



Plan and prepare for installation on Ubuntu or Debian

StorageGRID 11.8

NetApp
May 17, 2024

Table of Contents

- Plan and prepare for installation on Ubuntu or Debian 1
 - Required information and materials 1
 - Download and extract the StorageGRID installation files 2
 - Software requirements for Ubuntu and Debian 4
 - CPU and RAM requirements 5
 - Storage and performance requirements 6
 - Node container migration requirements 10
 - Prepare the hosts (Ubuntu or Debian) 12

Plan and prepare for installation on Ubuntu or Debian

Required information and materials

Before you install StorageGRID, gather and prepare the required information and materials.

Required information

Network plan

Which networks you intend to attach to each StorageGRID node. StorageGRID supports multiple networks for traffic separation, security, and administrative convenience.

See the StorageGRID [Networking guidelines](#).

Network information

Unless you are using DHCP, IP addresses to assign to each grid node and the IP addresses of the DNS and NTP servers.

Servers for grid nodes

Identify a set of servers (physical, virtual, or both) that, in aggregate, provide sufficient resources to support the number and type of StorageGRID nodes you plan to deploy.



If your StorageGRID installation will not use StorageGRID appliance (hardware) Storage Nodes, you must use hardware RAID storage with battery-backed write cache (BBWC). StorageGRID does not support the use of virtual storage area networks (vSANs), software RAID, or no RAID protection.

Node migration (if needed)

Understand the [requirements for node migration](#), if you want to perform scheduled maintenance on physical hosts without any service interruption.

Related information

[NetApp Interoperability Matrix Tool](#)

Required materials

NetApp StorageGRID license

You must have a valid, digitally signed NetApp license.



A non-production license, which can be used for testing and proof of concept grids, is included in the StorageGRID installation archive.

StorageGRID installation archive

[Download the StorageGRID installation archive and extract the files.](#)

Service laptop

The StorageGRID system is installed through a service laptop.

The service laptop must have:

- Network port
- SSH client (for example, PuTTY)
- [Supported web browser](#)

StorageGRID documentation

- [Release notes](#)
- [Instructions for administering StorageGRID](#)

Download and extract the StorageGRID installation files

You must download the StorageGRID installation archive and extract the required files.

Steps

1. Go to the [NetApp Downloads page for StorageGRID](#).
2. Select the button for downloading the latest release, or select another version from the drop-down menu and select **Go**.
3. Sign in with the username and password for your NetApp account.
4. If a Caution/MustRead statement appears, read it and select the checkbox.



You must apply any required hotfixes after you install the StorageGRID release. For more information, see the [hotfix procedure in the recovery and maintenance instructions](#)

5. Read the End User License Agreement, select the checkbox, and then select **Accept & Continue**.

The downloads page for the version you selected appears. The page contains three columns:

6. In the **Install StorageGRID** column, select the .tgz or .zip file for Ubuntu or Debian.



Select the .zip file if you are running Windows on the service laptop.

7. Save and extract the archive file.
8. Choose the files you need from the following list.

The set of files you need depends on your planned grid topology and how you will deploy your StorageGRID grid.



The paths listed in the table are relative to the top-level directory installed by the extracted installation archive.

Path and file name	Description
<code>./debs/README</code>	A text file that describes all of the files contained in the StorageGRID download file.
<code>./debs/NLF000000.txt</code>	A non-production NetApp License File that you can use for testing and proof of concept deployments.
<code>./debs/storagegrid-webscale-images-version-SHA.deb</code>	DEB package for installing the StorageGRID node images on Ubuntu or Debian hosts.
<code>./debs/storagegrid-webscale-images-version-SHA.deb.md5</code>	MD5 checksum for the file <code>./debs/storagegrid-webscale-images-version-SHA.deb</code> .
<code>./debs/storagegrid-webscale-service-version-SHA.deb</code>	DEB package for installing the StorageGRID host service on Ubuntu or Debian hosts.
Deployment scripting tool	Description
<code>./debs/configure-storagegrid.py</code>	A Python script used to automate the configuration of a StorageGRID system.
<code>./debs/configure-sga.py</code>	A Python script used to automate the configuration of StorageGRID appliances.
<code>./debs/storagegrid-ssoauth.py</code>	An example Python script that you can use to sign in to the Grid Management API when single sign-on is enabled. You can also use this script for Ping Federate.
<code>./debs/configure-storagegrid.sample.json</code>	An example configuration file for use with the <code>configure-storagegrid.py</code> script.
<code>./debs/configure-storagegrid.blank.json</code>	A blank configuration file for use with the <code>configure-storagegrid.py</code> script.
<code>./debs/extras/ansible</code>	Example Ansible role and playbook for configuring Ubuntu or Debian hosts for StorageGRID container deployment. You can customize the role or playbook as necessary.
<code>./debs/storagegrid-ssoauth-azure.py</code>	An example Python script that you can use to sign in to the Grid Management API when single sign-on (SSO) is enabled using Active Directory or Ping Federate.
<code>./debs/storagegrid-ssoauth-azure.js</code>	A helper script called by the companion <code>storagegrid-ssoauth-azure.py</code> Python script to perform SSO interactions with Azure.

Path and file name	Description
./debs/extras/api-schemas	<p>API schemas for StorageGRID.</p> <p>Note: Before you perform an upgrade, you can use these schemas to confirm that any code you have written to use StorageGRID management APIs will be compatible with the new StorageGRID release if you don't have a non-production StorageGRID environment for upgrade compatibility testing.</p>

Software requirements for Ubuntu and Debian

You can use a virtual machine to host any type of StorageGRID node. You need one virtual machine for each grid node.

