소프트웨어 기반 노드에 대한 설치 계획 및 준비

StorageGRID software

NetApp
July 23, 2026

This PDF was generated from https://docs.netapp.com/ko-kr/storagegrid-120/swnodes/required-materials.html on July 23, 2026. Always check docs.netapp.com for the latest.

목차

소프트웨어 기반 노드에 대한 설치 계획 및 준비 1
StorageGRID에 필요한 정보 및 자료 1
필수 정보 1
필수 재료 1
StorageGRID 설치 파일 다운로드 및 압축 해제 2
StorageGRID 설치 파일을 확인하십시오 7
StorageGRID의 소프트웨어 요구 사항 8
StorageGRID의 CPU 및 RAM 요구 사항 12
StorageGRID의 스토리지 및 성능 요구 사항 13
스토리지 범주 13
성능 요구사항 13
NetApp ONTAP 스토리지를 사용하는 호스트에 대한 요구 사항 14
필요한 호스트 수 14
노드당 스토리지 볼륨 수 14
호스트에 필요한 최소 스토리지 공간 15
예시: 호스트 또는 가상 머신의 스토리지 요구사항 계산 16
스토리지 노드에 대한 특정 스토리지 요구 사항 18
Linux용 StorageGRID 노드 컨테이너 마이그레이션 요구 사항 19
VMware Live Migration은 지원되지 않습니다. 20
일관된 네트워크 인터페이스 이름 20
공유 스토리지 21
호스트(Linux) 준비 21
StorageGRID에서 호스트 전체 설정을 변경하는 방법 21
Linux 그리드 호스트에 StorageGRID 설치 23
Ubuntu 및 Debian에서 StorageGRID용 AppArmor 프로필에 대해 알아보십시오 25
Linux 배포를 위한 StorageGRID 호스트 네트워크 구성 25
Linux 배포를 위한 StorageGRID 호스트 스토리지 구성 30
Linux에서 StorageGRID용 컨테이너 엔진 스토리지 볼륨 구성 33
Linux에 StorageGRID 호스트 서비스를 설치합니다 36

소프트웨어 기반 노드에 대한 설치 계획 및 준비

StorageGRID에 필요한 정보 및 자료

StorageGRID를 설치하기 전에 필요한 정보와 자료를 수집하고 준비하십시오.

필수 정보

네트워크 계획

각 StorageGRID 노드에 연결할 네트워크를 선택하십시오. StorageGRID는 트래픽 분리, 보안 및 관리 편의성을 위해 여러 네트워크를 지원합니다.

StorageGRID "네트워킹 가이드라인"을 참조하십시오.

네트워크 정보

각 그리드 노드에 할당할 IP 주소와 DNS 및 NTP 서버의 IP 주소입니다.

그리드 노드용 서버

배포하려는 StorageGRID 노드의 수와 유형을 지원하기에 충분한 리소스를 집합적으로 제공하는 서버 세트(물리적, 가상 또는 둘 다)를 식별하십시오.

> StorageGRID 설치에 StorageGRID 어플라이언스(하드웨어) 스토리지 노드를 사용하지 않는 경우, 배터리 백업 쓰기 캐시(BBWC)가 있는 하드웨어 RAID 스토리지를 사용해야 합니다. StorageGRID는 가상 스토리지 영역 네트워크(vSAN), 소프트웨어 RAID 또는 RAID 보호 기능이 없는 구성을 지원하지 않습니다.

노드 마이그레이션 **(Ubuntu** 및 **Debian**에만 해당**,** 필요한 경우**)**

서비스 중단 없이 물리적 호스트에 대한 예약된 유지 보수를 수행하려면 "노드 마이그레이션 요구 사항"을(를) 이해해야 합니다.

관련 정보

"NetApp 상호 운용성 매트릭스 도구"

필수 재료

NetApp StorageGRID 라이선스

유효하고 디지털 서명이 된 NetApp 라이선스가 있어야 합니다.

> StorageGRID 설치 아카이브에는 테스트 및 개념 증명 그리드에 사용할 수 있는 비프로덕션 라이선스가 포함되어 있습니다.

StorageGRID 설치 아카이브

"StorageGRID 설치 아카이브를 다운로드하고 파일을 압축 해제합니다.".

서비스 노트북

StorageGRID 시스템은 서비스용 노트북을 통해 설치됩니다.

서비스용 노트북은 다음 조건을 충족해야 합니다.

- 네트워크 포트
- SSH 클라이언트(예: PuTTY)
- "지원되는 웹 브라우저"

StorageGRID 문서

- "릴리스 노트". NetApp 계정으로 로그인하거나 계정을 생성하여 릴리스 노트에 액세스해야 합니다.
- "StorageGRID 관리 지침"

StorageGRID 설치 파일 다운로드 및 압축 해제

StorageGRID 설치 아카이브를 다운로드하고 필요한 파일을 압축 해제해야 합니다. 필요에 따라 설치 패키지에 포함된 파일을 수동으로 확인할 수 있습니다.

단계

1. "NetApp StorageGRID 다운로드 페이지"로 이동합니다.
2. 최신 버전을 다운로드하려면 버튼을 선택하거나 드롭다운 메뉴에서 다른 버전을 선택한 후 *Go*를 선택하십시오.
3. NetApp 계정의 사용자 이름과 비밀번호로 로그인하세요.
4. 주의/MustRead 문구가 표시되면 해당 문구를 읽고 확인란을 선택하십시오.

   StorageGRID 릴리스를 설치한 후에는 필요한 핫픽스를 적용해야 합니다. 자세한 내용은 "복구 및 유지 관리 지침의 핫픽스 절차"

5. 최종 사용자 라이선스 계약을 읽고 확인란을 선택한 다음 *Accept & Continue*를 선택하십시오.
6. **StorageGRID** 설치 열에서 사용 중인 소프트웨어 기반 노드 유형(RHEL, Ubuntu 또는 Debian, 또는 VMware)에 맞는 .tgz 또는 .zip 설치 아카이브를 선택하십시오.

   서비스용 노트북에서 Windows를 실행 중인 경우 `.zip` 파일을 사용하십시오.

7. 설치 아카이브를 저장합니다.
8. 코드 서명 검증은 Linux 노드에서 수동으로 수행됩니다. 선택적으로 설치 아카이브를 검증해야 하는 경우:
   a. StorageGRID 코드 서명 검증 패키지를 다운로드하십시오. 이 패키지의 파일 이름은 `StorageGRID_<version-number>_Code_Signature_Verification_Package.tar.gz` 형식을 사용하며, 여기서 `<version-number>`은(는) StorageGRID 소프트웨어 버전입니다.
   b. 다음 단계를 따르십시오"설치 파일을 수동으로 확인합니다".
9. 설치 압축 파일에서 파일을 추출하세요.
10. 필요한 파일을 선택하십시오.

    필요한 파일은 계획된 그리드 토폴로지와 StorageGRID 시스템 배포 방식에 따라 다릅니다.

    표에 나열된 경로는 압축 해제된 설치 아카이브가 설치한 최상위 디렉터리를 기준으로 한 상대 경로입니다.

