo mkdir -p /var/lib/docker
sudo mount container-storage-volume-device /var/lib/docker
```

- 对于 Podman：

```
sudo mkdir -p /var/lib/containers
sudo mount container-storage-volume-device /var/lib/containers
```

Ubuntu或Debian

```
sudo mkdir -p /var/lib/docker
sudo mount docker-storage-volume-device /var/lib/docker
```

- 对于 Podman：

```
sudo mkdir -p /var/lib/podman
sudo mount container-storage-volume-device /var/lib/podman
```

3. 将容器存储卷设备的条目添加到 /etc/fstab。
   - RHEL: container-storage-volume-device
   - Ubuntu 或 Debian：docker-storage-volume-device

   此步骤可确保主机重新启动后存储卷将自动重新装载。

安装 Docker

StorageGRID 系统可以作为容器集合在 Linux 上运行。

- 在为 Ubuntu 或 Debian 安装 StorageGRID 之前，您必须先安装 Docker。
- 如果您选择使用 Docker 容器引擎，请按照以下步骤安装 Docker。否则，安装 Podman。

> 对于纯软件部署，使用 Docker 作为容器引擎的支持已弃用。Docker 将在未来版本中替换为另一个容器引擎。

步骤

1. 按照适用于您的 Linux 发行版的说明安装 Docker。

如果您的 Linux 发行版不包含 Docker，您可以从 Docker 网站下载它。

2. 运行以下两个命令，以确保 Docker 已启用并启动：

```
sudo systemctl enable docker
```

```
sudo systemctl start docker
```

3. 通过输入以下命令，确认您已安装 Docker 的预期版本：

```
sudo docker version
```

客户端和服务器版本必须为 1.11.0 或更高版本。

安装 Podman

StorageGRID 系统作为容器集合运行。如果您选择使用 Podman 容器引擎，请按照以下步骤安装 Podman。否则，安装 Docker。

步骤

1. 按照 Linux 发行版的说明安装 Podman 和 Podman-Docker。

安装 Podman 时，还必须安装 Podman-Docker 软件包。

2. 通过输入以下命令，确认您已安装预期版本的 Podman 和 Podman-Docker：

```
sudo docker version
```

Podman-Docker 包允许您使用 Docker 命令。

客户端和服务器版本必须为 3.2.3 或更高版本。

```
Version: 3.2.3
API Version: 3.2.3
Go Version: go1.15.7
Built: Tue Jul 27 03:29:39 2021
OS/Arch: linux/amd64
```

相关信息

"配置主机存储"

在 Linux 上安装 StorageGRID 主机服务

您可以使用适用于您操作系统类型的 StorageGRID 软件包来安装 StorageGRID 主机服务。

"Linux"是指 RHEL、Ubuntu 或 Debian 部署。有关受支持版本的列表，请参见 "NetApp 互操作性矩阵工具 (IMT)"。

RHEL

您可以使用 StorageGRID RPM 包安装 StorageGRID 主机服务。

关于此任务

这些说明介绍了如何从 RPM 包安装主机服务。作为替代方案，您可以使用安装存档中包含的 DNF 存储库元数据远程安装 RPM 包。请参阅适用于您的 Linux 操作系统的 DNF 存储库说明。

步骤

1. 将 StorageGRID RPM 软件包复制到每个主机，或使其在共享存储上可用。

   例如，将它们放置在 `/tmp`目录中，以便您可以在下一步中使用示例命令。

2. 以 root 身份或使用具有 sudo 权限的帐户登录到每个主机，并按指定的顺序运行以下命令：

   ```
   sudo dnf --nogpgcheck localinstall /tmp/StorageGRID-Webscale-Images-version-SHA.rpm
   ```

   ```
   sudo dnf --nogpgcheck localinstall /tmp/StorageGRID-Webscale-Service-version-SHA.rpm
   ```

   必须先安装 Images 包，然后再安装 Service 包。

   如果将包放置在 `/tmp`以外的目录中，请修改命令以反映所使用的路径。

Ubuntu或Debian

您可以使用 StorageGRID DEB 包为 Ubuntu 或 Debian 安装 StorageGRID 主机服务。

关于此任务

这些说明描述了如何从 DEB 包安装主机服务。作为替代方案，您可以使用安装存档中包含的 APT 存储库元数据远程安装 DEB 包。请参阅适用于您的 Linux 操作系统的 APT 存储库说明。

步骤

1. 将 StorageGRID DEB 包复制到每个主机，或使其在共享存储上可用。

   例如，将它们放置在 `/tmp`目录中，以便您可以在下一步中使用示例命令。

2. 以 root 身份或使用具有 sudo 权限的帐户登录到每个主机，并运行以下命令。

   您必须先安装 `images`包，然后再安装 `service`包。如果将包放置在 `/tmp`以外的目录中，请修改命令以反映所使用的路径。

   ```
   sudo dpkg --install /tmp/storagegrid-webscale-images-version-SHA.deb
   ```

```
sudo dpkg --install /tmp/storagegrid-webscale-service-version-
SHA.deb
```

必须先安装 Python 3，然后才能安装 StorageGRID 软件包。 `sudo dpkg --install /tmp/storagegrid-webscale-images-version-SHA.deb` 命令将失败，直到您完成此操作。