To install StorageGRID on Ubuntu or Debian, you must install some third-party software packages. Some supported Linux distributions don't contain these packages by default. The software package versions that StorageGRID installations are tested on include those listed on this page.



If you select a Linux distribution and container runtime installation option that requires any of these packages, and they are not installed automatically by the Linux distribution, install one of the versions listed here if available from your provider or the supporting vendor for your Linux distribution. Otherwise, use the default package versions available from your vendor.



All installation options require either Podman or Docker. Do not install both packages. Install only the package required by your installation option.

Python versions tested

- 3.5.2-2
- 3.6.8-2
- 3.6.8-38
- 3.6.9-1
- 3.7.3-1
- 3.8.10-0
- 3.9.2-1
- 3.9.10-2
- 3.9.16-1
- 3.10.6-1
- 3.11.2-6

Podman versions tested

- 3.2.3-0

- 3.4.4+ds1
- 4.1.1-7
- 4.2.0-11
- 4.3.1+ds1-8+b1
- 4.4.1-8
- 4.4.1-12

Docker versions tested



Docker support is deprecated and will be removed in a future release.

- Docker-CE 20.10.7
- Docker-CE 20.10.20-3
- Docker-CE 23.0.6-1
- Docker-CE 24.0.2-1
- Docker-CE 24.0.4-1
- Docker-CE 24.0.5-1
- Docker-CE 24.0.7-1
- 1.5-2

CPU and RAM requirements

Before installing StorageGRID software, verify and configure the hardware so that it is ready to support the StorageGRID system.

Each StorageGRID node requires the following minimum resources:

- CPU cores: 8 per node
- RAM: At least 24 GB per node, and 2 to 16 GB less than the total system RAM, depending on the total RAM available and the amount of non-StorageGRID software running on the system

Ensure that the number of StorageGRID nodes you plan to run on each physical or virtual host does not exceed the number of CPU cores or the physical RAM available. If the hosts aren't dedicated to running StorageGRID (not recommended), be sure to consider the resource requirements of the other applications.



Monitor your CPU and memory usage regularly to ensure that these resources continue to accommodate your workload. For example, doubling the RAM and CPU allocation for virtual Storage Nodes would provide similar resources to those provided for StorageGRID appliance nodes. Additionally, if the amount of metadata per node exceeds 500 GB, consider increasing the RAM per node to 48 GB or more. For information about managing object metadata storage, increasing the Metadata Reserved Space setting, and monitoring CPU and memory usage, see the instructions for [administering](#), [monitoring](#), and [upgrading](#) StorageGRID.

If hyperthreading is enabled on the underlying physical hosts, you can provide 8 virtual cores (4 physical cores) per node. If hyperthreading is not enabled on the underlying physical hosts, you must provide 8 physical cores per node.

If you are using virtual machines as hosts and have control over the size and number of VMs, you should use a single VM for each StorageGRID node and size the VM accordingly.

For production deployments, you should not run multiple Storage Nodes on the same physical storage hardware or virtual host. Each Storage Node in a single StorageGRID deployment should be in its own isolated failure domain. You can maximize the durability and availability of object data if you ensure that a single hardware failure can only impact a single Storage Node.

See also [Storage and performance requirements](#).

Storage and performance requirements

You must understand the storage requirements for StorageGRID nodes, so you can provide enough space to support the initial configuration and future storage expansion.

StorageGRID nodes require three logical categories of storage:

- **Container pool** — Performance-tier (10K SAS or SSD) storage for the node containers, which will be assigned to the Docker storage driver when you install and configure Docker on the hosts that will support your StorageGRID nodes.
- **System data** — Performance-tier (10K SAS or SSD) storage for per-node persistent storage of system data and transaction logs, which the StorageGRID host services will consume and map into individual nodes.
- **Object data** — Performance-tier (10K SAS or SSD) storage and capacity-tier (NL-SAS/SATA) bulk storage for the persistent storage of object data and object metadata.

You must use RAID-backed block devices for all storage categories. Non-redundant disks, SSDs, or JBODs aren't supported. You can use shared or local RAID storage for any of the storage categories; however, if you want to use the node migration capability in StorageGRID, you must store both system data and object data on shared storage. For more information, see [Node container migration requirements](#).

Performance requirements

The performance of the volumes used for the container pool, system data, and object metadata significantly impacts the overall performance of the system. You should use performance-tier (10K SAS or SSD) storage for these volumes to ensure adequate disk performance in terms of latency, input/output operations per second (IOPS), and throughput. You can use capacity-tier (NL-SAS/SATA) storage for the persistent storage of object data.

The volumes used for the container pool, system data, and object data must have write-back caching enabled. The cache must be on a protected or persistent media.

Requirements for hosts that use NetApp ONTAP storage

If the StorageGRID node uses storage assigned from a NetApp ONTAP system, confirm that the volume does not have a FabricPool tiering policy enabled. Disabling FabricPool tiering for volumes used with StorageGRID nodes simplifies troubleshooting and storage operations.



Never use FabricPool to tier any data related to StorageGRID back to StorageGRID itself. Tiering StorageGRID data back to StorageGRID increases troubleshooting and operational complexity.

Number of hosts required

Each StorageGRID site requires a minimum of three Storage Nodes.