RHEL

경로 및 파일 이름	설명
	StorageGRID 다운로드 파일에 포함된 모든 파일을 설명하는 텍스트 파일입니다.
	제품에 대한 어떠한 지원 권한도 제공하지 않는 무료 라이선스입니다.
	RHEL 호스트에 StorageGRID 노드 이미지를 설치하기 위한 RPM 패키지입니다.
	RHEL 호스트에 StorageGRID 호스트 서비스를 설치하기 위한 RPM 패키지입니다.
배포 스크립팅 도구	설명
	StorageGRID 시스템 구성을 자동화하는 데 사용되는 Python 스크립트입니다.
	StorageGRID 어플라이언스의 구성을 자동화하는 데 사용되는 Python 스크립트입니다.
	`configure-storagegrid.py` 스크립트와 함께 사용할 수 있는 예제 구성 파일입니다.
	싱글 사인온(SSO)이 활성화된 경우 그리드 관리 API에 로그인하는 데 사용할 수 있는 Python 스크립트 예제입니다. 이 스크립트는 Ping Federate 통합에도 사용할 수 있습니다.
	`configure-storagegrid.py` 스크립트와 함께 사용할 빈 구성 파일입니다.
	StorageGRID 컨테이너 배포를 위해 RHEL 호스트를 구성하는 Ansible 역할 및 플레이북 예시입니다. 필요에 따라 역할 또는 플레이북을 사용자 지정할 수 있습니다.

경로 및 파일 이름	설명
	Active Directory 또는 Ping Federate를 사용하여 단일 로그인(SSO)이 활성화된 경우 그리드 관리 API에 로그인하는 데 사용할 수 있는 Python 스크립트 예제입니다.
	이 스크립트는 함께 실행되는 `storagegrid-ssoauth-azure.py`Python 스크립트에서 호출되어 Azure와의 SSO 상호 작용을 수행합니다.
	StorageGRID에 대한 API 스키마. 참고: 업그레이드를 수행하기 전에, 업그레이드 호환성 테스트를 위한 비프로덕션 StorageGRID 환경이 없는 경우, 이러한 스키마를 사용하여 StorageGRID 관리 API를 사용하도록 작성한 코드가 새 StorageGRID 릴리스와 호환되는지 확인할 수 있습니다.

Ubuntu 또는 **Debian**

경로 및 파일 이름	설명
	StorageGRID 다운로드 파일에 포함된 모든 파일을 설명하는 텍스트 파일입니다.
	테스트 및 개념 증명 배포에 사용할 수 있는 비프로덕션 NetApp 라이선스 파일입니다.
	Ubuntu 또는 Debian 호스트에 StorageGRID 노드 이미지를 설치하기 위한 DEB 패키지입니다.
	파일의 MD5 체크섬 `/debs/storagegrid-webscale-images-version-SHA.deb`입니다.
	Ubuntu 또는 Debian 호스트에 StorageGRID 호스트 서비스를 설치하기 위한 DEB 패키지입니다.
배포 스크립팅 도구	설명
	StorageGRID 시스템 구성을 자동화하는 데 사용되는 Python 스크립트입니다.
	StorageGRID 어플라이언스의 구성을 자동화하는 데 사용되는 Python 스크립트입니다.

경로 및 파일 이름	설명
	싱글 사인온(SSO)이 활성화된 경우 그리드 관리 API에 로그인하는 데 사용할 수 있는 Python 스크립트 예제입니다. 이 스크립트는 Ping Federate 통합에도 사용할 수 있습니다.
	`configure-storagegrid.py` 스크립트와 함께 사용할 수 있는 예제 구성 파일입니다.
	`configure-storagegrid.py` 스크립트와 함께 사용할 빈 구성 파일입니다.
	StorageGRID 컨테이너 배포를 위해 Ubuntu 또는 Debian 호스트를 구성하는 Ansible 역할 및 플레이북 예시입니다. 필요에 따라 역할 또는 플레이북을 사용자 지정할 수 있습니다.
	Active Directory 또는 Ping Federate를 사용하여 단일 로그인(SSO)이 활성화된 경우 그리드 관리 API에 로그인하는 데 사용할 수 있는 Python 스크립트 예제입니다.
	이 스크립트는 함께 실행되는 `storagegrid-ssoauth-azure.py`Python 스크립트에서 호출되어 Azure와의 SSO 상호 작용을 수행합니다.
	StorageGRID에 대한 API 스키마. 참고: 업그레이드를 수행하기 전에, 업그레이드 호환성 테스트를 위한 비프로덕션 StorageGRID 환경이 없는 경우, 이러한 스키마를 사용하여 StorageGRID 관리 API를 사용하도록 작성한 코드가 새 StorageGRID 릴리스와 호환되는지 확인할 수 있습니다.

VMware

경로 및 파일 이름	설명
	StorageGRID 다운로드 파일에 포함된 모든 파일을 설명하는 텍스트 파일입니다.

경로 및 파일 이름	설명
	제품에 대한 어떠한 지원 권한도 제공하지 않는 무료 라이선스입니다.
	그리드 노드 가상 머신을 생성하기 위한 템플릿으로 사용되는 가상 머신 디스크 파일입니다.
	기본 관리 노드를 배포하기 위한 Open Virtualization Format 템플릿 파일((`.ovf`)과 매니페스트 파일((`.mf`)입니다.
	기본이 아닌 관리 노드를 배포하기 위한 템플릿 파일((`.ovf`)과 매니페스트 파일((`.mf`)입니다.
	게이트웨이 노드를 배포하기 위한 템플릿 파일 (`.ovf`) 및 매니페스트 파일 (`.mf`)입니다.
	가상 머신 기반 스토리지 노드를 배포하기 위한 템플릿 파일((`.ovf`)과 매니페스트 파일((`.mf`)입니다.
배포 스크립팅 도구	설명
	가상 그리드 노드 배포를 자동화하는 데 사용되는 Bash 셸 스크립트입니다.
	`deploy-vsphere-ovftool.sh` 스크립트와 함께 사용할 수 있는 예제 구성 파일입니다.
	StorageGRID 시스템 구성을 자동화하는 데 사용되는 Python 스크립트입니다.
	StorageGRID 어플라이언스의 구성을 자동화하는 데 사용되는 Python 스크립트입니다.
	싱글 사인온(SSO)이 활성화된 경우 그리드 관리 API에 로그인하는 데 사용할 수 있는 Python 스크립트 예제입니다. 이 스크립트는 Ping Federate 통합에도 사용할 수 있습니다.

경로 및 파일 이름	설명
	`configure-storagegrid.py` 스크립트와 함께 사용할 수 있는 예제 구성 파일입니다.
	`configure-storagegrid.py` 스크립트와 함께 사용할 빈 구성 파일입니다.
	Active Directory 또는 Ping Federate를 사용하여 단일 로그인(SSO)이 활성화된 경우 그리드 관리 API에 로그인하는 데 사용할 수 있는 Python 스크립트 예제입니다.
	이 스크립트는 함께 실행되는 `storagegrid-ssoauth-azure.py`Python 스크립트에서 호출되어 Azure와의 SSO 상호 작용을 수행합니다.
	StorageGRID에 대한 API 스키마. 참고: 업그레이드를 수행하기 전에, 업그레이드 호환성 테스트를 위한 비프로덕션 StorageGRID 환경이 없는 경우, 이러한 스키마를 사용하여 StorageGRID 관리 API를 사용하도록 작성한 코드가 새 StorageGRID 릴리스와 호환되는지 확인할 수 있습니다.

