호스트(Linux) 준비

StorageGRID software

NetApp
July 23, 2026

목차

호스트(Linux) 준비

StorageGRID에서 호스트 전체 설정을 변경하는 방법

베어메탈 시스템에서 StorageGRID는 호스트 전체 `sysctl` 설정에 일부 변경 사항을 적용합니다.

> "Linux"는 RHEL, Ubuntu 또는 Debian 배포판을 의미합니다. 지원되는 버전 목록은 "NetApp 상호 운용성 매트릭스 툴(IMT)"을 참조하십시오.

다음과 같은 변경 사항이 적용됩니다.

```
# 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
```

Linux 그리드 호스트에 StorageGRID 설치

모든 Linux 그리드 호스트에 StorageGRID를 설치해야 합니다. 지원되는 버전 목록은 NetApp

상호 운용성 매트릭스 도구를 참조하십시오.

시작하기 전에

운영 체제가 아래에 명시된 StorageGRID의 최소 커널 버전 요구 사항을 충족하는지 확인하십시오. `uname -r` 명령을 사용하여 운영 체제의 커널 버전을 확인하거나 운영 체제 공급업체에 문의하십시오.

"Linux"는 RHEL, Ubuntu 또는 Debian 배포판을 의미합니다. 지원되는 버전 목록은 "NetApp 상호 운용성 매트릭스 툴(IMT)"을 참조하십시오.

RHEL

RHEL 버전	최소 커널 버전	커널 패키지 이름
8.8 (사용 중단됨)	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 (사용 중단됨)	5.14.0-70.22.1.el9_0.x86_64	kernel-5.14.0-70.22.1.el9_0.x86_64
9.2 (사용 중단됨)	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

참고**:** Ubuntu 18.04 및 20.04 버전에 대한 지원은 더 이상 사용되지 않으며 향후 릴리스에서 제거될 예정입니다.

Ubuntu 버전	최소 커널 버전	커널 패키지 이름
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,now 6.8.0-31.31

Debian

참고**:** Debian 버전 11 지원은 더 이상 사용되지 않으며 향후 릴리스에서 제거될 예정입니다.

Debian 버전	최소 커널 버전	커널 패키지 이름
11(지원 중단됨)	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

단계

1. 총판의 지침 또는 표준 절차에 따라 모든 물리적 또는 가상 그리드 호스트에 Linux를 설치하십시오.

   그래픽 데스크톱 환경을 설치하지 마십시오.

   - RHEL을 설치할 때 표준 Linux 설치 프로그램을 사용하는 경우, 가능하다면 "컴퓨팅 노드" 소프트웨어 구성을 선택하거나 "최소 설치" 기본 환경을 선택하십시오.

◦ 우분투를 설치할 때 *표준 시스템 유틸리티*를 선택해야 합니다. 우분투 호스트에 SSH로 접속하려면 *OpenSSH 서버*를 선택하는 것이 좋습니다. 나머지 옵션은 모두 선택 해제된 상태로 두어도 됩니다.

2. 모든 호스트가 RHEL용 Extras 채널을 포함한 패키지 저장소에 액세스할 수 있는지 확인하십시오.
3. 스왑 기능이 활성화된 경우:
   a. 다음 명령을 실행합니다. `$ sudo swapoff --all`
   b. 설정을 유지하려면 `/etc/fstab`에서 모든 스왑 항목을 제거하십시오.

> ⓘ 스왑 기능을 완전히 비활성화하지 않으면 성능이 심각하게 저하될 수 있습니다.

Ubuntu 및 Debian에서 StorageGRID용 AppArmor 프로필에 대해 알아보십시오

자체 배포된 Ubuntu 환경에서 AppArmor 강제 액세스 제어 시스템을 사용하는 경우, 기본 시스템에 설치한 패키지와 연결된 AppArmor 프로필이 StorageGRID와 함께 설치된 해당 패키지에 의해 차단될 수 있습니다.

기본적으로 AppArmor 프로필은 기본 운영 체제에 설치하는 패키지에 대해 설치됩니다. StorageGRID 시스템 컨테이너에서 이러한 패키지를 실행하면 AppArmor 프로필이 차단됩니다. DHCP, MySQL, NTP 및 tcdump 기본 패키지는 AppArmor와 충돌하며, 다른 기본 패키지도 충돌할 수 있습니다.

AppArmor 프로필을 처리하는 방법에는 두 가지가 있습니다.