In a production deployment, don't run more than one Storage Node on a single physical or virtual host. Using a dedicated host for each Storage Node provides an isolated failure domain.

Other types of nodes, such as Admin Nodes or Gateway Nodes, can be deployed on the same hosts, or they can be deployed on their own dedicated hosts as required.

Number of storage volumes for each host

The following table shows the number of storage volumes (LUNs) required for each host and the minimum size required for each LUN, based on which nodes will be deployed on that host.

The maximum tested LUN size is 39 TB.



These numbers are for each host, not for the entire grid.

LUN purpose	Storage category	Number of LUNs	Minimum size/LUN
Container engine storage pool	Container pool	1	Total number of nodes × 100 GB
/var/local volume	System data	1 for each node on this host	90 GB
Storage Node	Object data	3 for each Storage Node on this host Note: A software-based Storage Node can have 1 to 16 storage volumes; at least 3 storage volumes are recommended.	12 TB (4 TB/LUN) See Storage requirements for Storage Nodes for more information.
Storage Node (metadata-only)	Object metadata	1	4 TB See Storage requirements for Storage Nodes for more information. Note: Only one rangedb is required for metadata-only Storage Nodes.
Admin Node audit logs	System data	1 for each Admin Node on this host	200 GB
Admin Node tables	System data	1 for each Admin Node on this host	200 GB



Depending on the audit level configured, the size of user inputs such as S3 object key name, and how much audit log data you need to preserve, you might need to increase the size of the audit log LUN on each Admin Node. Generally, a grid generates approximately 1 KB of audit data per S3 operation, which would mean that a 200 GB LUN would support 70 million operations per day or 800 operations per second for two to three days.

Minimum storage space for a host

The following table shows the minimum storage space required for each type of node. You can use this table to determine the minimum amount of storage you must provide to the host in each storage category, based on which nodes will be deployed on that host.



Disk snapshots can't be used to restore grid nodes. Instead, refer to the [grid node recovery](#) procedures for each type of node.

Type of node	Container pool	System data	Object data
Storage Node	100 GB	90 GB	4,000 GB
Admin Node	100 GB	490 GB (3 LUNs)	<i>not applicable</i>
Gateway Node	100 GB	90 GB	<i>not applicable</i>
Archive Node	100 GB	90 GB	<i>not applicable</i>

Example: Calculating the storage requirements for a host

Suppose you plan to deploy three nodes on the same host: one Storage Node, one Admin Node, and one Gateway Node. You should provide a minimum of nine storage volumes to the host. You will need a minimum of 300 GB of performance-tier storage for the node containers, 670 GB of performance-tier storage for system data and transaction logs, and 12 TB of capacity-tier storage for object data.

Type of node	LUN purpose	Number of LUNs	LUN size
Storage Node	Docker storage pool	1	300 GB (100 GB/node)
Storage Node	<code>/var/local</code> volume	1	90 GB
Storage Node	Object data	3	12 TB (4 TB/LUN)
Admin Node	<code>/var/local</code> volume	1	90 GB
Admin Node	Admin Node audit logs	1	200 GB
Admin Node	Admin Node tables	1	200 GB
Gateway Node	<code>/var/local</code> volume	1	90 GB

Type of node	LUN purpose	Number of LUNs	LUN size
Total		9	Container pool: 300 GB System data: 670 GB Object data: 12,000 GB

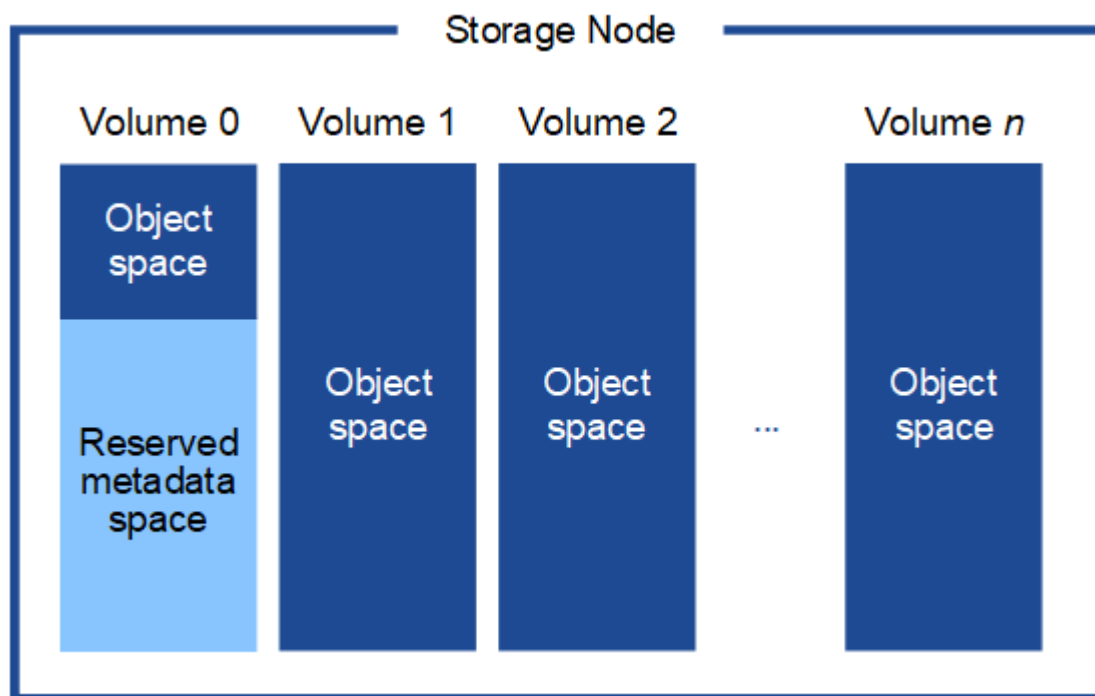
Storage requirements for Storage Nodes

A software-based Storage Node can have 1 to 16 storage volumes—3 or more storage volumes are recommended. Each storage volume should be 4 TB or larger.