StorageGRID 설치 파일을 확인하십시오

필요한 경우 StorageGRID 설치 아카이브에 있는 파일을 수동으로 확인할 수 있습니다.

시작하기 전에

다음에서 "검증 패키지를 다운로드했습니다."을(를) 받았습니다. "NetApp StorageGRID 다운로드 페이지"

단계

1. 검증 패키지에서 아티팩트를 추출합니다.

```
tar -xf StorageGRID_12.0.0_Code_Signature_Verification_Package.tar.gz
```

2. 이러한 아티팩트가 추출되었는지 확인하십시오.
   - 리프 인증서: `Leaf-Cert.pem`
   - 인증서 체인: `CA-Int-Cert.pem`

- 타임스탬프 응답 체인: `TS-Cert.pem`
- 체크섬 파일: `sha256sum`
- 체크섬 서명: `sha256sum.sig`
- 타임스탬프 응답 파일: `sha256sum.sig.tsr`

3. 체인을 사용하여 리프 인증서가 유효한지 확인하십시오.

   예: `openssl verify -CAfile CA-Int-Cert.pem Leaf-Cert.pem`

   예상 출력: `Leaf-Cert.pem: OK`

4. 2단계에서 리프 인증서가 만료되어 실패한 경우, `tsr` 파일을 사용하여 확인하십시오.

   예: `openssl ts -CAfile CA-Int-Cert.pem -untrusted TS-Cert.pem -verify -data sha256sum.sig -in sha256sum.sig.tsr`

   예상 출력 내용: `Verification: OK`

5. 리프 인증서에서 공개 키 파일을 생성합니다.

   예: `openssl x509 -pubkey -noout -in Leaf-Cert.pem > Leaf-Cert.pub`

   예상 출력: 없음

6. 공개 키를 사용하여 `sha256sum` 파일을 `sha256sum.sig`에 대해 검증하십시오.

   예: `openssl dgst -sha256 -verify Leaf-Cert.pub -signature sha256sum.sig sha256sum`

   예상 출력: `Verified OK`

7. 새로 생성된 체크섬과 `sha256sum` 파일 내용을 대조하여 검증합니다.

   예: `sha256sum -c sha256sum`

   예상 출력: `<filename>: OK`
   `<filename>`는 다운로드한 압축 파일의 이름입니다.

8. "나머지 단계를 완료합니다" 설치 아카이브에서 적절한 파일을 추출하고 선택합니다.

StorageGRID의 소프트웨어 요구 사항

가상 머신을 사용하여 모든 유형의 StorageGRID 노드를 호스팅할 수 있습니다. 그리드 노드 하나당 가상 머신이 하나씩 필요합니다.

RHEL

RHEL에 StorageGRID를 설치하려면 일부 타사 소프트웨어 패키지를 설치해야 합니다. 일부 지원되는 Linux 배포판에는 이러한 패키지가 기본적으로 포함되어 있지 않습니다. StorageGRID 설치 테스트에 사용된 소프트웨어 패키지 버전은 이 페이지에 나열되어 있습니다.

Linux 배포판 및 컨테이너 런타임 설치 옵션에서 이러한 패키지 중 하나라도 필요로 하는 경우, 해당 패키지가 Linux 배포판에서 자동으로 설치되지 않으면, 공급업체 또는 Linux 배포판 지원 업체에서 제공하는 여기에 나열된 버전 중 하나를 설치하십시오. 그렇지 않으면 공급업체에서 제공하는 기본 패키지 버전을 사용하십시오.

모든 설치 옵션에는 Podman 또는 Docker 중 하나가 필요합니다. 두 패키지를 모두 설치하지 마십시오. 설치 옵션에 필요한 패키지만 설치하십시오.

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

테스트에 사용된 **Python** 버전

- 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 버전 테스트 완료

- 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

테스트 완료된 **Docker** 버전

> ⓘ Docker 지원은 더 이상 사용되지 않으며 향후 릴리스에서 제거될 예정입니다.

- 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

Ubuntu 및 **Debian**

Ubuntu 또는 Debian에 StorageGRID를 설치하려면 일부 타사 소프트웨어 패키지를 설치해야 합니다. 일부 지원되는 Linux 배포판에는 이러한 패키지가 기본적으로 포함되어 있지 않습니다. StorageGRID 설치 테스트에 사용된 소프트웨어 패키지 버전은 이 페이지에 나열되어 있습니다.

Linux 배포판 및 컨테이너 런타임 설치 옵션에서 이러한 패키지 중 하나라도 필요로 하는 경우, 해당 패키지가 Linux 배포판에서 자동으로 설치되지 않으면, 공급업체 또는 Linux 배포판 지원 업체에서 제공하는 여기에 나열된 버전 중 하나를 설치하십시오. 그렇지 않으면 공급업체에서 제공하는 기본 패키지 버전을 사용하십시오.

모든 설치 옵션에는 Podman 또는 Docker 중 하나가 필요합니다. 두 패키지를 모두 설치하지 마십시오. 설치 옵션에 필요한 패키지만 설치하십시오.

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

테스트에 사용된 **Python** 버전

- 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 버전 테스트 완료

- 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

테스트 완료된 **Docker** 버전

Docker 지원은 더 이상 사용되지 않으며 향후 릴리스에서 제거될 예정입니다.

- 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

VMware

VMware vSphere 하이퍼바이저

VMware vSphere Hypervisor를 준비된 물리적 서버에 설치해야 합니다. VMware 소프트웨어를 설치하기 전에 하드웨어가 올바르게 구성되어 있어야 합니다(펌웨어 버전 및 BIOS 설정 포함).

- 설치하려는 StorageGRID 시스템의 네트워킹을 지원하도록 하이퍼바이저에서 필요한 네트워킹을 구성하십시오.

  "네트워킹 가이드라인"

- 그리드 노드를 호스팅하는 데 필요한 가상 머신과 가상 디스크를 저장할 수 있을 만큼 데이터 저장소의 크기가 충분한지 확인하십시오.
- 데이터 저장소를 두 개 이상 생성하는 경우, 가상 머신을 생성할 때 각 그리드 노드에 사용할 데이터 저장소를 쉽게 식별할 수 있도록 각 데이터 저장소에 이름을 지정하십시오.

ESX 호스트 구성 요구 사항

각 ESX 호스트에서 NTP(네트워크 시간 프로토콜)를 올바르게 구성해야 합니다. 호스트 시간이 올바르지 않으면 데이터 손실을 포함한 부정적인 영향이 발생할 수 있습니다.

VMware 구성 요구 사항

StorageGRID 노드를 배포하기 전에 VMware vSphere 및 vCenter를 설치하고 구성해야 합니다.

지원되는 VMware vSphere 하이퍼바이저 및 VMware vCenter Server 소프트웨어 버전에 대한 자세한 내용은 "NetApp 상호 운용성 매트릭스 도구"을 참조하십시오.