- StorageGRID 시스템 컨테이너의 패키지와 겹치는 기본 시스템에 설치된 패키지에 대한 개별 프로필을 비활성화합니다. 개별 프로필을 비활성화하면 StorageGRID 로그 파일에 AppArmor가 활성화되었다는 항목이 나타납니다.

  다음 명령을 사용합니다.

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

  예:

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

- AppArmor를 완전히 비활성화하세요. Ubuntu 9.10 이상 버전의 경우 Ubuntu 온라인 커뮤니티의 지침을 따르세요: "AppArmor 비활성화". 최신 Ubuntu 버전에서는 AppArmor를 완전히 비활성화하는 것이 불가능할 수 있습니다.

  AppArmor를 비활성화하면 AppArmor가 활성화되었음을 나타내는 항목이 StorageGRID 로그 파일에 더 이상 나타나지 않습니다.

Linux 배포를 위한 StorageGRID 호스트 네트워크 구성

호스트에 Linux 설치를 완료한 후에 나중에 배포할 StorageGRID 노드에 매핑하기에 적합한 네트워크 인터페이스 세트를 각 호스트에서 준비하기 위해 추가 구성을 수행해야 할 수 있습니다.

"Linux"는 RHEL, Ubuntu 또는 Debian 배포판을 의미합니다. 지원되는 버전 목록은 "NetApp 상호 운용성 매트릭스 툴(IMT)"을 참조하십시오.

시작하기 전에

- "StorageGRID 네트워킹 가이드라인"를 검토했습니다.
- "노드 컨테이너 마이그레이션 요구 사항"에 대한 정보를 검토하셨습니다.
- 가상 호스트를 사용하는 경우 호스트 네트워크를 구성하기 전에 MAC 주소 복제에 대한 고려 사항 및 권장 사항을 읽어보시기 바랍니다.

가상 머신을 호스트로 사용하는 경우 가상 네트워크 어댑터로 VMXNET 3을 선택해야 합니다. VMware E1000 네트워크 어댑터는 특정 Linux 배포판에 배포된 StorageGRID 컨테이너에서 연결 문제를 일으킨 것으로 알려져 있습니다.

이 작업 정보

그리드 노드는 그리드 네트워크에 액세스할 수 있어야 하며, 선택적으로 관리 네트워크 및 클라이언트 네트워크에도 액세스할 수 있어야 합니다. 이러한 액세스는 호스트의 물리적 인터페이스를 각 그리드 노드의 가상 인터페이스에 연결하는 매핑을 생성하여 제공합니다. 호스트 인터페이스를 생성할 때는 모든 호스트에 걸쳐 배포를 용이하게 하고 마이그레이션을 지원하기 위해 친숙한 이름을 사용하십시오.

호스트와 하나 이상의 노드 간에 동일한 인터페이스를 공유할 수 있습니다. 예를 들어, 호스트 및 노드 유지 관리를 용이하게 하기 위해 호스트 액세스와 노드 관리 네트워크 액세스에 동일한 인터페이스를 사용할 수 있습니다. 호스트와 개별 노드 간에 동일한 인터페이스를 공유할 수 있지만, 모든 노드는 서로 다른 IP 주소를 가져야 합니다. 노드 간 또는 호스트와 노드 간에 IP 주소를 공유할 수 없습니다.

호스트의 모든 StorageGRID 노드에 그리드 네트워크 인터페이스를 제공하기 위해 동일한 호스트 네트워크 인터페이스를 사용할 수도 있고, 각 노드마다 다른 호스트 네트워크 인터페이스를 사용할 수도 있으며, 그 중간 방식을 사용할 수도 있습니다. 하지만 일반적으로 단일 노드에 대해 그리드 네트워크 인터페이스와 관리 네트워크 인터페이스로 동일한 호스트 네트워크 인터페이스를 제공하거나, 한 노드에는 그리드 네트워크 인터페이스를, 다른 노드에는 클라이언트 네트워크 인터페이스를 제공하지는 않습니다.

이 작업은 여러 가지 방법으로 완료할 수 있습니다. 예를 들어 호스트가 가상 머신이고 각 호스트에 StorageGRID 노드를 하나 또는 두 개 배포하는 경우 하이퍼바이저에서 올바른 수의 네트워크 인터페이스를 생성하고 1:1 매핑을 사용할 수 있습니다. 프로덕션 환경에서 베어메탈 호스트에 여러 노드를 배포하는 경우 Linux 네트워킹 스택의 VLAN 및 LACP 지원 기능을 활용하여 내결함성과 대역폭 공유를 구현할 수 있습니다. 다음 섹션에서는 이 두 가지 예시에 대한 자세한 접근 방식을 제공합니다. 이러한 예시 중 어느 것도 반드시 사용해야 하는 것은 아니며, 필요에 맞는 다른 접근 방식을 사용할 수 있습니다.