An appliance Storage Node can have up to 48 storage volumes.

As shown in the figure, StorageGRID reserves space for object metadata on storage volume 0 of each Storage Node. Any remaining space on storage volume 0 and any other storage volumes in the Storage Node are used exclusively for object data.



To provide redundancy and to protect object metadata from loss, StorageGRID stores three copies of the metadata for all objects in the system at each site. The three copies of object metadata are evenly distributed across all Storage Nodes at each site.

When installing a grid with metadata-only Storage Nodes, the grid must also contain a minimum number of nodes for object storage. See [Types of Storage Nodes](#) for more information about metadata-only Storage Nodes.

- For a single-site grid, at least two Storage Nodes are configured for objects and metadata.
- For a multi-site grid, at least one Storage Node per site are configured for objects and metadata.

When you assign space to volume 0 of a new Storage Node, you must ensure there is adequate space for that

node's portion of all object metadata.

- At a minimum, you must assign at least 4 TB to volume 0.



If you use only one storage volume for a Storage Node and you assign 4 TB or less to the volume, the Storage Node might enter the Storage Read-Only state on startup and store object metadata only.



If you assign less than 500 GB to volume 0 (non-production use only), 10% of the storage volume's capacity is reserved for metadata.

- If you are installing a new system (StorageGRID 11.6 or higher) and each Storage Node has 128 GB or more of RAM, assign 8 TB or more to volume 0. Using a larger value for volume 0 can increase the space allowed for metadata on each Storage Node.
- When configuring different Storage Nodes for a site, use the same setting for volume 0 if possible. If a site contains Storage Nodes of different sizes, the Storage Node with the smallest volume 0 will determine the metadata capacity of that site.

For details, go to [Manage object metadata storage](#).

Node container migration requirements

The node migration feature allows you to manually move a node from one host to another. Typically, both hosts are in the same physical data center.

Node migration allows you to perform physical host maintenance without disrupting grid operations. You move all StorageGRID nodes, one at a time, to another host before taking the physical host offline. Migrating nodes requires only a short downtime for each node and should not affect operation or availability of grid services.

If you want to use the StorageGRID node migration feature, your deployment must meet additional requirements:

- Consistent network interface names across hosts in a single physical data center
- Shared storage for StorageGRID metadata and object repository volumes that is accessible by all hosts in a single physical data center. For example, you might use NetApp E-Series storage arrays.

If you are using virtual hosts and the underlying hypervisor layer supports VM migration, you might want to use this capability instead of the node migration feature in StorageGRID. In this case, you can ignore these additional requirements.

Before performing migration or hypervisor maintenance, shut down the nodes gracefully. See the instructions for [shutting down a grid node](#).

VMware Live Migration not supported

When performing bare-metal installation on VMware VMs, OpenStack Live Migration and VMware live vMotion cause the virtual machine clock time to jump and aren't supported for grid nodes of any type. Though rare, incorrect clock times can result in loss of data or configuration updates.

Cold migration is supported. In cold migration, you shut down the StorageGRID nodes before migrating them between hosts. See the instructions for [shutting down a grid node](#).

Consistent network interface names

To move a node from one host to another, the StorageGRID host service needs to have some confidence that the external network connectivity the node has at its current location can be duplicated at the new location. It gets this confidence through the use of consistent network interface names in the hosts.

Suppose, for example, that StorageGRID NodeA running on Host1 has been configured with the following interface mappings:

eth0 → **bond0.1001**

eth1 → **bond0.1002**

eth2 → **bond0.1003**

The lefthand side of the arrows corresponds to the traditional interfaces as viewed from within a StorageGRID container (that is, the Grid, Admin, and Client Network interfaces, respectively). The righthand side of the arrows corresponds to the actual host interfaces providing these networks, which are three VLAN interfaces subordinate to the same physical interface bond.

Now, suppose you want to migrate NodeA to Host2. If Host2 also has interfaces named bond0.1001, bond0.1002, and bond0.1003, the system will allow the move, assuming that the like-named interfaces will provide the same connectivity on Host2 as they do on Host1. If Host2 does not have interfaces with the same names, the move will not be allowed.

There are many ways to achieve consistent network interface naming across multiple hosts; see [Configure the host network](#) for some examples.

Shared storage

To achieve rapid, low-overhead node migrations, the StorageGRID node migration feature does not physically move node data. Instead, node migration is performed as a pair of export and import operations, as follows:

Steps

1. During the "node export" operation, a small amount of persistent state data is extracted from the node container running on HostA and cached on that node's system data volume. Then, the node container on HostA is deinstantiated.
2. During the "node import" operation, the node container on HostB that uses the same network interface and block storage mappings that were in effect on HostA is instantiated. Then, the cached persistent state data is inserted into the new instance.