이러한 VMware 제품을 설치하는 데 필요한 단계는 VMware 설명서를 참조하십시오.

StorageGRID의 CPU 및 RAM 요구 사항

StorageGRID 소프트웨어를 설치하기 전에 하드웨어가 StorageGRID 시스템을 지원할 수 있도록 준비되었는지 확인하고 구성하십시오.

각 StorageGRID 노드에는 다음과 같은 최소 리소스가 필요합니다.

- CPU 코어: 노드당 8개
- RAM: 사용 가능한 총 RAM 용량과 시스템에서 실행 중인 non-StorageGRID 소프트웨어 양에 따라 달라집니다.
  - 일반적으로 노드당 최소 24GB, 그리고 전체 시스템 RAM보다 2~16GB 적은 용량이 필요합니다.
  - 약 5,000개의 버킷을 보유할 각 테넌트에 대해 최소 64GB

소프트웨어 기반 메타데이터 전용 노드 리소스는 기존 스토리지 노드 리소스와 일치해야 합니다. 예를 들면 다음과 같습니다.

- 기존 StorageGRID 사이트에서 SG6000 또는 SG6100 어플라이언스를 사용하는 경우, 소프트웨어 기반 메타데이터 전용 노드는 다음 최소 요구 사항을 충족해야 합니다.
  - 128GB RAM
  - 8코어 CPU
  - Cassandra 데이터베이스(rangedb/0)용 8TB SSD 또는 동급 스토리지
- 기존 StorageGRID 사이트에서 24GB RAM, 8코어 CPU, 3TB 또는 4TB의 메타데이터 스토리지를 갖춘 가상 스토리지 노드를 사용하는 경우 소프트웨어 기반 메타데이터 전용 노드도 유사한 리소스(24GB RAM, 8코어 CPU, 4TB의 메타데이터 스토리지(rangedb/0))를 사용해야 합니다.

새로운 StorageGRID 사이트를 추가할 때, 새 사이트의 총 메타데이터 용량은 최소한 기존 StorageGRID 사이트와 일치해야 하며, 새 사이트의 리소스는 기존 StorageGRID 사이트의 스토리지 노드와 일치해야 합니다.

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

Linux

각 물리적 또는 가상 호스트에서 실행할 StorageGRID 노드 수가 사용 가능한 CPU 코어 수 또는 물리적 RAM을 초과하지 않도록 하십시오. 호스트가 StorageGRID 실행 전용이 아닌 경우(권장하지 않음) 다른 애플리케이션의 리소스 요구 사항도 고려해야 합니다.

VMware

VMware는 가상 머신당 하나의 노드를 지원합니다. StorageGRID 노드가 사용 가능한 물리적 RAM을 초과하지 않도록 하십시오. 각 가상 머신은 StorageGRID 실행 전용이어야 합니다.

CPU 및 메모리 사용량을 정기적으로 모니터링하여 이러한 리소스가 워크로드를 지속적으로 수용할 수 있도록 하십시오. 예를 들어 가상 스토리지 노드의 RAM 및 CPU 할당량을 두 배로 늘리면 StorageGRID 어플라이언스 노드에 제공되는 것과 유사한 리소스를 확보할 수 있습니다. 또한 노드당 메타데이터 양이 500GB를 초과하는 경우 노드당 RAM을 48GB 이상으로 늘리는 것을 고려하십시오. 객체 메타데이터 스토리지 관리, 메타데이터 예약 공간 설정 증가, CPU 및 메모리 사용량 모니터링에 대한 자세한 내용은 "관리", "모니터링" 및 "업그레이드" StorageGRID에 대한 지침을 참조하십시오.

기본 물리적 호스트에서 하이퍼스레딩이 활성화된 경우 노드당 8개의 가상 코어(4개의 물리적 코어)를 제공할 수 있습니다. 기본 물리적 호스트에서 하이퍼스레딩이 활성화되지 않은 경우 노드당 8개의 물리적 코어를 제공해야 합니다.

가상 머신을 호스트로 사용하고 VM의 크기와 개수를 제어할 수 있는 경우, StorageGRID 노드마다 하나의 VM을 사용하고 VM의 크기를 그에 맞게 조정해야 합니다.

(RHEL, Debian 및 Ubuntu에만 해당) 운영 구축 환경에서는 동일한 물리적 스토리지 하드웨어 또는 가상 호스트에서 여러 스토리지 노드를 실행해서는 안 됩니다. 단일 StorageGRID 배포 환경의 각 스토리지 노드는 자체적으로 격리된 장애 도메인에 있어야 합니다. 단일 하드웨어 장애가 단일 스토리지 노드에만 영향을 미치도록 하면 객체 데이터의 내구성과 가용성을 극대화할 수 있습니다.

참고 항목: "스토리지 및 성능 요구 사항".

StorageGRID의 스토리지 및 성능 요구 사항

StorageGRID 노드의 스토리지 요구 사항을 이해해야 초기 구성 및 향후 스토리지 확장을 지원하기에 충분한 공간을 제공할 수 있습니다.

스토리지 및 성능 요구 사항은 소프트웨어 기반 노드 구현 방식에 따라 다릅니다.

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

스토리지 범주

StorageGRID 노드에는 세 가지 논리적 범주의 스토리지가 필요합니다.

- 컨테이너 풀 — 노드 컨테이너용 성능 등급(10K SAS 또는 SSD) 스토리지입니다. 이 스토리지는 StorageGRID 노드를 지원할 호스트에 컨테이너 엔진을 설치 및 구성할 때 컨테이너 엔진 스토리지 드라이버에 할당됩니다.
- 시스템 데이터 — 시스템 데이터 및 트랜잭션 로그의 노드별 영구 스토리지를 위한 성능 등급(10K SAS 또는 SSD) 스토리지로, StorageGRID 호스트 서비스가 이를 사용하여 개별 노드에 매핑합니다.
- 객체 데이터 — 객체 데이터 및 객체 메타데이터의 영구 스토리지를 위한 성능 계층(10K SAS 또는 SSD) 스토리지 및 용량 계층(NL-SAS/SATA) 대용량 스토리지.

모든 스토리지 범주에는 RAID 기반 블록 장치를 사용해야 합니다. 비중복 디스크, SSD 또는 JBOD는 지원되지 않습니다. 공유 또는 로컬 RAID 스토리지를 모든 스토리지 범주에 사용할 수 있지만, StorageGRID에서 노드 마이그레이션 기능을 사용하려면 시스템 데이터와 객체 데이터를 모두 공유 스토리지에 저장해야 합니다. 자세한 내용은 "Node 컨테이너 마이그레이션 요구 사항"을 참조하십시오.

성능 요구사항

컨테이너 풀, 시스템 데이터 및 객체 메타데이터에 사용되는 볼륨의 성능은 시스템의 전반적인 성능에 상당한 영향을 미칩니다. 지연 시간, 초당 입출력 작업 수(IOPS) 및 처리량 측면에서 적절한 디스크 성능을 확보하려면 이러한 볼륨에는 성능 등급(10K SAS 또는 SSD) 스토리지를 사용해야 합니다. 객체 데이터의 영구 스토리지에는 용량 등급(NL-SAS/SATA) 스토리지를 사용할 수 있습니다.