본드 또는 브리지 장치를 컨테이너 네트워크 인터페이스로 직접 사용하지 마십시오. 이렇게 하면 컨테이너 네임스페이스에서 본드 및 브리지 장치와 함께 MACVLAN을 사용할 때 발생하는 커널 문제로 인해 노드 시작이 중단될 수 있습니다. 대신 VLAN 또는 가상 이더넷(veth) 쌍과 같은 비본드 장치를 사용하십시오. 이 장치를 노드 구성 파일의 네트워크 인터페이스로 지정하십시오.

MAC 주소 복제에 대한 고려 사항 및 권장 사항

MAC 주소 복제를 사용하면 컨테이너는 호스트의 MAC 주소를 사용하고, 호스트는 사용자가 지정한 주소 또는 임의로 생성된 주소의 MAC 주소를 사용하게 됩니다. 무차별 모드 네트워크 구성을 피하려면 MAC 주소 복제를 사용하는 것이 좋습니다.

MAC 복제 활성화

특정 환경에서는 MAC 주소 복제를 통해 보안을 강화할 수 있습니다. 이를 통해 관리 네트워크, 그리드 네트워크 및 클라이언트 네트워크에 각각 전용 가상 NIC를 사용할 수 있기 때문입니다. 컨테이너가 호스트의 전용 NIC의 MAC 주소를 사용하도록 설정하면 무차별 모드 네트워크 구성을 사용하지 않아도 됩니다.

> MAC 주소 복제는 가상 서버 설치 환경에서 사용하도록 설계되었으며, 모든 물리적 어플라이언스 구성에서 제대로 작동하지 않을 수 있습니다.

> MAC 주소 복제를 위해 지정된 인터페이스가 사용 중이어서 노드 시작에 실패하는 경우, 노드를 시작하기 전에 해당 링크를 "down" 상태로 설정해야 할 수 있습니다. 또한, 링크가 활성화된 상태에서 가상 환경이 네트워크 인터페이스의 MAC 주소 복제를 차단할 수 있습니다. 인터페이스 사용 중으로 인해 노드가 MAC 주소를 설정하고 시작하지 못하는 경우, 노드를 시작하기 전에 해당 링크를 "down" 상태로 설정하면 문제가 해결될 수 있습니다.

MAC 주소 복제는 기본적으로 비활성화되어 있으며 노드 구성 키를 통해 설정해야 합니다. StorageGRID를 설치할 때 활성화해야 합니다.

각 네트워크마다 하나의 키가 있습니다.

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

이 키를 "true"로 설정하면 컨테이너가 호스트 NIC의 MAC 주소를 사용하게 됩니다. 또한 호스트는 지정된 컨테이너 네트워크의 MAC 주소를 사용하게 됩니다. 기본적으로 컨테이너 주소는 임의로 생성된 주소이지만, `_NETWORK_MAC` 노드 구성 키를 사용하여 주소를 설정한 경우 해당 주소가 대신 사용됩니다. 호스트와 컨테이너는 항상 서로 다른 MAC 주소를 갖게 됩니다.

> 하이퍼바이저에서 무차별 모드를 활성화하지 않고 가상 호스트에서 MAC 복제를 활성화하면 호스트 인터페이스를 사용하는 Linux 호스트 네트워킹이 작동을 멈출 수 있습니다.

MAC 복제 활용 사례

MAC 복제에는 두 가지 사용 사례가 있습니다.

- MAC 복제 미사용: 노드 구성 파일의 `_CLONE_MAC` 키가 설정되지 않았거나 "false"로 설정된 경우, 호스트는 호스트 NIC MAC 주소를 사용하고 컨테이너는 `_NETWORK_MAC` 키에 MAC 주소가 지정되지 않은 한 StorageGRID에서 생성한 MAC 주소를 사용합니다. `_NETWORK_MAC` 키에 주소가 설정된 경우 컨테이너는 `_NETWORK_MAC` 키에 지정된 주소를 사용합니다. 이러한 키 구성에는 무차별 모드(promiscuous mode) 사용이 필요합니다.
- MAC 복제 활성화: 노드 구성 파일의 `_CLONE_MAC` 키가 "true"로 설정된 경우 컨테이너는 호스트 NIC의 MAC

주소를 사용하고, 호스트는 `_NETWORK_MAC` 키에 MAC 주소가 지정되지 않은 경우 StorageGRID에서 생성한 MAC 주소를 사용합니다. `_NETWORK_MAC` 키에 주소가 설정된 경우 호스트는 생성된 주소 대신 지정된 주소를 사용합니다. 이 키 구성에서는 무차별 모드를 사용해서는 안 됩니다.