Given this mode of operation, all of the node's system data and object storage volumes must be accessible from both HostA and HostB for the migration to be allowed, and to work. In addition, they must have been mapped into the node using names that are guaranteed to refer to the same LUNs on HostA and HostB.

The following example shows one solution for block device mapping for a StorageGRID Storage Node, where DM multipathing is in use on the hosts, and the alias field has been used in `/etc/multipath.conf` to provide consistent, friendly block device names available on all hosts.

`/var/local` → `/dev/mapper/sgws-sn1-var-local`
`rangedb0` → `/dev/mapper/sgws-sn1-rangedb0`
`rangedb1` → `/dev/mapper/sgws-sn1-rangedb1`
`rangedb2` → `/dev/mapper/sgws-sn1-rangedb2`
`rangedb3` → `/dev/mapper/sgws-sn1-rangedb3`

Prepare the hosts (Ubuntu or Debian)

How host-wide settings change during installation

On bare metal systems, StorageGRID makes some changes to host-wide `sysctl` settings.

The following changes are made:

```
# 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 RAM
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
```

Install Linux

You must install StorageGRID on all Ubuntu or Debian grid hosts. For a list of supported versions, use the NetApp Interoperability Matrix Tool.



Ensure that your operating system is upgraded to Linux kernel 4.15 or higher.

Steps

1. Install Linux on all physical or virtual grid hosts according to the distributor's instructions or your standard procedure.



Don't install any graphical desktop environments. When installing Ubuntu, you must select **standard system utilities**. Selecting **OpenSSH server** is recommended to enable ssh access to your Ubuntu hosts. All other options can remain cleared.

2. Ensure that all hosts have access to Ubuntu or Debian package repositories.
3. If swap is enabled:
 - a. Run the following command: `$ sudo swapoff --all`
 - b. Remove all swap entries from `/etc/fstab` to persist the settings.



Failing to disable swap entirely can severely lower performance.

Understand AppArmor profile installation

If you are operating in a self-deployed Ubuntu environment and using the AppArmor mandatory access control system, the AppArmor profiles associated with packages you install on the base system might be blocked by the corresponding packages installed with StorageGRID.

By default, AppArmor profiles are installed for packages that you install on the base operating system. When you run these packages from the StorageGRID system container, the AppArmor profiles are blocked. The DHCP, MySQL, NTP, and tcdump base packages conflict with AppArmor, and other base packages might also conflict.

You have two choices for handling AppArmor profiles:

- Disable individual profiles for the packages installed on the base system that overlap with the packages in the StorageGRID system container. When you disable individual profiles, an entry appears in the StorageGRID log files indicating that AppArmor is enabled.

Use the following commands:


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

Example:

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

- Disable AppArmor altogether. For Ubuntu 9.10 or later, follow the instructions in the Ubuntu online community: [Disable AppArmor](#). Disabling AppArmor altogether might not be possible on newer Ubuntu versions.

After you disable AppArmor, no entries indicating that AppArmor is enabled will appear in the StorageGRID log files.

Configure the host network (Ubuntu or Debian)

After completing the Linux installation on your hosts, you might need to perform some additional configuration to prepare a set of network interfaces on each host that are suitable for mapping into the StorageGRID nodes you will deploy later.

Before you begin

- You have reviewed the [StorageGRID networking guidelines](#).
- You have reviewed the information about [node container migration requirements](#).
- If you are using virtual hosts, you have read the [considerations and recommendations for MAC address cloning](#) before configuring the host network.



If you are using VMs as hosts, you should select VMXNET 3 as the virtual network adapter. The VMware E1000 network adapter has caused connectivity issues with StorageGRID containers deployed on certain distributions of Linux.

About this task

Grid nodes must be able to access the Grid Network and, optionally, the Admin and Client Networks. You provide this access by creating mappings that associate the host's physical interface to the virtual interfaces for each grid node. When creating host interfaces, use friendly names to facilitate deployment across all hosts, and to enable migration.

The same interface can be shared between the host and one or more nodes. For example, you might use the same interface for host access and node Admin Network access, to facilitate host and node maintenance. Although the same interface can be shared between the host and individual nodes, all must have different IP addresses. IP addresses can't be shared between nodes or between the host and any node.

You can use the same host network interface to provide the Grid Network interface for all StorageGRID nodes on the host; you can use a different host network interface for each node; or you can do something in between. However, you would not typically provide the same host network interface as both the Grid and Admin Network interfaces for a single node, or as the Grid Network interface for one node and the Client Network interface for another.

You can complete this task in many ways. For example, if your hosts are virtual machines and you are deploying one or two StorageGRID nodes for each host, you can create the correct number of network interfaces in the hypervisor, and use a 1-to-1 mapping. If you are deploying multiple nodes on bare metal hosts for production use, you can leverage the Linux networking stack's support for VLAN and LACP for fault tolerance and bandwidth sharing. The following sections provide detailed approaches for both of these examples. You don't need to use either of these examples; you can use any approach that meets your needs.



Don't use bond or bridge devices directly as the container network interface. Doing so could prevent node start-up caused by a kernel issue with the use of MACVLAN with bond and bridge devices in the container namespace. Instead, use a non-bond device, such as a VLAN or virtual Ethernet (veth) pair. Specify this device as the network interface in the node configuration file.

Considerations and recommendations for MAC address cloning

MAC address cloning causes the container to use the MAC address of the host, and the host to use the MAC address of either an address you specify or a randomly generated one. You should use MAC address cloning to avoid the use of promiscuous mode network configurations.

