



호스트 준비(Red Hat) StorageGRID 11.8

NetApp
March 19, 2024

목차

호스트 준비(Red Hat)	1
설치 중에 호스트 전체의 설정이 변경되는 방식	1
Linux를 설치합니다	2
호스트 네트워크 구성(Red Hat Enterprise Linux)	3
호스트 스토리지를 구성합니다	7
컨테이너 엔진 저장소 볼륨을 구성합니다	10
StorageGRID 호스트 서비스를 설치합니다	11

호스트 준비(Red Hat)

설치 중에 호스트 전체의 설정이 변경되는 방식

베어 메탈 시스템에서 StorageGRID는 호스트 전체를 일부 변경합니다 `sysctl` 설정.

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
```

Linux를 설치합니다

모든 Red Hat Enterprise Linux 그리드 호스트에 StorageGRID를 설치해야 합니다. 지원되는 버전 목록은 NetApp 상호 운용성 매트릭스 툴을 참조하십시오.



운영 체제가 Linux 커널 4.15 이상으로 업그레이드되었는지 확인합니다.

단계

1. 배포자의 지침 또는 표준 절차에 따라 모든 물리적 또는 가상 그리드 호스트에 Linux를 설치합니다.



표준 Linux 설치 프로그램을 사용하는 경우 NetApp에서는 "컴퓨팅 노드" 소프트웨어 구성을 선택하거나(사용 가능한 경우) "최소 설치" 기본 환경을 선택하는 것이 좋습니다. 그래픽 데스크톱 환경을 설치하지 마십시오.

2. Extras 채널을 포함하여 모든 호스트가 패키지 리포지토리에 액세스할 수 있는지 확인합니다.

이 설치 절차의 뒷부분에서 이러한 추가 패키지가 필요할 수 있습니다.

3. 스왑이 활성화된 경우:

- a. 다음 명령을 실행합니다. `$ sudo swapoff --all`
- b. 에서 모든 스왑 항목을 제거합니다 `/etc/fstab` 를 눌러 설정을 유지합니다.



스왑을 완전히 사용하지 않도록 설정하면 성능이 크게 저하될 수 있습니다.

호스트 네트워크 구성(Red Hat Enterprise Linux)

호스트에서 Linux 설치를 완료한 후 나중에 배포할 StorageGRID 노드에 매핑하는 데 적합한 네트워크 인터페이스 세트를 준비하기 위해 몇 가지 추가 구성을 수행해야 할 수 있습니다.

시작하기 전에

- 을(를) 검토했습니다 ["StorageGRID 네트워킹 지침"](#).
- 에 대한 정보를 검토했습니다 ["노드 컨테이너 마이그레이션 요구사항"](#).
- 가상 호스트를 사용하는 경우 를 읽었습니다 [MAC 주소 복제에 대한 고려 사항 및 권장 사항](#) 호스트 네트워크를 구성하기 전에



VM을 호스트로 사용하는 경우 가상 네트워크 어댑터로 VMXNET 3을 선택해야 합니다. VMware E1000 네트워크 어댑터로 인해 특정 Linux 배포판에 배포된 StorageGRID 컨테이너의 연결 문제가 발생했습니다.

이 작업에 대해

그리드 노드는 그리드 네트워크와 선택적으로 관리자 및 클라이언트 네트워크에 액세스할 수 있어야 합니다. 호스트의 물리적 인터페이스를 각 그리드 노드의 가상 인터페이스에 연결하는 매핑을 생성하여 이 액세스를 제공합니다. 호스트 인터페이스를 생성할 때 이름을 friendly 로 사용하여 모든 호스트에 쉽게 구축하고 마이그레이션을 설정할 수 있습니다.

호스트와 하나 이상의 노드 간에 동일한 인터페이스를 공유할 수 있습니다. 예를 들어, 호스트 액세스 및 노드 관리 네트워크 액세스에 동일한 인터페이스를 사용하여 호스트 및 노드 유지 관리를 용이하게 할 수 있습니다. 호스트와 개별 노드 간에 동일한 인터페이스를 공유할 수 있지만 모두 IP 주소가 서로 달라야 합니다. IP 주소는 노드 간 또는 호스트와 노드 간에 공유할 수 없습니다.

동일한 호스트 네트워크 인터페이스를 사용하여 호스트의 모든 StorageGRID 노드에 그리드 네트워크 인터페이스를 제공하거나, 각 노드에 대해 다른 호스트 네트워크 인터페이스를 사용하거나, 둘 사이에 작업을 수행할 수 있습니다. 그러나 일반적으로 단일 노드에 대한 Grid 및 Admin Network 인터페이스와 동일한 호스트 네트워크 인터페이스를 제공하거나 한 노드에 대한 Grid Network 인터페이스와 다른 노드에 대한 Client Network 인터페이스를 제공하지 않습니다.

이 작업은 여러 가지 방법으로 완료할 수 있습니다. 예를 들어, 호스트가 가상 머신이고 각 호스트에 대해 하나 또는 두 개의 StorageGRID 노드를 구축하는 경우 하이퍼바이저에서 올바른 수의 네트워크 인터페이스를 생성하고 일대일 매핑을 사용할 수 있습니다. 운영 용도로 베어 메탈 호스트에 여러 노드를 구축하는 경우 Linux 네트워킹 스택이 VLAN 및 LACP 지원을 활용하여 내결함성 및 대역폭 공유를 제공할 수 있습니다. 다음 섹션에서는 이러한 두 가지 예에 대해 자세히 설명합니다. 이러한 예제 중 하나를 사용할 필요가 없습니다. 필요에 맞는 방법을 사용할 수 있습니다.



Bond 또는 Bridge 장치를 컨테이너 네트워크 인터페이스로 직접 사용하지 마십시오. 이렇게 하면 컨테이너 네임스페이스의 연결 및 브리지 장치와 함께 MACVLAN을 사용하는 커널 문제로 인해 노드 시작이 방지될 수 있습니다. 대신 VLAN 또는 가상 이더넷(veth) 쌍과 같은 비연결 장치를 사용하십시오. 이 디바이스를 노드 구성 파일의 네트워크 인터페이스로 지정합니다.

관련 정보

["노드 구성 파일을 만드는 중입니다"](#)

MAC 주소 복제에 대한 고려 사항 및 권장 사항

MAC 주소 클로닝은 컨테이너가 호스트의 MAC 주소를 사용하고 호스트는 사용자가 지정한 주소나 임의로 생성된 주소의 MAC 주소를 사용하게 합니다. 무차별 모드 네트워크 구성을 사용하지 않으려면 MAC 주소 복제를 사용해야 합니다.

MAC 클론 생성 활성화

특정 환경에서는 관리 네트워크, 그리드 네트워크 및 클라이언트 네트워크에 전용 가상 NIC를 사용할 수 있으므로 MAC 주소 클로닝을 통해 보안을 강화할 수 있습니다. 컨테이너가 호스트에 있는 전용 NIC의 MAC 주소를 사용하도록 하면 무차별 모드 네트워크 구성을 사용하지 않도록 할 수 있습니다.



MAC 주소 복제는 가상 서버 설치에 사용하기 위한 것이며 모든 물리적 어플라이언스 구성에서 제대로 작동하지 않을 수 있습니다.



MAC 클론 대상 인터페이스가 사용 중이어서 노드가 시작되지 않는 경우 노드를 시작하기 전에 링크를 "다운"으로 설정해야 할 수 있습니다. 또한 링크가 작동 중일 때 가상 환경에서 네트워크 인터페이스에서 MAC 클로닝을 방지할 수 있습니다. 노드가 MAC 주소를 설정하지 못하고 사용 중인 인터페이스로 인해 시작되는 경우 노드를 시작하기 전에 링크를 "다운"으로 설정하면 문제가 해결될 수 있습니다.

MAC 주소 복제는 기본적으로 해제되어 있으며 노드 구성 키로 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

키를 "true"로 설정하면 컨테이너가 호스트 NIC의 MAC 주소를 사용하게 됩니다. 또한 호스트는 지정된 컨테이너 네트워크의 MAC 주소를 사용합니다. 기본적으로 컨테이너 주소는 임의로 생성된 주소이지만 을 사용하여 주소를 설정한 경우 `_NETWORK_MAC` 노드 구성 키로, 해당 주소가 대신 사용됩니다. 호스트와 컨테이너의 MAC 주소는 항상 다릅니다.



하이퍼바이저에서 무차별 모드를 설정하지 않고 가상 호스트에서 MAC 클로닝을 활성화하면 호스트의 인터페이스를 사용하는 Linux 호스트 네트워킹이 작동하지 않을 수 있습니다.

Mac 클론 복제 활용 사례

MAC 클로닝에는 다음 두 가지 사용 사례를 고려해야 합니다.

- Mac 클론 생성이 활성화되지 않음: 이 활성화된 경우 `_CLONE_MAC` 노드 구성 파일의 키가 설정되지 않았거나 "false"로 설정된 경우 호스트는 호스트 NIC MAC을 사용하고 컨테이너에 MAC이 지정되지 않은 경우 StorageGRID 생성 MAC을 갖게 됩니다. `_NETWORK_MAC` 키. 에 주소가 설정되어 있는 경우 `_NETWORK_MAC` 키, 컨테이너에 에 지정된 주소가 있습니다 `_NETWORK_MAC` 키. 이러한 키 구성을 위해서는 무차별 모드를 사용해야 합니다.
- Mac 클론 생성 활성화 `_CLONE_MAC` 노드 구성 파일의 키는 "true"로 설정되어 있고, 컨테이너는 호스트 NIC MAC을 사용하며, 호스트는 에 MAC을 지정하지 않는 한 StorageGRID 생성 MAC을 사용합니다. `_NETWORK_MAC` 키. 에 주소가 설정되어 있는 경우 `_NETWORK_MAC` 키, 호스트는 생성된 주소 대신 지정된 주소를 사용합니다. 이 키 구성에서 무차별 모드를 사용해서는 안 됩니다.



MAC 주소 클로닝을 사용하지 않고 하이퍼바이저에 의해 할당된 것이 아닌 MAC 주소에 대한 데이터를 모든 인터페이스에서 수신 및 전송하도록 허용하려면 가상 스위치 및 포트 그룹 수준의 보안 속성이 Promiscuous Mode, MAC Address 변경 및 Forged 전송에 대해 *Accept* 로 설정되어 있는지 확인합니다. 가상 스위치에 설정된 값은 포트 그룹 수준의 값으로 재정의할 수 있으므로 두 위치에서 설정이 동일한지 확인합니다.

MAC 클로닝을 활성화하려면 를 참조하십시오 ["노드 구성 파일 생성 지침"](#).

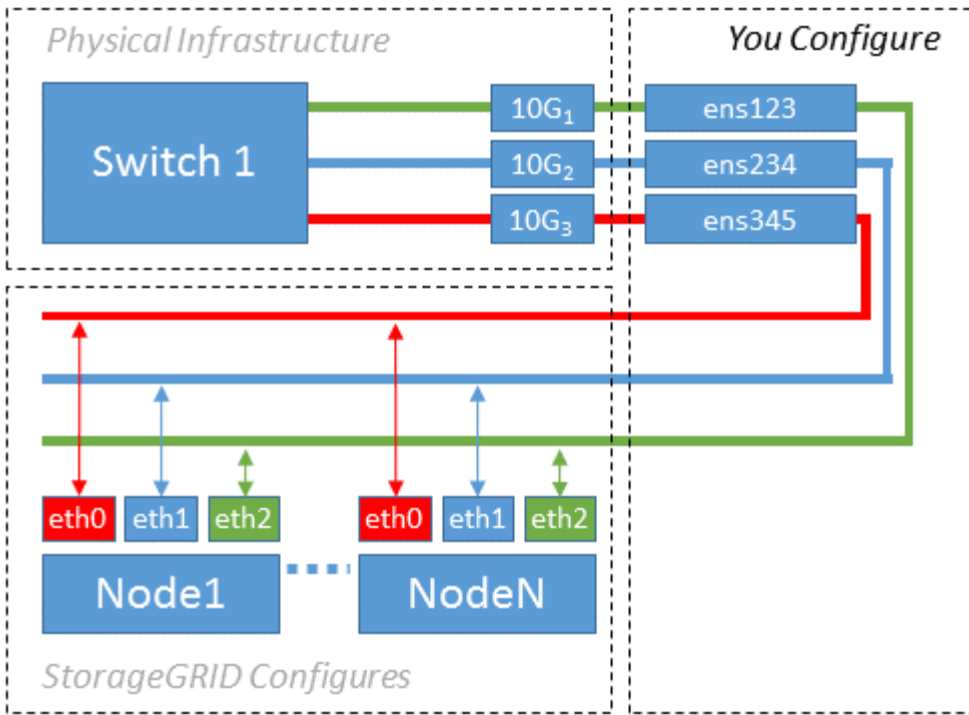
Mac 클론 복제의 예

인터페이스 `ens256`의 경우 MAC 주소가 `11:22:33:44:55:66`이고 노드 구성 파일의 경우 다음 키가 있는 호스트에서 활성화된 MAC 클론 복제의 예:

- `ADMIN_NETWORK_TARGET = ens256`
- `ADMIN_NETWORK_MAC = b2:9c:02:c2:27:10`
- `ADMIN_NETWORK_TARGET_TYPE_INTERFACE_CLONE_MAC = true`
- 결과 *: `en256`의 호스트 MAC은 `B2:9c:02:C2:27:10`이고 관리 네트워크 MAC은 `11:22:33:44:55:66`입니다

예 1: 물리적 NIC 또는 가상 NIC에 1:1 대 1 매핑

예제 1에서는 호스트측 구성이 거의 또는 전혀 필요하지 않은 간단한 물리적 인터페이스 매핑에 대해 설명합니다.



Linux 운영 체제에서 를 생성합니다 ensXYZ 설치 또는 부팅 중 또는 인터페이스가 핫 애드 될 때 자동으로 인터페이스가 설정됩니다. 부팅 후 인터페이스가 자동으로 실행되도록 설정하는 것 외에는 구성이 필요하지 않습니다. 어떤 것을 결정해야 합니다 ensXYZ 나중에 구성 프로세스에서 올바른 매핑을 제공할 수 있도록 StorageGRID 네트워크(그리드, 관리자 또는 클라이언트)에 해당합니다.

이 그림에서는 여러 StorageGRID 노드를 보여 줍니다. 그러나 일반적으로 단일 노드 VM에 이 구성을 사용합니다.

스위치 1이 물리적 스위치인 경우 액세스 모드에 대해 인터페이스 10G1 - 10G3에 연결된 포트를 구성하고 해당 VLAN에 배치해야 합니다.

예 2: VLAN을 전달하는 LACP 결합

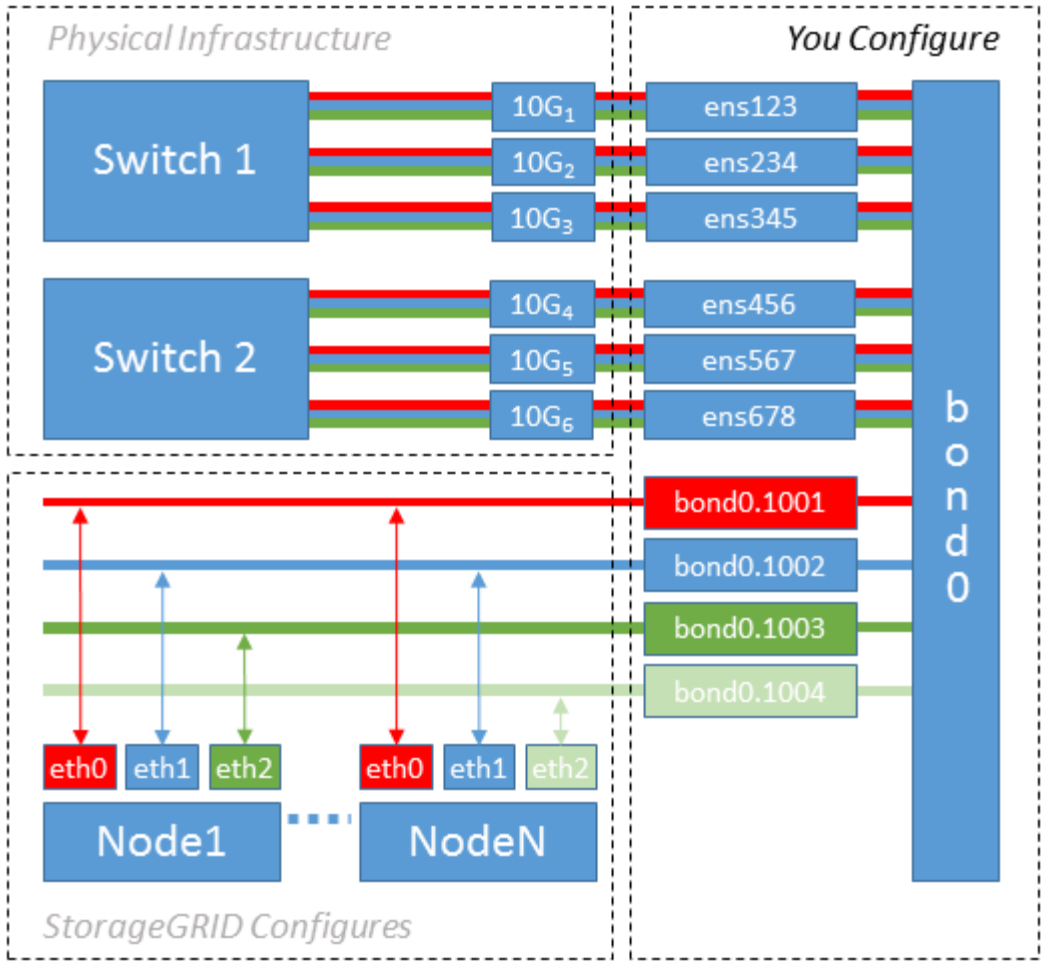
이 작업에 대해

예제 2에서는 네트워크 인터페이스를 결합하거나 사용 중인 Linux 배포판에서 VLAN 인터페이스를 만드는 방법에 대해 잘 알고 있다고 가정합니다.

예제 2에서는 단일 호스트의 모든 노드에서 사용 가능한 모든 네트워크 대역폭을 쉽게 공유할 수 있도록 지원하는 일반, 유연한 VLAN 기반 체계를 설명합니다. 이 예는 특히 베어 메탈 호스트에 적용할 수 있습니다.

이 예제를 이해하려면 각 데이터 센터에 그리드, 관리자 및 클라이언트 네트워크에 대한 세 개의 개별 서브넷이 있다고 가정합니다. 서브넷은 별도의 VLAN(1001, 1002 및 1003)에 있으며 LACP 결합 트렁크 포트(bond0)의 호스트에 제공됩니다. Bond.0.1001, bond.0.1002 및 bond.0.1003의 세 가지 VLAN 인터페이스를 구성합니다.

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



단계

1. StorageGRID 네트워크 연결에 사용할 모든 물리적 네트워크 인터페이스를 단일 LACP 결합으로 통합합니다.

모든 호스트에서 본드에도 동일한 이름을 사용합니다. 예를 들면, 다음과 같습니다. bond0.

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

1단계와 2단계는 네트워크 링크의 다른 끝을 종료하는 에지 스위치에 적절한 구성이 필요합니다. 에지 스위치 포트도 LACP 포트 채널로 집계되고 트렁크로 구성되어 필요한 모든 VLAN을 통과할 수 있도록 허용해야 합니다.

호스트별 네트워킹 구성 체계에 대한 샘플 인터페이스 구성 파일이 제공됩니다.

관련 정보

"예 [/etc/sysconfig/network-scripts](#)"

호스트 스토리지를 구성합니다

각 호스트에 블록 스토리지 볼륨을 할당해야 합니다.

시작하기 전에

이 과제를 수행하는 데 필요한 정보를 제공하는 다음 주제를 검토했습니다.

"요구사항을 충족해야 합니다"

"노드 컨테이너 마이그레이션 요구사항"

이 작업에 대해

블록 스토리지 볼륨(LUN)을 호스트에 할당할 때 "스토리지 요구 사항"의 표를 사용하여 다음을 확인합니다.

- 각 호스트에 필요한 볼륨 수(해당 호스트에 구축할 노드 수 및 유형 기준)
- 각 볼륨의 스토리지 범주(즉, 시스템 데이터 또는 오브젝트 데이터)
- 각 볼륨의 크기입니다

호스트에 StorageGRID 노드를 배포할 때 이 정보와 Linux가 각 물리적 볼륨에 할당한 영구 이름을 사용합니다.



이러한 볼륨을 파티션, 포맷 또는 마운트할 필요가 없습니다. 호스트가 볼 수 있도록 해야 합니다.



메타데이터 전용 스토리지 노드에는 하나의 오브젝트 데이터 LUN만 필요합니다.

"RAW" 특수 장치 파일을 사용하지 마십시오 (/dev/sdb`예를 들어) 볼륨 이름 목록을 작성할 때 사용합니다. 이러한 파일은 호스트의 재부팅 시 변경될 수 있으며, 이는 시스템의 올바른 작동에 영향을 줍니다. iSCSI LUN 및 Device Mapper Multipathing을 사용하는 경우 에서 다중 경로 별칭을 사용하는 것이 좋습니다. /dev/mapper 디렉토리 - 특히 SAN 토폴로지에 공유 스토리지에 대한 이중 네트워크 경로가 포함된 경우 또는 에서 시스템에서 만든 소프트링크를 사용할 수도 있습니다 /dev/disk/by-path/ 를 클릭합니다.

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
```

각 설치 환경에 따라 결과가 달라집니다.

각 블록 스토리지 볼륨에 알기 쉬운 이름을 할당하여 초기 StorageGRID 설치 및 향후 유지 관리 절차를 간소화하십시오. 공유 스토리지 볼륨에 대한 중복 액세스를 위해 장치 매퍼 다중 경로 드라이버를 사용하는 경우 를

사용할 수 있습니다 alias 필드에 입력합니다 /etc/multipath.conf 파일.

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
  }
  ...
}
```

그러면 별칭이 에서 블록 디바이스로 나타납니다 /dev/mapper 호스트의 디렉토리로, 구성 또는 유지보수 작업에서 블록 스토리지 볼륨을 지정해야 할 때마다 친숙하고 쉽게 검증된 이름을 지정할 수 있습니다.



StorageGRID 노드 마이그레이션을 지원하고 장치 매퍼 다중 경로를 사용하도록 공유 스토리지를 설정하는 경우 공통 을 만들고 설치할 수 있습니다 /etc/multipath.conf 모든 동일 위치 호스트. 각 호스트에서 다른 컨테이너 엔진 스토리지 볼륨을 사용해야 합니다. 별칭을 사용하고 각 컨테이너 엔진 스토리지 볼륨 LUN의 별칭에 타겟 호스트 이름을 포함시키면 기억하기 쉽고 권장됩니다.

관련 정보

["컨테이너 엔진 저장소 볼륨을 구성합니다"](#)

컨테이너 엔진 저장소 볼륨을 구성합니다

컨테이너 엔진(Docker 또는 Podman)을 설치하기 전에 스토리지 볼륨을 포맷하고 마운트해야 할 수 있습니다.

이 작업에 대해

Docker 또는 Podman 스토리지 볼륨에 로컬 스토리지를 사용할 계획이며 이 포함된 호스트 파티션에 사용 가능한 공간이 충분한 경우 이 단계를 건너뛸 수 있습니다 /var/lib/docker Docker 및 용 /var/lib/containers 팟맨용.



Podman은 Red Hat Enterprise Linux(RHEL)에서만 지원됩니다.

단계

1. 컨테이너 엔진 스토리지 볼륨에 파일 시스템을 생성합니다.

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

2. 컨테이너 엔진 저장소 볼륨을 마운트합니다.

◦ 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
```

3. /etc/fstab에 container-storage-volume-device에 대한 항목을 추가합니다.

이 단계를 수행하면 호스트가 재부팅된 후 스토리지 볼륨이 자동으로 다시 마운트됩니다.

Docker를 설치합니다

StorageGRID 시스템은 Red Hat Enterprise Linux에서 컨테이너 모음으로 실행됩니다. Docker 컨테이너 엔진을 사용하도록 선택한 경우 다음 단계에 따라 Docker를 설치합니다. 그렇지 않으면 [Podman을 설치합니다](#).

단계

1. Linux 배포에 대한 지침에 따라 Docker를 설치합니다.



Docker가 Linux 배포판에 포함되어 있지 않은 경우 Docker 웹 사이트에서 다운로드할 수 있습니다.

2. 다음 두 명령을 실행하여 Docker를 활성화하고 시작했는지 확인합니다.

```
sudo systemctl enable docker
```

```
sudo systemctl start docker
```

3. 다음을 입력하여 예상 버전의 Docker를 설치했는지 확인합니다.

```
sudo docker version
```

클라이언트 및 서버 버전은 1.11.0 이상이어야 합니다.

Podman을 설치합니다

StorageGRID 시스템은 Red Hat Enterprise Linux에서 컨테이너 모음으로 실행됩니다. Podman 컨테이너 엔진을 사용하도록 선택한 경우 다음 단계에 따라 Podman을 설치합니다. 그렇지 않으면 [Docker를 설치합니다](#).



Podman은 Red Hat Enterprise Linux(RHEL)에서만 지원됩니다.

단계

1. Linux 배포에 대한 지침에 따라 Podman 및 Podman-Docker를 설치합니다.



Podman을 설치할 때 Podman-docker 패키지도 설치해야 합니다.

2. 다음을 입력하여 예상 버전의 Podman 및 Podman-Docker를 설치했는지 확인합니다.

```
sudo docker version
```



Podman-Docker 패키지를 사용하면 Docker 명령을 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
```

StorageGRID 호스트 서비스를 설치합니다

StorageGRID RPM 패키지를 사용하여 StorageGRID 호스트 서비스를 설치합니다.

이 작업에 대해

다음 지침은 RPM 패키지에서 호스트 서비스를 설치하는 방법을 설명합니다. 또는 설치 아카이브에 포함된 Yum 리포지토리 메타데이터를 사용하여 RPM 패키지를 원격으로 설치할 수 있습니다. Linux 운영 체제에 대한 Yum 리포지토리 지침을 참조하십시오.

단계

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

예를 들어, 에 배치합니다 /tmp 다음 단계에서 예제 명령을 사용할 수 있도록 디렉터리를 지정합니다.

2. 각 호스트에 루트로 로그인하거나 sudo 권한이 있는 계정을 사용하여 다음 명령을 지정된 순서대로 실행합니다.

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

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



먼저 이미지 패키지를 설치하고 서비스 패키지를 다시 설치해야 합니다.



를 제외한 다른 디렉터리에 패키지를 배치한 경우 `/tmp`에서 명령을 수정하여 사용한 경로를 반영합니다.

저작권 정보

Copyright © 2024 NetApp, Inc. All Rights Reserved. 미국에서 인쇄된 본 문서의 어떠한 부분도 저작권 소유자의 사전 서면 승인 없이는 어떠한 형식이나 수단(복사, 녹음, 녹화 또는 전자 검색 시스템에 저장하는 것을 비롯한 그래픽, 전자적 또는 기계적 방법)으로도 복제될 수 없습니다.

NetApp이 저작권을 가진 자료에 있는 소프트웨어에는 아래의 라이선스와 고지사항이 적용됩니다.

본 소프트웨어는 NetApp에 의해 '있는 그대로' 제공되며 상품성 및 특정 목적에의 적합성에 대한 명시적 또는 묵시적 보증을 포함하여(이에 제한되지 않음) 어떠한 보증도 하지 않습니다. NetApp은 대체품 또는 대체 서비스의 조달, 사용 불능, 데이터 손실, 이익 손실, 영업 중단을 포함하여(이에 국한되지 않음), 이 소프트웨어의 사용으로 인해 발생하는 모든 직접 및 간접 손해, 우발적 손해, 특별 손해, 징벌적 손해, 결과적 손해의 발생에 대하여 그 발생 이유, 책임론, 계약 여부, 엄격한 책임, 불법 행위(과실 또는 그렇지 않은 경우)와 관계없이 어떠한 책임도 지지 않으며, 이와 같은 손실의 발생 가능성이 통지되었다 하더라도 마찬가지입니다.

NetApp은 본 문서에 설명된 제품을 언제든지 예고 없이 변경할 권리를 보유합니다. NetApp은 NetApp의 명시적인 서면 동의를 받은 경우를 제외하고 본 문서에 설명된 제품을 사용하여 발생하는 어떠한 문제에도 책임을 지지 않습니다. 본 제품의 사용 또는 구매의 경우 NetApp에서는 어떠한 특허권, 상표권 또는 기타 지적 재산권이 적용되는 라이선스도 제공하지 않습니다.

본 설명서에 설명된 제품은 하나 이상의 미국 특허, 해외 특허 또는 출원 중인 특허로 보호됩니다.

제한적 권리 표시: 정부에 의한 사용, 복제 또는 공개에는 DFARS 252.227-7013(2014년 2월) 및 FAR 52.227-19(2007년 12월)의 기술 데이터-비상업적 품목에 대한 권리(Rights in Technical Data -Noncommercial Items) 조항의 하위 조항 (b)(3)에 설명된 제한사항이 적용됩니다.

여기에 포함된 데이터는 상업용 제품 및/또는 상업용 서비스(FAR 2.101에 정의)에 해당하며 NetApp, Inc.의 독점 자산입니다. 본 계약에 따라 제공되는 모든 NetApp 기술 데이터 및 컴퓨터 소프트웨어는 본질적으로 상업용이며 개인 비용만으로 개발되었습니다. 미국 정부는 데이터가 제공된 미국 계약과 관련하여 해당 계약을 지원하는 데에만 데이터에 대한 전 세계적으로 비독점적이고 양도할 수 없으며 재사용이 불가능하며 취소 불가능한 라이선스를 제한적으로 가집니다. 여기에 제공된 경우를 제외하고 NetApp, Inc.의 사전 서면 승인 없이는 이 데이터를 사용, 공개, 재생산, 수정, 수행 또는 표시할 수 없습니다. 미국 국방부에 대한 정부 라이선스는 DFARS 조항 252.227-7015(b)(2014년 2월)에 명시된 권한으로 제한됩니다.

상표 정보

NETAPP, NETAPP 로고 및 <http://www.netapp.com/TM>에 나열된 마크는 NetApp, Inc.의 상표입니다. 기타 회사 및 제품 이름은 해당 소유자의 상표일 수 있습니다.