> MAC 주소 복제를 사용하지 않고 하이퍼바이저가 할당한 MAC 주소 이외의 MAC 주소에 대해 모든 인터페이스가 데이터를 송수신할 수 있도록 허용하려면 가상 스위치 및 포트 그룹 수준의 보안 속성에서 무차별 모드, MAC 주소 변경 및 위조 전송에 대해 *허용*으로 설정해야 합니다. 가상 스위치에서 설정한 값은 포트 그룹 수준에서 설정한 값으로 재정의될 수 있으므로 두 위치의 설정이 동일한지 확인하십시오.

MAC 복제를 활성화하려면 "노드 구성 파일 생성 지침"을 참조하십시오.

MAC 복제 예시

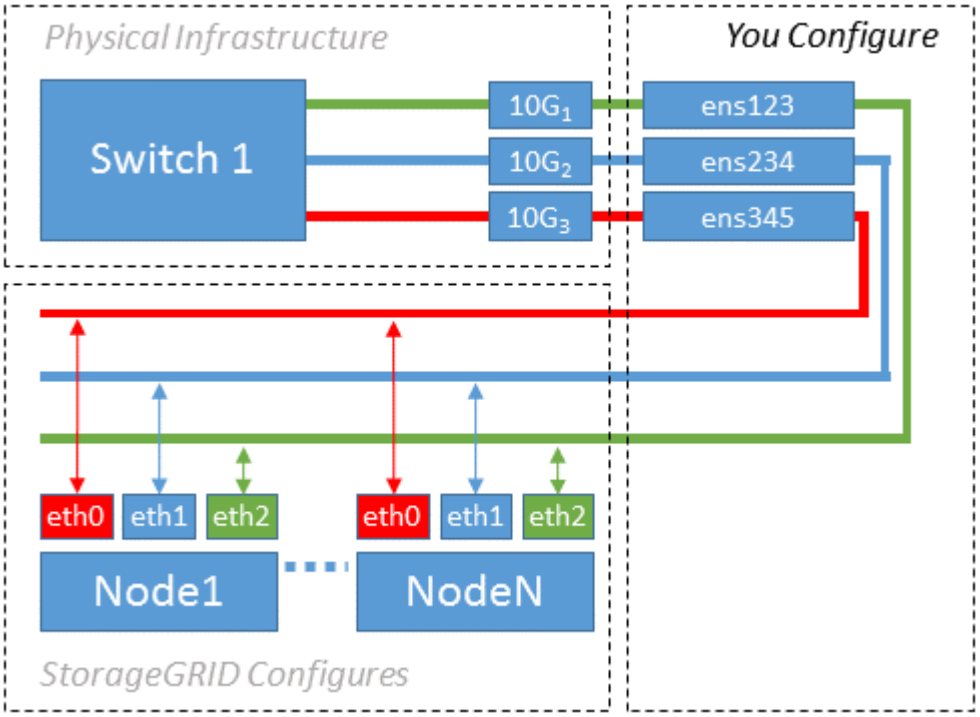
MAC 주소가 11:22:33:44:55:66인 호스트에서 인터페이스 ens256에 대해 MAC 클로닝이 활성화된 예이며, 노드 구성 파일에 다음 키가 있습니다.

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

결과: ens256의 호스트 MAC 주소는 b2:9c:02:c2:27:10이고 관리 네트워크 MAC 주소는 11:22:33:44:55:66입니다.

예시 **1:** 물리적 또는 가상 **NIC**에 대한 **1:1** 매핑

예제 1은 호스트 측 구성이 거의 또는 전혀 필요하지 않은 간단한 물리적 인터페이스 매핑에 대해 설명합니다.



Linux 운영 체제는 설치 또는 부팅 중에, 혹은 인터페이스를 핫 추가할 때 `ensXYZ`인터페이스를 자동으로 생성합니다. 부팅 후 인터페이스가 자동으로 활성화되도록 설정하는 것 외에는 별도의 구성이 필요하지 않습니다. 다만, 나중에 구성 과정에서 올바른 매핑을 제공할 수 있도록 어떤 ensXYZ 인터페이스가 어떤 StorageGRID 네트워크(Grid, Admin 또는

Client)에 해당하는지 미리 파악해 두어야 합니다.