Enabling MAC cloning

In certain environments, security can be enhanced through MAC address cloning because it enables you to use a dedicated virtual NIC for the Admin Network, Grid Network, and Client Network. Having the container use the MAC address of the dedicated NIC on the host allows you to avoid using promiscuous mode network configurations.



MAC address cloning is intended to be used with virtual server installations and might not function properly with all physical appliance configurations.



If a node fails to start due to a MAC cloning targeted interface being busy, you might need to set the link to "down" before starting node. Additionally, it is possible that the virtual environment might prevent MAC cloning on a network interface while the link is up. If a node fails to set the MAC address and start due to an interface being busy, setting the link to "down" before starting the node might fix the issue.

MAC address cloning is disabled by default and must be set by node configuration keys. You should enable it when you install StorageGRID.

There is one key for each network:

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

Setting the key to "true" causes the container to use the MAC address of the host's NIC. Additionally, the host will then use the MAC address of the specified container network. By default, the container address is a randomly generated address, but if you have set one using the `_NETWORK_MAC` node configuration key, that address is used instead. The host and container will always have different MAC addresses.



Enabling MAC cloning on a virtual host without also enabling promiscuous mode on the hypervisor might cause Linux host networking using the host's interface to stop working.

MAC cloning use cases

There are two use cases to consider with MAC cloning:

- **MAC cloning not enabled:** When the `_CLONE_MAC` key in the node configuration file is not set, or set to "false," the host will use the host NIC MAC and the container will have a StorageGRID-generated MAC unless a MAC is specified in the `_NETWORK_MAC` key. If an address is set in the `_NETWORK_MAC` key, the container will have the address specified in the `_NETWORK_MAC` key. This configuration of keys requires the use of promiscuous mode.
- **MAC cloning enabled:** When the `_CLONE_MAC` key in the node configuration file is set to "true," the container uses the host NIC MAC, and the host uses a StorageGRID-generated MAC unless a MAC is specified in the `_NETWORK_MAC` key. If an address is set in the `_NETWORK_MAC` key, the host uses the specified address instead of a generated one. In this configuration of keys, you should not use promiscuous mode.



If you don't want to use MAC address cloning and would rather allow all interfaces to receive and transmit data for MAC addresses other than the ones assigned by the hypervisor, ensure that the security properties at the virtual switch and port group levels are set to **Accept** for Promiscuous Mode, MAC Address Changes, and Forged Transmits. The values set on the virtual switch can be overridden by the values at the port group level, so ensure that settings are the same in both places.

To enable MAC cloning, see the [instructions for creating node configuration files](#).

MAC cloning example

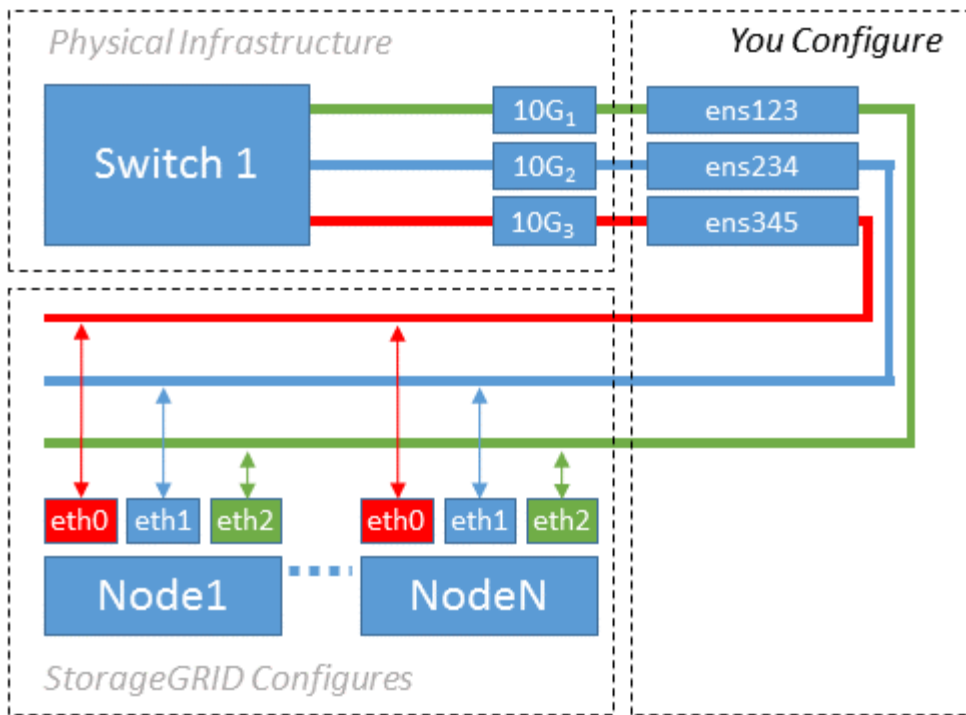
Example of MAC cloning enabled with a host having MAC address of 11:22:33:44:55:66 for the interface `ens256` and the following keys in the node configuration file:

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

Result: the host MAC for `ens256` is `b2:9c:02:c2:27:10` and the Admin Network MAC is `11:22:33:44:55:66`

Example 1: 1-to-1 mapping to physical or virtual NICs

Example 1 describes a simple physical interface mapping that requires little or no host-side configuration.