컨테이너 풀, 시스템 데이터 및 객체 데이터에 사용되는 볼륨에는 쓰기 캐시가 활성화되어 있어야 합니다. 캐시는 보호된 미디어 또는 영구 미디어에 있어야 합니다.

NetApp ONTAP 스토리지를 사용하는 호스트에 대한 요구 사항

StorageGRID 노드가 NetApp ONTAP 시스템에서 할당된 스토리지를 사용하는 경우 해당 볼륨에 FabricPool 티어링 정책이 활성화되어 있지 않은지 확인하십시오. StorageGRID 노드에서 사용하는 볼륨에 대해 FabricPool 티어링을 비활성화하면 문제 해결 및 스토리지 운영이 간소화됩니다.

> FabricPool를 사용하여 StorageGRID와 관련된 데이터를 StorageGRID 자체로 계층화하지 마십시오. StorageGRID 데이터를 StorageGRID로 다시 계층화하면 문제 해결 및 운영 복잡성이 증가합니다.

필요한 호스트 수

각 StorageGRID 사이트에는 최소 3개의 스토리지 노드가 필요합니다.

> 운영 구축 시 단일 물리적 또는 가상 호스트에서 스토리지 노드를 두 개 이상 실행하지 마십시오. 각 스토리지 노드에 전용 호스트를 사용하면 장애 발생 시 격리된 환경을 구축할 수 있습니다.

${post_edited_translations.segment}

> 디스크 스냅샷은 그리드 노드를 복원하는 데 사용할 수 없습니다. 대신 "그리드 노드 복구" 각 노드 유형에 대한 절차를 참조하십시오.

노드당 스토리지 볼륨 수

다음 표는 각 호스트에 필요한 스토리지 볼륨(LUN)의 수와 각 LUN에 필요한 최소 크기를 보여줍니다. 이는 해당 호스트에 배포될 노드를 기준으로 합니다.

테스트된 최대 LUN 크기는 39TB입니다.

> 이 수치는 전체 그리드가 아닌 각 호스트에 대한 수치입니다.

${post_edited_translations.segment}	**${post_edited_translations.segment}**	**LUN** 개수	최소 크기/**LUN**
컨테이너 엔진 스토리지 풀	컨테이너 풀	1	${post_edited_translations.segment}
`/var/local` 볼륨	시스템 데이터	이 호스트의 각 노드마다 1개씩	100 GB

${post_edited_translations.segment}	${post_edited_translations.segment}	LUN 개수	최소 크기/LUN
스토리지 노드	객체 데이터	${post_edited_translations.segment} 참고: Linux 소프트웨어 기반 스토리지 노드는 1~48개의 스토리지 볼륨을 가질 수 있습니다. VMware 소프트웨어 기반 스토리지 노드는 1~16개의 스토리지 볼륨을 가질 수 있습니다. 최소 3개의 스토리지 볼륨을 권장합니다.	12 TB (최소 4 TB/LUN) 최대 테스트 LUN 크기: 39TB. 자세한 내용은 스토리지 노드에 필요한 스토리지 용량을 참조하십시오.
스토리지 노드(메타데이터 전용)	객체 메타데이터	1	최소 4TB/LUN 최대 테스트 LUN 크기: 39TB. 자세한 내용은 스토리지 노드에 필요한 스토리지 용량을 참조하십시오. 참고: 메타데이터 전용 스토리지 노드에는 rangedb가 하나만 필요합니다.
관리자 노드 감사 로그	시스템 데이터	이 호스트의 각 관리 노드마다 1개	200 GB
관리자 노드 테이블	시스템 데이터	이 호스트의 각 관리 노드마다 1개	200 GB

구성된 감사 수준, S3 객체 키 이름과 같은 사용자 입력 크기, 보존해야 하는 감사 로그 데이터 양에 따라 각 관리 노드의 감사 로그 LUN 크기를 늘려야 할 수 있습니다. 일반적으로 그리드 환경에서는 S3 작업당 약 1KB의 감사 데이터가 생성되므로 200GB LUN은 하루 7천만 건의 작업 또는 초당 800건의 작업을 2~3일 동안 지원할 수 있습니다.

호스트에 필요한 최소 스토리지 공간

다음 표는 각 노드 유형에 필요한 최소 스토리지 공간을 보여줍니다. 이 표를 사용하여 호스트에 배포할 노드 유형에 따라 각 스토리지 범주별로 호스트에 제공해야 하는 최소 스토리지 공간을 확인할 수 있습니다.

디스크 스냅샷은 그리드 노드를 복원하는 데 사용할 수 없습니다. 대신 "그리드 노드 복구" 각 노드 유형에 대한 절차를 참조하십시오.

각 노드 호스트에는 운영 체제용으로 100GB LUN이 필요합니다.

노드 유형	컨테이너 풀	시스템 데이터	객체 데이터
스토리지 노드	100 GB	100 GB	4,000 GB
관리자 노드	100 GB	500GB (LUN 3개)	해당 사항 없음
게이트웨이 노드	100 GB	100 GB	해당 사항 없음

예시: 호스트 또는 가상 머신의 스토리지 요구사항 계산

동일한 호스트 또는 가상 머신에 스토리지 노드, 관리 노드, 게이트웨이 노드 이렇게 세 개의 노드를 배포할 계획이라고 가정해 보겠습니다. 호스트에는 최소 9개의 스토리지 볼륨을 제공해야 합니다. 노드 컨테이너에는 최소 300GB의 성능 계층 스토리지가, 시스템 데이터 및 트랜잭션 로그에는 700GB의 성능 계층 스토리지가, 객체 데이터에는 12TB의 용량 계층 스토리지가 필요합니다.

Linux 호스트 예시

노드 유형	**${post_edited_translations.segment}**	**LUN** 개수	**LUN** 크기
스토리지 노드	컨테이너 엔진 스토리지 풀	1	${post_edited_translations.segment}
스토리지 노드	`/var/local` 볼륨	1	100 GB
스토리지 노드	객체 데이터	3	${post_edited_translations.segment}
관리자 노드	`/var/local` 볼륨	1	100 GB
관리자 노드	관리자 노드 감사 로그	1	200 GB
관리자 노드	관리자 노드 테이블	1	200 GB
게이트웨이 노드	`/var/local` 볼륨	1	100 GB
총		**9**	컨테이너 풀**:** 300 GB 시스템 데이터**:** 700 GB 객체 데이터**:** 12,000 GB

VMware 가상 머신 예시

노드 유형	**${post_edited_translations.segment}**	**LUN** 개수	**LUN** 크기
스토리지 노드	OS 볼륨	1	100 GB
스토리지 노드	객체 데이터	3	${post_edited_translations.segment}
관리자 노드	OS 볼륨	1	100 GB
관리자 노드	관리자 노드 감사 로그	1	200 GB
관리자 노드	관리자 노드 테이블	1	200 GB
게이트웨이 노드	OS 볼륨	1	100 GB

노드 유형	**${post_edited_translations.segment}**	**LUN** 개수	**LUN** 크기
총		**8**	시스템 데이터**:** 700 GB 객체 데이터**:** 12,000 GB

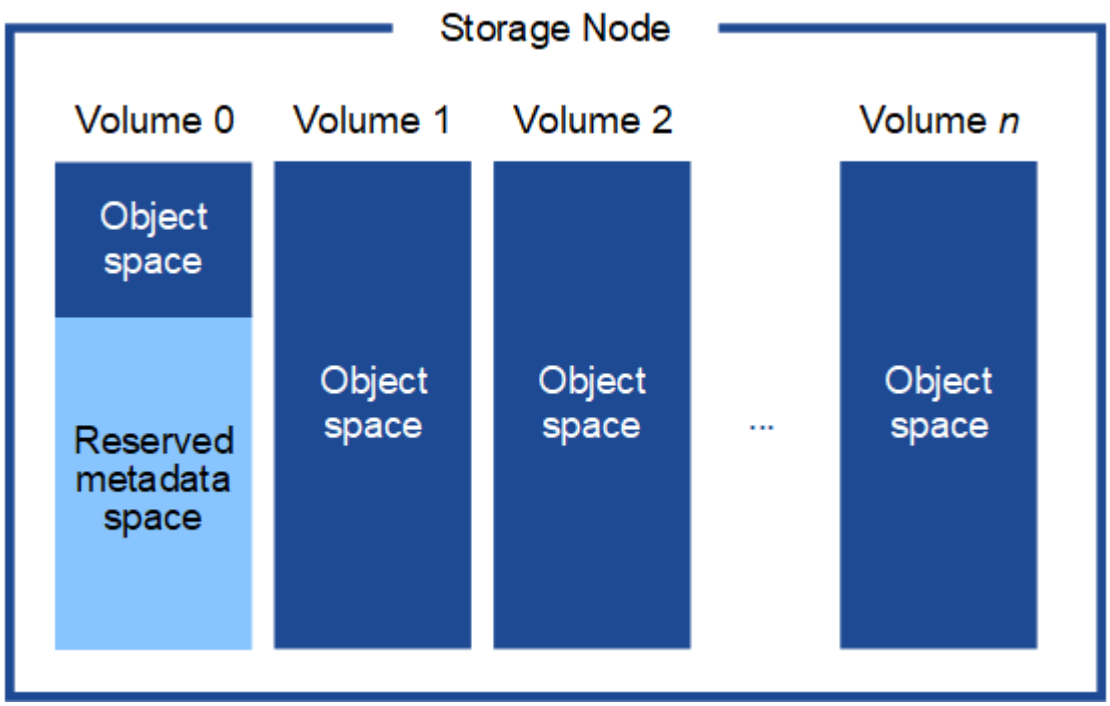
스토리지 노드에 대한 특정 스토리지 요구 사항

Linux와 VMware는 스토리지 노드에 대한 스토리지 요구 사항이 다릅니다.

- ${post_edited_translations.segment}
- VMware 소프트웨어 기반 스토리지 노드는 1개에서 16개까지의 스토리지 볼륨을 가질 수 있습니다.
- 3개 이상의 스토리지 볼륨을 권장합니다.
- 각 스토리지 볼륨은 4TB 이상이어야 합니다.

어플라이언스 스토리지 노드는 최대 48개의 스토리지 볼륨을 가질 수 있습니다.

그림에서 보는 바와 같이 StorageGRID는 각 스토리지 노드의 스토리지 볼륨 0에 객체 메타데이터를 위한 공간을 예약합니다. 스토리지 볼륨 0 및 스토리지 노드의 다른 스토리지 볼륨에 남아 있는 공간은 모두 객체 데이터 저장에만 사용됩니다.



이중화를 제공하고 객체 메타데이터를 손실로부터 보호하기 위해 StorageGRID는 시스템 내 모든 객체의 메타데이터 사본 3개를 각 사이트에 저장합니다. 이 3개의 객체 메타데이터 사본은 각 사이트의 모든 스토리지 노드에 고르게 분산됩니다.

메타데이터 전용 스토리지 노드를 사용하여 그리드를 설치할 경우, 그리드에는 객체 스토리지를 위한 최소 개수의 노드도 포함되어야 합니다. 메타데이터 전용 스토리지 노드에 대한 자세한 내용은 "스토리지 노드의 유형"을

참조하십시오.

- 단일 사이트 그리드의 경우 객체 및 메타데이터에 대해 최소 두 개의 스토리지 노드가 구성됩니다.
- ${post_edited_translations.segment}

${post_edited_translations.segment}

- ${post_edited_translations.segment}

> 스토리지 노드에 스토리지 볼륨을 하나만 사용하고 해당 볼륨에 4TB 이하의 용량을 할당하는 경우, 스토리지 노드는 시작 시 스토리지 읽기 전용 상태로 전환되어 객체 메타데이터만 저장할 수 있습니다.

> 볼륨 0(비프로덕션 용도 전용)에 500GB 미만의 용량을 할당하면 스토리지 볼륨 용량의 10%가 메타데이터용으로 예약됩니다.

- 소프트웨어 기반 메타데이터 전용 노드 리소스는 기존 스토리지 노드 리소스와 일치해야 합니다. 예를 들면 다음과 같습니다.
  - 기존 StorageGRID 사이트에서 SG6000 또는 SG6100 어플라이언스를 사용하는 경우, 소프트웨어 기반 메타데이터 전용 노드는 다음 최소 요구 사항을 충족해야 합니다.
    - 128GB RAM
    - 8코어 CPU
    - Cassandra 데이터베이스(rangedb/0)용 8TB SSD 또는 동급 스토리지
  - 기존 StorageGRID 사이트에서 24GB RAM, 8코어 CPU, 3TB 또는 4TB의 메타데이터 스토리지를 갖춘 가상 스토리지 노드를 사용하는 경우 소프트웨어 기반 메타데이터 전용 노드도 유사한 리소스(24GB RAM, 8코어 CPU, 4TB의 메타데이터 스토리지(rangedb/0))를 사용해야 합니다.

    새로운 StorageGRID 사이트를 추가할 때, 새 사이트의 총 메타데이터 용량은 최소한 기존 StorageGRID 사이트와 일치해야 하며, 새 사이트의 리소스는 기존 StorageGRID 사이트의 스토리지 노드와 일치해야 합니다.

- 새 시스템(StorageGRID 11.6 이상)을 설치하고 각 스토리지 노드에 128GB 이상의 RAM이 있는 경우, 볼륨 0에 8TB 이상을 할당하십시오. 볼륨 0에 더 큰 값을 사용하면 각 스토리지 노드의 메타데이터에 허용되는 공간을 늘릴 수 있습니다.
- 사이트에 대해 서로 다른 스토리지 노드를 구성할 때 가능하면 볼륨 0에 대해 동일한 설정을 사용하십시오. 사이트에 크기가 다른 스토리지 노드가 포함된 경우 볼륨 0의 크기가 가장 작은 스토리지 노드가 해당 사이트의 메타데이터 용량을 결정합니다.

자세한 내용은 "객체 메타데이터 저장소 관리"을 참조하십시오.

Linux용 StorageGRID 노드 컨테이너 마이그레이션 요구 사항

노드 마이그레이션 기능을 사용하면 노드를 한 호스트에서 다른 호스트로 수동으로 이동할 수 있습니다. 일반적으로 두 호스트는 동일한 물리적 데이터 센터에 있습니다.

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

노드 마이그레이션을 통해 그리드 운영을 중단하지 않고 물리적 호스트 유지 관리를 수행할 수 있습니다. 모든 StorageGRID 노드를 한 번에 하나씩 다른 호스트로 이동한 후 물리적 호스트를 오프라인으로 전환합니다. 노드 마이그레이션에는 각 노드당 짧은 다운타임만 필요하며 그리드 서비스의 운영이나 가용성에 영향을 미치지 않습니다.

StorageGRID 노드 마이그레이션 기능을 사용하려면 배포 환경이 추가 요구 사항을 충족해야 합니다.

- 단일 물리적 데이터 센터 내 호스트 전체에서 일관된 네트워크 인터페이스 이름
- StorageGRID 메타데이터 및 객체 리포지토리 볼륨을 위한 공유 스토리지로, 단일 물리적 데이터 센터 내의 모든 호스트에서 액세스할 수 있습니다. 예를 들어 NetApp E-Series 스토리지 어레이를 사용할 수 있습니다.

가상 호스트를 사용하고 있고 기본 하이퍼바이저 계층에서 VM 마이그레이션을 지원하는 경우, StorageGRID의 노드 마이그레이션 기능 대신 이 기능을 사용하는 것이 좋습니다. 이 경우 추가 요구 사항은 무시해도 됩니다.

마이그레이션 또는 하이퍼바이저 유지 관리를 수행하기 전에 노드를 정상적으로 종료하십시오. 자세한 내용은 "그리드 노드 종료"에 대한 지침을 참조하십시오.

VMware Live Migration은 지원되지 않습니다.

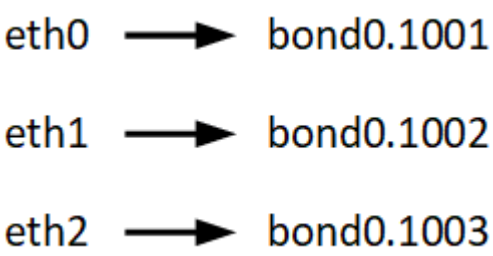
VMware VM에 베어메탈 설치를 수행할 때 OpenStack Live Migration 및 VMware live vMotion은 가상 머신의 시계 시간을 변경시키며, 모든 유형의 그리드 노드에서 지원되지 않습니다. 드물지만 시계 시간이 잘못되면 데이터 손실이나 구성 업데이트 오류가 발생할 수 있습니다.

콜드 마이그레이션이 지원됩니다. 콜드 마이그레이션에서는 StorageGRID 노드를 종료한 후 호스트 간에 마이그레이션합니다. 자세한 내용은 "그리드 노드 종료" 지침을 참조하십시오.

일관된 네트워크 인터페이스 이름

노드를 한 호스트에서 다른 호스트로 이동하려면 StorageGRID 호스트 서비스는 노드가 현재 위치에서 사용하는 외부 네트워크 연결이 새 위치에서도 동일하게 유지될 수 있다는 확신을 가져야 합니다. 이러한 확신은 호스트에서 일관된 네트워크 인터페이스 이름을 사용함으로써 확보됩니다.

예를 들어 Host1에서 실행 중인 StorageGRID NodeA가 다음과 같은 인터페이스 매핑으로 구성되었다고 가정해 보겠습니다.



화살표의 왼쪽은 StorageGRID 컨테이너 내에서 본 기존 인터페이스(즉, 각각 Grid, Admin 및 Client Network 인터페이스)에 해당합니다. 화살표의 오른쪽은 이러한 네트워크를 제공하는 실제 호스트 인터페이스로, 동일한 물리적 인터페이스 본드에 종속된 세 개의 VLAN 인터페이스입니다.

이제 NodeA를 Host2로 마이그레이션한다고 가정해 보겠습니다. Host2에도 bond0.1001, bond0.1002, bond0.1003과 같은 이름의 인터페이스가 있다면, 시스템은 동일한 이름의 인터페이스가 Host1에서와 마찬가지로 Host2에서도 동일한 연결성을 제공한다고 판단하여 마이그레이션을 허용합니다. 하지만 Host2에 동일한 이름의 인터페이스가 없다면 마이그레이션은 허용되지 않습니다.

여러 호스트에서 일관된 네트워크 인터페이스 이름을 지정하는 방법은 여러 가지가 있습니다. 몇 가지 예는 "호스트 네트워크 구성"을 참조하십시오.

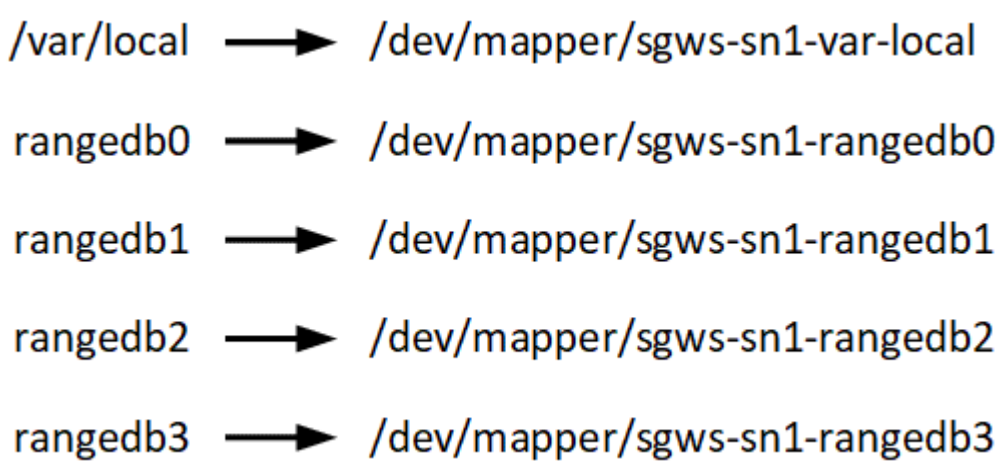
공유 스토리지

StorageGRID 노드 마이그레이션 기능은 빠르고 오버헤드가 적은 노드 마이그레이션을 구현하기 위해 노드 데이터를 물리적으로 이동하지 않습니다. 대신, 노드 마이그레이션은 다음과 같은 내보내기 및 가져오기 작업 쌍으로 수행됩니다.

- "노드 내보내기" 작업 중에 HostA에서 실행 중인 노드 컨테이너에서 소량의 영구 상태 데이터가 추출되어 해당 노드의 시스템 데이터 볼륨에 캐시됩니다. 그런 다음 HostA의 노드 컨테이너가 인스턴스 해제됩니다.
- "노드 가져오기" 작업 중에는 HostA에서 사용되던 것과 동일한 네트워크 인터페이스 및 블록 스토리지 매핑을 사용하는 HostB의 노드 컨테이너가 인스턴스화됩니다. 그런 다음 캐시된 영구 상태 데이터가 새 인스턴스에 삽입됩니다.

이러한 운영 모드에서 마이그레이션이 허용되고 정상적으로 작동하려면 노드의 모든 시스템 데이터 및 객체 스토리지 볼륨에 HostA와 HostB 모두에서 액세스할 수 있어야 합니다. 또한, HostA와 HostB에서 동일한 LUN을 참조하도록 보장된 이름을 사용하여 노드에 매핑되어야 합니다.

다음 예는 StorageGRID 스토리지 노드의 블록 장치 매핑에 대한 한 가지 솔루션을 보여줍니다. 이 솔루션에서는 호스트에서 DM 멀티패싱이 사용되고 있으며, 별칭 필드가 `/etc/multipath.conf`에서 사용되어 모든 호스트에서 일관되고 친숙한 블록 장치 이름을 제공합니다.



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

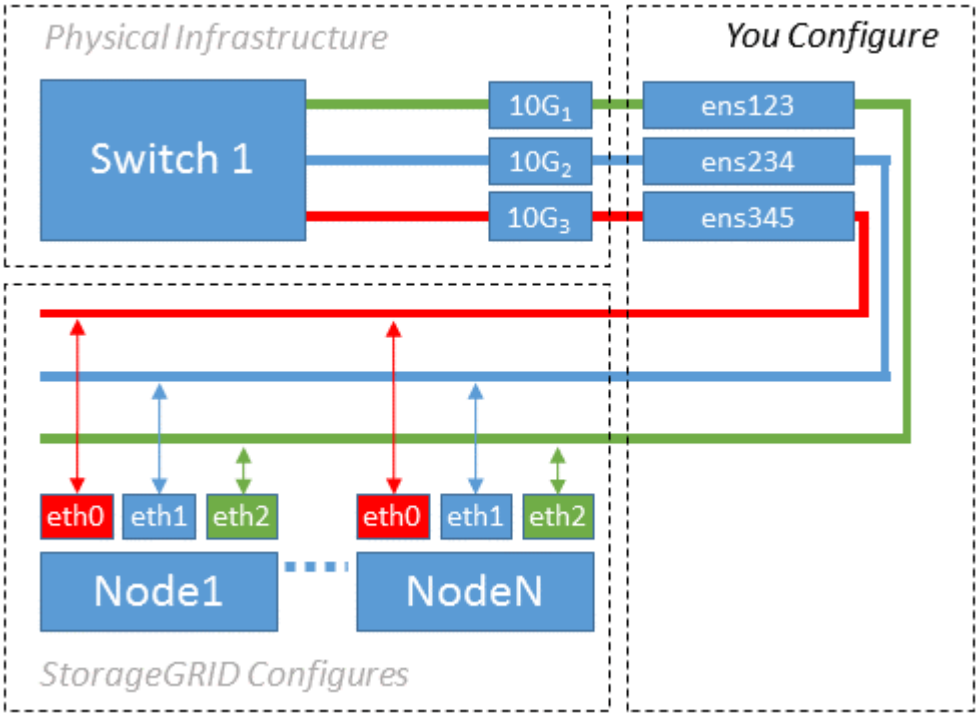
MAC 주소가 11:22:33:44:55:66인 호스트에서 인터페이스 ens256에 대해 MAC 클로닝이 활성화된 예이며, 노드 구성 파일에 다음 키가 있습니다.

- `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 NIC에 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: VLAN**을 전송하는 **LACP** 본딩

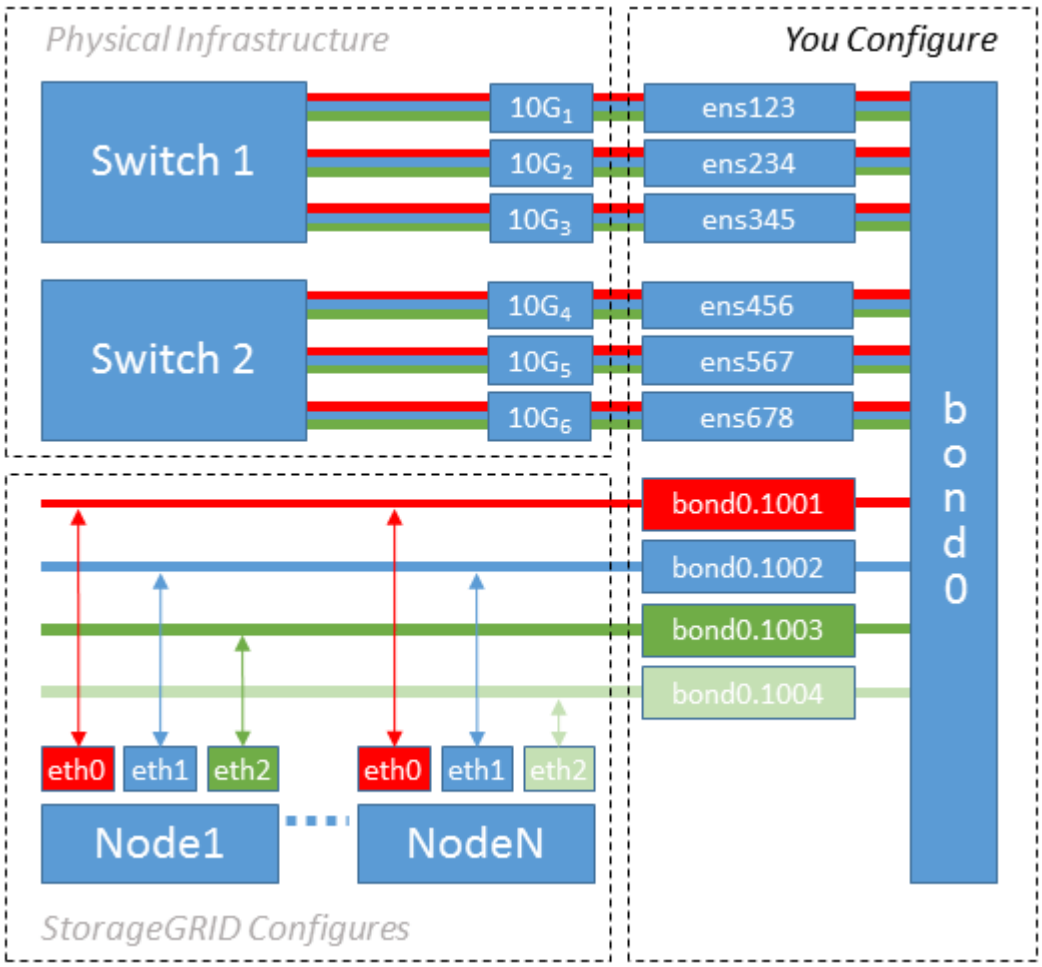
예제 2는 사용자가 네트워크 인터페이스 본딩과 사용 중인 Linux 배포판에서 VLAN 인터페이스를 생성하는 방법에 익숙하다고 가정합니다.

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.

저작권 정보

Copyright © 2026 NetApp, Inc. All Rights Reserved. 미국에서 인쇄됨 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 라이센스와 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 라이센스도 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 라이센스를 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 라이센스는 DFARS 조항 252.227-7015(b)(2014년 2월)에 명시된 권한으로 제한됩니다.

상표 정보

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