그림에는 여러 StorageGRID 노드가 나와 있지만, 일반적으로 이 구성은 단일 노드 VM에 사용합니다.

스위치 1이 물리적 스위치인 경우, 인터페이스 10G1부터 10G3에 연결된 포트를 액세스 모드로 구성하고 적절한 VLAN에 배치해야 합니다.

예시 2: VLAN을 전송하는 LACP 본딩

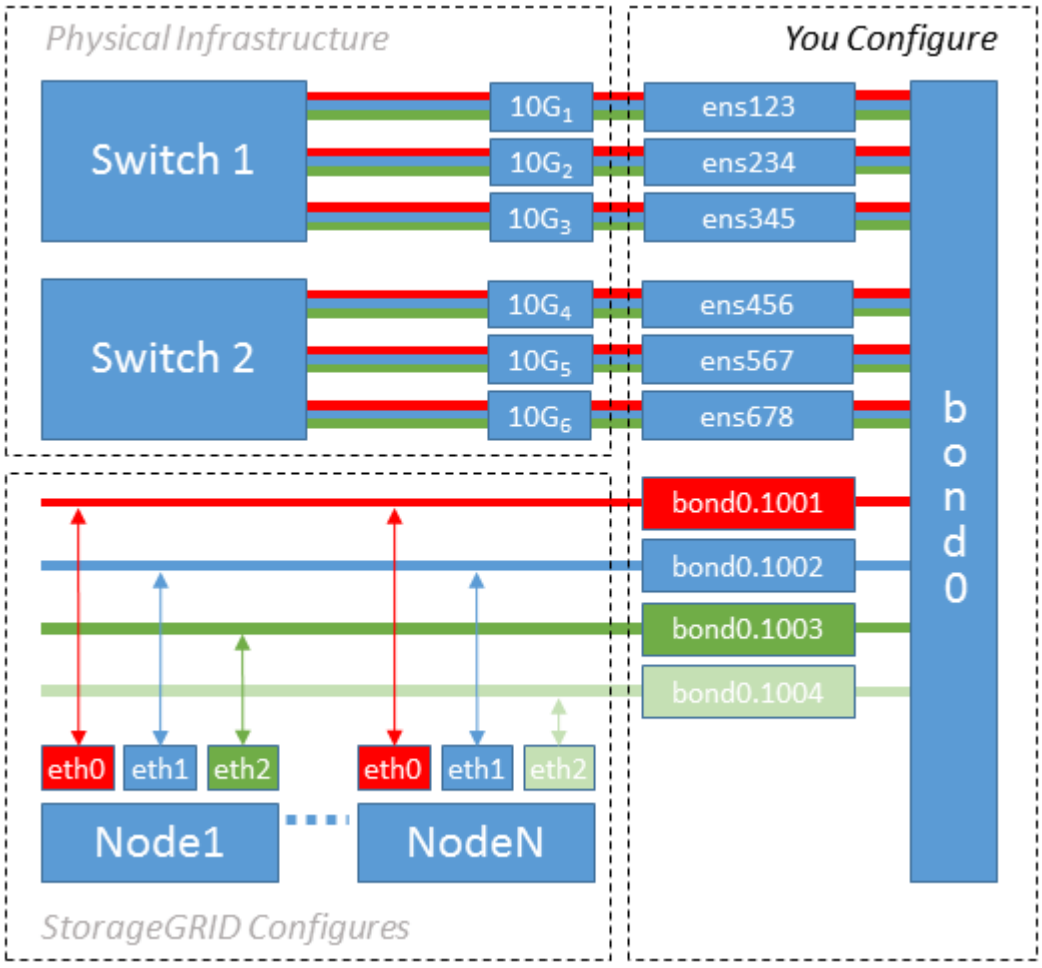
예제 2는 사용자가 네트워크 인터페이스 본딩과 사용 중인 Linux 배포판에서 VLAN 인터페이스를 생성하는 방법에 익숙하다고 가정합니다.

이 작업 정보

예제 2는 단일 호스트의 모든 노드에서 사용 가능한 네트워크 대역폭을 공유할 수 있도록 하는 일반적이고 유연한 VLAN 기반 체계를 설명합니다. 이 예제는 특히 베어메탈 호스트에 적용 가능합니다.

이 예시를 이해하기 위해 각 데이터 센터에 그리드, 관리 및 클라이언트 네트워크에 대해 각각 별도의 서브넷이 있다고 가정해 보겠습니다. 이 서브넷들은 서로 다른 VLAN(1001, 1002, 1003)에 있으며, LACP로 연결된 트렁크 포트(bond0)를 통해 호스트에 제공됩니다. 본드에 bond0.1001, bond0.1002, bond0.1003의 세 가지 VLAN 인터페이스를 구성해야 합니다.

동일한 호스트에서 노드 네트워크에 대해 별도의 VLAN 및 서브넷이 필요한 경우, 본드에 VLAN 인터페이스를 추가하고 이를 호스트에 매핑할 수 있습니다(그림에서 bond0.1004로 표시됨).



단계

1. StorageGRID 네트워크 연결에 사용될 모든 물리적 네트워크 인터페이스를 단일 LACP 본드로 집계합니다.

   모든 호스트에서 동일한 본드 이름을 사용하십시오. 예를 들어, `bond0`.

2. 이 본드를 관련 "물리적 장치"로 사용하는 VLAN 인터페이스를 표준 VLAN 인터페이스 명명 규칙 `physdev-name.VLAN ID`을 사용하여 생성합니다.

   1단계와 2단계를 수행하려면 네트워크 링크의 다른 끝단에 있는 에지 스위치에 적절한 구성이 필요합니다. 에지 스위치 포트도 LACP 포트 채널로 집계되고 트렁크로 구성되어야 하며 필요한 모든 VLAN이 통과할 수 있도록 허용되어야 합니다.

   호스트별 네트워킹 구성 체계에 대한 샘플 인터페이스 구성 파일이 제공됩니다.

관련 정보

- "Ubuntu 및 Debian의 /etc/network/interfaces 예시"
- "RHEL용 /etc/sysconfig/network-scripts 예시"

Linux 배포를 위한 StorageGRID 호스트 스토리지 구성

각 Linux 호스트에 블록 스토리지 볼륨을 할당해야 합니다.

시작하기 전에

다음 항목을 검토하셨으며, 이 항목들은 이 작업을 수행하는 데 필요한 정보를 제공합니다.

- "스토리지 및 성능 요구 사항"
- "Node 컨테이너 마이그레이션 요구 사항"

> "Linux"는 RHEL, Ubuntu 또는 Debian 배포판을 의미합니다. 지원되는 버전 목록은 "NetApp 상호 운용성 매트릭스 툴(IMT)"을 참조하십시오.

이 작업 정보

호스트에 블록 스토리지 볼륨(LUN)을 할당할 때는 "스토리지 요구 사항"의 표를 사용하여 다음 사항을 확인하십시오.

- 호스트당 필요한 볼륨 수(해당 호스트에 배포될 노드의 수 및 유형에 따라 결정됨)
- 각 볼륨의 저장 범주(즉, 시스템 데이터 또는 객체 데이터)
- 각 볼륨의 크기

호스트에 StorageGRID 노드를 배포할 때 이 정보와 Linux에서 각 물리적 볼륨에 할당한 영구 이름을 사용하게 됩니다.

> 이러한 볼륨을 파티션하거나 포맷하거나 마운트할 필요는 없습니다. 호스트에서 해당 볼륨이 보이도록만 하면 됩니다.

> 메타데이터 전용 스토리지 노드에는 객체 데이터 LUN이 하나만 필요합니다.

볼륨 이름 목록을 작성할 때 "raw" 특수 장치 파일(예: `/dev/sdb`) 사용을 피하십시오. 이러한 파일은 호스트 재부팅 시

변경될 수 있으며, 이는 시스템의 정상적인 작동에 영향을 미칠 수 있습니다. iSCSI LUN 및 Device Mapper 다중 경로를 사용하는 경우, 특히 SAN 토폴로지에 공유 스토리지에 대한 중복 네트워크 경로가 포함된 경우 `/dev/mapper` 디렉토리에서 다중 경로 별칭을 사용하는 것을 고려하십시오. 또는 영구 장치 이름에 `/dev/disk/by-path/` 아래의 시스템에서 생성된 소프트링크를 사용할 수도 있습니다.

예를 들면 다음과 같습니다.

```
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
```

설치 환경에 따라 결과가 다를 수 있습니다.

StorageGRID 설치 및 향후 유지 관리 절차를 간소화하기 위해 각 블록 스토리지 볼륨에 친숙한 이름을 지정하십시오. 공유 스토리지 볼륨에 대한 이중화 액세스를 위해 장치 매퍼 다중 경로 드라이버를 사용하는 경우 `alias` 파일의 `/etc/multipath.conf` 필드를 사용할 수 있습니다.

예를 들면 다음과 같습니다.

```
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
      }
      …
```

이와 같이 별칭 필드를 사용하면 별칭이 호스트의 `/dev/mapper` 디렉터리에 블록 장치로 나타나므로 구성 또는 유지 관리 작업에서 블록 스토리지 볼륨을 지정해야 할 때마다 친숙하고 유효성 검사가 쉬운 이름을 지정할 수 있습니다.

StorageGRID 노드 마이그레이션을 지원하기 위해 공유 스토리지를 설정하고 Device Mapper 다중 경로를 사용하는 경우, 모든 동일 위치 호스트에 공통 `/etc/multipath.conf`을 생성하고 설치할 수 있습니다. 단, 각 호스트에서 서로 다른 컨테이너 엔진 스토리지 볼륨을 사용해야 합니다. 각 컨테이너 엔진 스토리지 볼륨 LUN에 별칭을 사용하고 별칭에 대상 호스트 이름을 포함시키면 기억하기 쉬우므로 권장됩니다.

> ⓘ 소프트웨어 전용 배포를 위한 컨테이너 엔진으로서 Docker 지원이 더 이상 권장되지 않습니다. Docker는 향후 릴리스에서 다른 컨테이너 엔진으로 대체될 예정입니다.

관련 정보

- "컨테이너 엔진 스토리지 볼륨 구성"
- "스토리지 및 성능 요구 사항"

- "Node 컨테이너 마이그레이션 요구 사항"

Linux에서 StorageGRID용 컨테이너 엔진 스토리지 볼륨 구성

Docker 또는 Podman 컨테이너 엔진을 설치하기 전에 스토리지 볼륨을 포맷하고 마운트해야 할 수도 있습니다.

> 소프트웨어 전용 배포를 위한 컨테이너 엔진으로서 Docker 지원이 더 이상 권장되지 않습니다. Docker는 향후 릴리스에서 다른 컨테이너 엔진으로 대체될 예정입니다.

> "Linux"는 RHEL, Ubuntu 또는 Debian 배포판을 의미합니다. 지원되는 버전 목록은 "NetApp 상호 운용성 매트릭스 툴(IMT)"을 참조하십시오.

이 작업 정보

Docker 또는 Podman 스토리지 볼륨에 루트 볼륨을 사용할 계획이고 다음을 포함하는 호스트 파티션에 충분한 여유 공간이 있는 경우 이 단계를 건너뛸 수 있습니다.

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

단계

1. 컨테이너 엔진 스토리지 볼륨에 파일 시스템을 생성합니다.

RHEL

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

Ubuntu 또는 **Debian**

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

2. 컨테이너 엔진 저장 볼륨을 마운트합니다.

RHEL

- Docker의 경우:

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

- Podman의 경우:

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

Ubuntu 또는 **Debian**

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

- Podman의 경우:

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

3. 컨테이너 스토리지 볼륨 장치에 대한 항목을 /etc/fstab 파일에 추가합니다.
   - RHEL: container-storage-volume-device
   - Ubuntu 또는 Debian: docker-storage-volume-device

   이 단계를 통해 호스트 재부팅 후 스토리지 볼륨이 자동으로 다시 마운트됩니다.

Docker 설치

StorageGRID 시스템은 컨테이너 모음으로 Linux에서 실행될 수 있습니다.

- Ubuntu 또는 Debian용 StorageGRID를 설치하기 전에 Docker를 설치해야 합니다.
- Docker 컨테이너 엔진을 사용하기로 선택했다면 다음 단계를 따라 Docker를 설치하십시오. 그렇지 않은 경우, Podman 설치.

> 소프트웨어 전용 배포를 위한 컨테이너 엔진으로서 Docker 지원이 더 이상 권장되지 않습니다. Docker는 향후 릴리스에서 다른 컨테이너 엔진으로 대체될 예정입니다.

단계

1. 사용하는 Linux 배포판의 지침에 따라 Docker를 설치합니다.

사용하는 Linux 배포판에 Docker가 포함되어 있지 않다면 Docker 웹사이트에서 다운로드할 수 있습니다.

2. Docker가 활성화되고 시작되었는지 확인하려면 다음 두 명령을 실행하십시오.

```
sudo systemctl enable docker
```

```
sudo systemctl start docker
```

3. 다음 명령어를 입력하여 예상되는 버전의 Docker가 설치되었는지 확인하십시오.

```
sudo docker version
```

클라이언트 및 서버 버전은 1.11.0 이상이어야 합니다.

Podman 설치

StorageGRID 시스템은 컨테이너 모음으로 실행됩니다. Podman 컨테이너 엔진을 사용하기로 선택한 경우 다음 단계를 따라 Podman을 설치하십시오. 그렇지 않은 경우 Docker 설치.

단계

1. 사용하는 Linux 배포판의 지침에 따라 Podman과 Podman-Docker를 설치하십시오.

Podman을 설치할 때 Podman-Docker 패키지도 함께 설치해야 합니다.

2. 다음 명령어를 입력하여 Podman 및 Podman-Docker의 예상 버전이 설치되었는지 확인하십시오.

```
sudo docker version
```

Podman-Docker 패키지를 사용하면 Docker 명령어를 사용할 수 있습니다.

클라이언트 및 서버 버전은 3.2.3 이상이어야 합니다.

```
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
```

관련 정보

"호스트 스토리지 구성"

Linux에 StorageGRID 호스트 서비스를 설치합니다

운영 체제 유형에 맞는 StorageGRID 패키지를 사용하여 StorageGRID 호스트 서비스를 설치합니다.

"Linux"는 RHEL, Ubuntu 또는 Debian 배포판을 의미합니다. 지원되는 버전 목록은 "NetApp 상호 운용성 매트릭스 툴(IMT)"을 참조하십시오.

RHEL

StorageGRID 호스트 서비스를 설치하려면 StorageGRID RPM 패키지를 사용합니다.

이 작업 정보

이 지침에서는 RPM 패키지를 사용하여 호스트 서비스를 설치하는 방법을 설명합니다. 또는 설치 아카이브에 포함된 DNF 저장소 메타데이터를 사용하여 RPM 패키지를 원격으로 설치할 수도 있습니다. 사용 중인 Linux 운영 체제에 맞는 DNF 저장소 지침을 참조하십시오.

단계

1. StorageGRID RPM 패키지를 각 호스트에 복사하거나 공유 스토리지에서 사용할 수 있도록 합니다.

   예를 들어, 다음 단계에서 예제 명령을 사용할 수 있도록 해당 파일들을 `/tmp` 디렉토리에 넣어 두십시오.

2. 루트 사용자 또는 sudo 권한이 있는 계정으로 각 호스트에 로그인한 다음, 지정된 순서대로 다음 명령어를 실행하십시오.

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

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

   이미지 패키지를 먼저 설치하고, 그 다음에 서비스 패키지를 설치해야 합니다.

   패키지를 `/tmp` 이외의 다른 디렉토리에 배치한 경우, 사용한 경로를 반영하도록 명령을 수정하십시오.

Ubuntu 또는 **Debian**

StorageGRID DEB 패키지를 사용하여 Ubuntu 또는 Debian용 StorageGRID 호스트 서비스를 설치합니다.

이 작업 정보

이 지침에서는 DEB 패키지에서 호스트 서비스를 설치하는 방법을 설명합니다. 또는 설치 아카이브에 포함된 APT 저장소 메타데이터를 사용하여 DEB 패키지를 원격으로 설치할 수도 있습니다. 사용 중인 Linux 운영 체제에 맞는 APT 저장소 지침을 참조하십시오.

단계

1. StorageGRID DEB 패키지를 각 호스트에 복사하거나 공유 스토리지에서 사용할 수 있도록 합니다.

   예를 들어, 다음 단계에서 예제 명령을 사용할 수 있도록 해당 파일들을 `/tmp` 디렉토리에 넣어 두십시오.

2. 루트 사용자 또는 sudo 권한이 있는 계정으로 각 호스트에 로그인한 후 다음 명령을 실행하십시오.

   먼저 `images` 패키지를 설치하고, 그 다음에 `service` 패키지를 설치해야 합니다. 패키지를 `/tmp` 이외의 다른 디렉토리에 배치한 경우, 사용한 경로를 반영하도록 명령을 수정하십시오.

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

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

StorageGRID 패키지를 설치하려면 Python 3이 이미 설치되어 있어야 합니다. `sudo dpkg --install /tmp/storagegrid-webscale-images-version-SHA.deb` 명령은 Python 3을 설치하기 전까지 실패합니다.