The Linux operating system creates the `ensXYZ` interfaces automatically during installation or boot, or when the interfaces are hot-added. No configuration is required other than ensuring that the interfaces are set to come up automatically after boot. You do have to determine which `ensXYZ` corresponds to which StorageGRID network (Grid, Admin, or Client) so you can provide the correct mappings later in the configuration process.

Note that the figure shows multiple StorageGRID nodes; however, you would normally use this configuration for single-node VMs.

If Switch 1 is a physical switch, you should configure the ports connected to interfaces $10G_1$ through $10G_3$ for access mode, and place them on the appropriate VLANs.

Example 2: LACP bond carrying VLANs

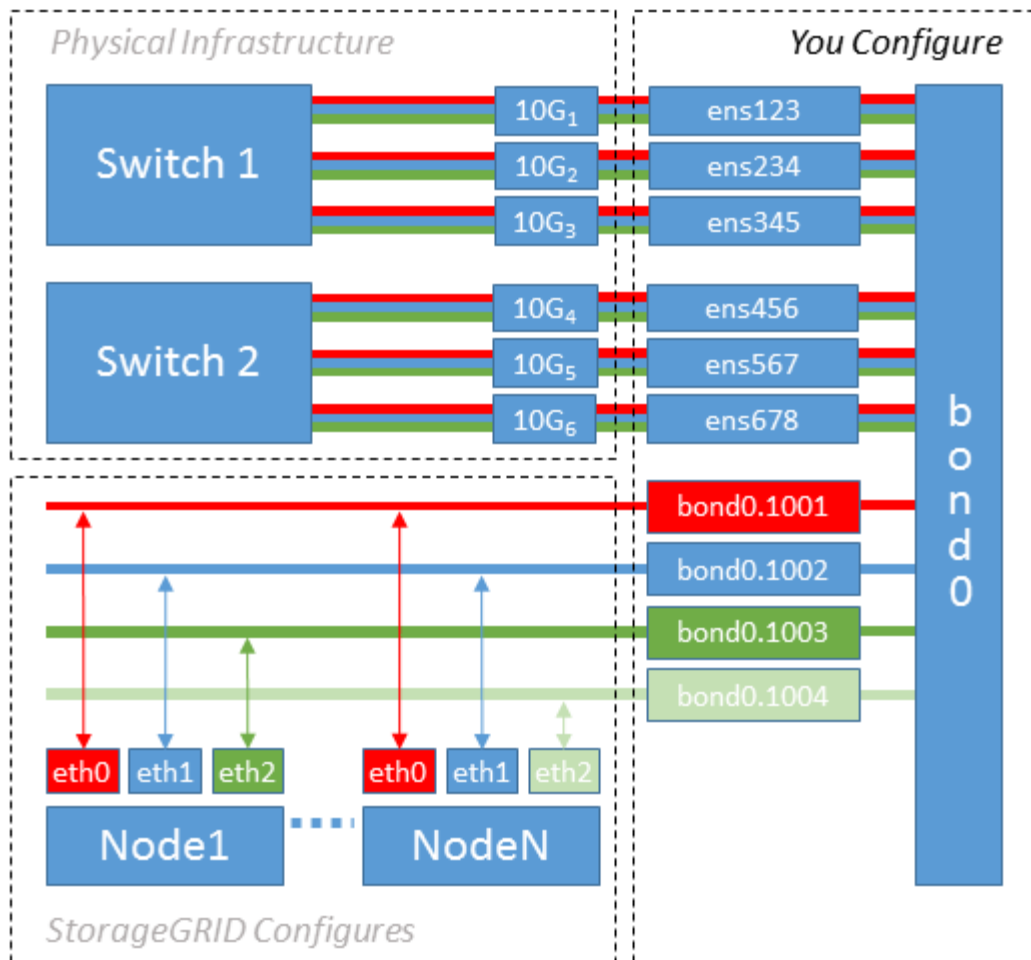
Example 2 assumes you are familiar with bonding network interfaces and with creating VLAN interfaces on the Linux distribution you are using.

About this task

Example 2 describes a generic, flexible, VLAN-based scheme that facilitates the sharing of all available network bandwidth across all nodes on a single host. This example is particularly applicable to bare metal hosts.

To understand this example, suppose you have three separate subnets for the Grid, Admin, and Client Networks at each data center. The subnets are on separate VLANs (1001, 1002, and 1003) and are presented to the host on a LACP-bonded trunk port (`bond0`). You would configure three VLAN interfaces on the bond: `bond0.1001`, `bond0.1002`, and `bond0.1003`.

If you require separate VLANs and subnets for node networks on the same host, you can add VLAN interfaces on the bond and map them into the host (shown as `bond0.1004` in the illustration).



Steps

1. Aggregate all physical network interfaces that will be used for StorageGRID network connectivity into a single LACP bond.

Use the same name for the bond on every host, for example, bond0.

2. Create VLAN interfaces that use this bond as their associated "physical device," using the standard VLAN interface naming convention `physdev-name.VLAN ID`.

Note that steps 1 and 2 require appropriate configuration on the edge switches terminating the other ends of the network links. The edge switch ports must also be aggregated into a LACP port channel, configured as a trunk, and allowed to pass all required VLANs.

Example interface configuration files for this per-host networking configuration scheme are provided.

Related information

[Example /etc/network/interfaces](#)

Configure host storage

You must allocate block storage volumes to each host.

Before you begin

You have reviewed the following topics, which provide information you need to accomplish this task:

Storage and performance requirements

Node container migration requirements

About this task

When allocating block storage volumes (LUNs) to hosts, use the tables in "Storage requirements" to determine the following:

- Number of volumes required for each host (based on the number and types of nodes that will be deployed on that host)
- Storage category for each volume (that is, System Data or Object Data)
- Size of each volume

You will use this information as well as the persistent name assigned by Linux to each physical volume when you deploy StorageGRID nodes on the host.



You don't need to partition, format, or mount any of these volumes; you just need to ensure they are visible to the hosts.



Only one object-data LUN is required for metadata-only Storage Nodes.

Avoid using "raw" special device files (`/dev/sdb`, for example) as you compose your list of volume names. These files can change across reboots of the host, which will impact proper operation of the system. If you are using iSCSI LUNs and Device Mapper Multipathing, consider using multipath aliases in the `/dev/mapper` directory, especially if your SAN topology includes redundant network paths to the shared storage. Alternatively, you can use the system-created softlinks under `/dev/disk/by-path/` for your persistent device names.

For example:

```
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
```

Results will differ for each installation.

Assign friendly names to each of these block storage volumes to simplify the initial StorageGRID installation and future maintenance procedures. If you are using the device mapper multipath driver for redundant access to shared storage volumes, you can use the `alias` field in your `/etc/multipath.conf` file.

For example:

```
multipaths {
    multipath {
        wwid 3600a09800059d6df00005df2573c2c30
        alias docker-storage-volume-hostA
    }
    multipath {
        wwid 3600a09800059d6df00005df3573c2c30
        alias sgws-adml-var-local
    }
    multipath {
        wwid 3600a09800059d6df00005df4573c2c30
        alias sgws-adml-audit-logs
    }
    multipath {
        wwid 3600a09800059d6df00005df5573c2c30
        alias sgws-adml-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
    }
    ...
}
```

This will cause the aliases to appear as block devices in the `/dev/mapper` directory on the host, allowing you to specify a friendly, easily-validated name whenever a configuration or maintenance operation requires specifying a block storage volume.



If you are setting up shared storage to support StorageGRID node migration and using Device Mapper Multipathing, you can create and install a common `/etc/multipath.conf` on all co-located hosts. Just make sure to use a different Docker storage volume on each host. Using aliases and including the target hostname in the alias for each Docker storage volume LUN will make this easy to remember and is recommended.

Related information

[Storage and performance requirements](#)

[Node container migration requirements](#)

Configure the Docker storage volume

Before installing Docker, you might need to format the Docker storage volume and mount it on `/var/lib/docker`.

About this task

You can skip these steps if you plan to use local storage for the Docker storage volume and have sufficient space available on the host partition containing `/var/lib`.

Steps

1. Create a file system on the Docker storage volume:

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

2. Mount the Docker storage volume:

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

3. Add an entry for `docker-storage-volume-device` to `/etc/fstab`.

This step ensures that the storage volume will remount automatically after host reboots.

Install Docker

The StorageGRID system runs on Linux as a collection of Docker containers. Before you can install StorageGRID, you must install Docker.

Steps

1. Install Docker by following the instructions for your Linux distribution.



If Docker is not included with your Linux distribution, you can download it from the Docker website.

2. Ensure Docker has been enabled and started by running the following two commands:

```
sudo systemctl enable docker
```

```
sudo systemctl start docker
```


3. Confirm you have installed the expected version of Docker by entering the following:

```
sudo docker version
```

The Client and Server versions must be 1.11.0 or later.

Related information

[Configure host storage](#)

Install StorageGRID host services

You use the StorageGRID DEB package to install the StorageGRID host services.

About this task

These instructions describe how to install the host services from the DEB packages. As an alternative, you can use the APT repository metadata included in the installation archive to install the DEB packages remotely. See the APT repository instructions for your Linux operating system.

Steps

1. Copy the StorageGRID DEB packages to each of your hosts, or make them available on shared storage.

For example, place them in the `/tmp` directory, so you can use the example command in the next step.

2. Log in to each host as root or using an account with sudo permission, and run the following commands.

You must install the `images` package first, and the `service` package second. If you placed the packages in a directory other than `/tmp`, modify the command to reflect the path you used.

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

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



Python 2.7 must already be installed before the StorageGRID packages can be installed. The `sudo dpkg --install /tmp/storagegrid-webscale-images-version-SHA.deb` command will fail until you have done so.

Copyright information

Copyright © 2024 NetApp, Inc. All Rights Reserved. Printed in the U.S. No part of this document covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—without prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP “AS IS” AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

LIMITED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (b)(3) of the Rights in Technical Data -Noncommercial Items at DFARS 252.227-7013 (FEB 2014) and FAR 52.227-19 (DEC 2007).

Data contained herein pertains to a commercial product and/or commercial service (as defined in FAR 2.101) and is proprietary to NetApp, Inc. All NetApp technical data and computer software provided under this Agreement is commercial in nature and developed solely at private expense. The U.S. Government has a non-exclusive, non-transferrable, nonsublicensable, worldwide, limited irrevocable license to use the Data only in connection with and in support of the U.S. Government contract under which the Data was delivered. Except as provided herein, the Data may not be used, disclosed, reproduced, modified, performed, or displayed without the prior written approval of NetApp, Inc. United States Government license rights for the Department of Defense are limited to those rights identified in DFARS clause 252.227-7015(b) (FEB 2014).

Trademark information

NETAPP, the NETAPP logo, and the marks listed at <http://www.netapp.com/TM> are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners.