



설치를 자동화합니다

StorageGRID software

NetApp
January 15, 2026

목차

설치를 자동화합니다	1
설치 자동화(Linux)	1
StorageGRID 호스트 서비스의 설치 및 구성을 자동화합니다	1
StorageGRID의 구성을 자동화합니다	2
설치 자동화(VMware)	4
그리드 노드 구축을 자동화합니다	4
Bash 스크립트를 실행합니다	15
StorageGRID의 구성을 자동화합니다	16

설치를 자동화합니다

설치 자동화(Linux)

StorageGRID 호스트 서비스 설치 및 그리드 노드 구성을 자동화할 수 있습니다.



이 작업에 대해

"Linux"는 RHEL, Ubuntu 또는 Debian 배포를 의미합니다. 지원되는 버전 목록은 다음을 참조하세요. ["NetApp 상호 운용성 매트릭스 툴\(IMT\)"](#).

구축 자동화는 다음 경우에 유용할 수 있습니다.

- 이미 Ansible, Puppet 또는 Chef와 같은 표준 오케스트레이션 프레임워크를 사용하여 물리적 호스트 또는 가상 호스트를 구축 및 구성합니다.
- 여러 StorageGRID 인스턴스를 배포하려고 합니다.
- 크고 복잡한 StorageGRID 인스턴스를 구축하고 있습니다.

StorageGRID 호스트 서비스는 패키지에 의해 설치되며 구성 파일에 의해 구동됩니다. 다음 방법 중 하나를 사용하여 구성 파일을 만들 수 있습니다.

- ["구성 파일을 만듭니다"](#) 수동 설치 중에 대화형으로 작동합니다.
- 이 문서에 설명되어 있는 대로 표준 오케스트레이션 프레임워크를 사용하여 자동 설치를 수행할 수 있도록 구성 파일을 사전에(또는 프로그래밍 방식으로) 준비합니다.

StorageGRID는 StorageGRID 어플라이언스 및 전체 StorageGRID 시스템("그리드")의 구성을 자동화하기 위한 선택적 Python 스크립트를 제공합니다. 이러한 스크립트를 직접 사용하거나 직접 개발한 그리드 배포 및 구성 도구의 사용 방법을 알아보기 위해 해당 스크립트를 검사할 수 ["StorageGRID 설치 REST API"](#) 있습니다.

StorageGRID 호스트 서비스의 설치 및 구성을 자동화합니다

Anabilities, Puppet, Chef, Fabric 또는 SaltStack과 같은 표준 오케스트레이션 프레임워크를 사용하여 StorageGRID 호스트 서비스의 설치를 자동화할 수 있습니다.

StorageGRID 호스트 서비스는 DEB(Ubuntu 또는 Debian) 또는 RPM(RHEL)으로 패키징되어 있으며, 미리(또는 프로그래밍 방식으로) 준비하여 자동 설치를 활성화할 수 있는 구성 파일에 의해 구동됩니다. 이미 표준 오케스트레이션 프레임워크를 사용하여 Linux 배포를 설치하고 구성하고 있다면 플레이북이나 레시피에 StorageGRID 추가하는 것은 간단할 것입니다.

호스트 준비 및 가상 그리드 노드 구축을 위한 모든 단계를 자동화할 수 있습니다.

Ansible 역할 및 플레이북 예

예제 Ansible 역할 및 플레이북은 폴더에 설치 아카이브와 함께 `/extras` 제공됩니다. Ansible 플레이북에서는 역할이 호스트를 준비하고 타겟 서버에 StorageGRID를 설치하는 방법을 보여줍니다 `storagegrid`. 필요에 따라 역할 또는 플레이북을 사용자 지정할 수 있습니다.



예제 플레이북에는 StorageGRID 호스트 서비스를 시작하기 전에 네트워크 디바이스를 생성하는 데 필요한 단계가 포함되어 있지 않습니다. 이 단계를 추가한 후 플레이북을 사용하여 작업을 완료합니다.

RHEL

RHEL의 경우 제공된 설치 작업은 다음과 같습니다. storagegrid 역할 예제 사용 `ansible.builtin.dnf` 로컬 RPM 파일이나 원격 Yum 저장소에서 설치를 수행하는 모듈입니다. 모듈을 사용할 수 없거나 지원되지 않는 경우 다음 파일에서 적절한 Ansible 작업을 편집하여 사용해야 할 수 있습니다. `yum` 또는 `ansible.builtin.yum` 기준 치수:

- `roles/storagegrid/tasks/rhel_install_from_repo.yml`
- `roles/storagegrid/tasks/rhel_install_from_local.yml`

Ubuntu 또는 Debian

Ubuntu 또는 Debian의 경우 제공된 설치 작업 storagegrid 역할 예제 사용 `ansible.builtin.apt` 로컬 DEB 파일이나 원격 apt 저장소에서 설치를 수행하는 모듈입니다. 모듈을 사용할 수 없거나 지원되지 않는 경우 다음 파일에서 적절한 Ansible 작업을 편집하여 사용해야 할 수 있습니다. `ansible.builtin.apt` 기준 치수:

- `roles/storagegrid/tasks/deb_install_from_repo.yml`
- `roles/storagegrid/tasks/deb_install_from_local.yml`

StorageGRID의 구성을 자동화합니다

그리드 노드를 구축한 후 StorageGRID 시스템 구성을 자동화할 수 있습니다.

시작하기 전에

- 설치 아카이브에서 다음 파일의 위치를 알고 있습니다.

파일 이름	설명
<code>configure-storagegrid.py</code>	구성을 자동화하는 데 사용되는 Python 스크립트입니다
<code>configure -StorageGrid.sample.json</code>	스크립트와 함께 사용할 예제 구성 파일
<code>configure -StorageGrid.blank.json</code> 을 지정합니다	스크립트에 사용할 빈 구성 파일입니다

- `configure-storagegrid.json` 구성 파일을 만들었습니다. 이 파일을 작성하려면 예제 구성 파일 (`configure-storagegrid.sample.json)이나 빈 구성 파일을 수정할 수 (`configure-storagegrid.blank.json` 있습니다.`



수정된 암호 섹션에서 관리 암호와 프로비저닝 암호를 저장합니다. `configure-storagegrid.json` 구성 파일을 안전한 위치에 보관하세요. 이러한 비밀번호는 설치, 확장 및 유지 관리 절차에 필요합니다. 수정된 내용도 백업해야 합니다. `configure-storagegrid.json` 구성 파일을 만들어 안전한 위치에 저장하세요.

이 작업에 대해

Python 스크립트와 `configure-storagegrid.json` 구성 파일을 사용하여 StorageGRID 시스템 구성을 자동화할 수 `configure-storagegrid.py` 있습니다.



그리드 관리자 또는 설치 API를 사용하여 시스템을 구성할 수도 있습니다.

단계

1. Python 스크립트를 실행하기 위해 사용 중인 Linux 시스템에 로그인합니다.
2. 설치 아카이브를 추출한 디렉토리로 변경합니다.

예를 들면 다음과 같습니다.

```
cd StorageGRID-Webscale-version/platform
```

여기서 `platform` 는 `debs`, `rpms` 또는 ``vsphere``입니다.

3. Python 스크립트를 실행하고 생성한 구성 파일을 사용합니다.

예를 들면 다음과 같습니다.

```
./configure-storagegrid.py ./configure-storagegrid.json --start-install
```

결과

복구 패키지 `.zip` 파일은 구성 프로세스 중에 생성되며 설치 및 구성 프로세스를 실행 중인 디렉터리에 다운로드됩니다. 하나 이상의 그리드 노드에 장애가 발생할 경우 StorageGRID 시스템을 복구할 수 있도록 복구 패키지 파일을 백업해야 합니다. 예를 들어, 안전한 백업 네트워크 위치 및 안전한 클라우드 저장소 위치에 복사합니다.



복구 패키지 파일은 StorageGRID 시스템에서 데이터를 가져오는 데 사용할 수 있는 암호화 키와 암호가 포함되어 있으므로 보안을 유지해야 합니다.

임의의 암호를 생성하도록 지정한 경우 파일을 열고 `Passwords.txt` StorageGRID 시스템에 액세스하는 데 필요한 암호를 찾습니다.

```
#####  
##### The StorageGRID "Recovery Package" has been downloaded as: #####  
#####      ./sgws-recovery-package-994078-rev1.zip      #####  
#####   Safeguard this file as it will be needed in case of a   #####  
#####           StorageGRID node recovery.           #####  
#####
```

확인 메시지가 표시되면 StorageGRID 시스템이 설치 및 구성됩니다.

```
StorageGRID has been configured and installed.
```

설치 자동화(VMware)

VMware OVF Tool을 사용하여 그리드 노드의 구축을 자동화할 수 있습니다. StorageGRID의 구성을 자동화할 수도 있습니다.

그리드 노드 구축을 자동화합니다

VMware OVF Tool을 사용하여 그리드 노드의 구축을 자동화합니다.

시작하기 전에

- Bash 3.2 이상이 설치된 Linux/Unix 시스템에 액세스할 수 있습니다.
- vCenter가 포함된 VMware vSphere를 사용하고 있습니다
- VMware OVF Tool을 설치하고 올바르게 구성했습니다.
- OVF 툴을 사용하여 VMware vSphere에 액세스하기 위한 사용자 이름과 암호를 알고 있습니다
- OVF 파일에서 VM을 배포하고 전원을 켤 수 있는 충분한 권한과 VM에 연결할 추가 볼륨을 생성할 수 있는 권한이 있습니다. `ovftool` 자세한 내용은 설명서를 참조하십시오.
- StorageGRID 가상 머신을 구축할 vSphere의 위치에 대한 VI(가상 인프라) URL을 알고 있습니다. 이 URL은 일반적으로 vApp 또는 리소스 풀이 됩니다. 예를 들면 다음과 같습니다.
`vi://vcenter.example.com/vi/sgws`



VMware 유틸리티를 사용하여 이 값을 확인할 수 `ovftool` 있습니다(자세한 내용은 설명서 참조 `ovftool`).



vApp에 구축할 경우 가상 머신이 처음 시작될 때 자동으로 시작되지 않으며 수동으로 전원을 설정해야 합니다.

- 배포 구성 파일에 필요한 모든 정보를 수집했습니다. 자세한 내용은 ["배포 환경에 대한 정보를 수집합니다"](#) 참조하십시오.
- StorageGRID용 VMware 설치 아카이브에서 다음 파일에 액세스할 수 있습니다.

파일 이름	설명
NetApp-SG-version-SHA.vmdk입니다	그리드 노드 가상 머신을 생성하기 위한 템플릿으로 사용되는 가상 머신 디스크 파일입니다. <ul style="list-style-type: none">• 참고: * 이 파일은 및 <code>.mf</code> 파일과 동일한 폴더에 있어야 <code>.ovf</code> 합니다.
vsphere-primary-admin.ovf vsphere-primary-admin.mf	Open Virtualization Format 템플릿 파일(<code>.ovf</code>) 및 매니페스트 파일(<code>.mf</code>)을 사용하여 기본 관리자 노드를 배포할 수 있습니다.
vsphere - non-primary-admin.ovf vsphere - non-primary-admin.mf	템플릿 파일(<code>.ovf</code>) 및 매니페스트 파일(<code>.mf</code>)을 사용하여 비기본 관리 노드를 배포합니다.

파일 이름	설명
vsphere-gateway.ovf vsphere-gateway.mf	템플릿 파일(.ovf) 및 매니페스트 파일(.mf)을 사용하여 게이트웨이 노드를 배포할 수 있습니다.
vsphere-storage.ovf vsphere-storage.mf	템플릿 파일(.ovf) 및 매니페스트 파일(.mf)을 사용하여 가상 머신 기반 스토리지 노드를 구축합니다.
deploy-vsphere-ovftool.sh	가상 그리드 노드의 배포를 자동화하는 데 사용되는 Bash 셸 스크립트입니다.
deploy-vsphere-ovftool-sample.ini	스크립트와 함께 사용할 예제 구성 deploy-vsphere-ovftool.sh 파일입니다.

배포를 위한 구성 파일을 정의합니다

Bash 스크립트에서 사용하는 구성 파일에 StorageGRID용 가상 그리드 노드를 배포하는 데 필요한 정보를 지정합니다 deploy-vsphere-ovftool.sh. 예제 구성 파일을 수정하여 처음부터 새로 만들 필요가 없도록 할 수 있습니다.

단계

1. 예제 구성 파일의 복사본을 (deploy-vsphere-ovftool.sample.ini`만듭니다. 새 파일을 `deploy-vsphere-ovftool.ini`와 같은 디렉토리에 deploy-vsphere-ovftool.sh 저장합니다.
2. `deploy-vsphere-ovftool.ini`개방형:
3. VMware 가상 그리드 노드를 구축하는 데 필요한 모든 정보를 입력합니다.

자세한 내용은 [구성 파일 설정](#)참조하십시오.

4. 필요한 모든 정보를 입력하고 확인했으면 파일을 저장하고 닫습니다.

구성 파일 설정

`deploy-vsphere-ovftool.ini`구성 파일에는 가상 그리드 노드를 배포하는 데 필요한 설정이 포함되어 있습니다.

구성 파일은 먼저 글로벌 매개 변수를 나열한 다음 노드 이름으로 정의된 섹션에 노드별 매개 변수를 나열합니다. 파일이 사용되는 경우:

- `_Global parameters_`는 모든 그리드 노드에 적용됩니다.
- `_노드별 parameters_override` 전역 매개 변수입니다.

전역 매개 변수

전역 매개 변수는 개별 섹션의 설정에 의해 재정의되지 않는 한 모든 그리드 노드에 적용됩니다. 전역 매개 변수 섹션에서 여러 노드에 적용되는 매개 변수를 배치한 다음 필요에 따라 개별 노드의 섹션에서 이러한 설정을 재정의합니다.

- * **OVFTOOL_Arguments** *: OVFTOOL_Arguments을 전역 설정으로 지정하거나 특정 노드에 인수를 개별적으로 적용할 수 있습니다. 예를 들면 다음과 같습니다.

```
OVFTOOL_ARGUMENTS = --powerOn --noSSLVerify --diskMode=eagerZeroedThick
--datastore='datastore_name'
```

및 --overwrite 옵션을 사용하여 기존 가상 머신을 종료하고 교체할 수 --powerOffTarget 있습니다.



노드를 서로 다른 데이터 저장소에 배포하고 전역이 아닌 각 노드에 대해 OVFTOOL_MOUMENT를 지정해야 합니다.

- * **소스** *: StorageGRID 가상 머신 템플릿의 경로(.vmdk) 파일 및 .mf 개별 그리드 노드의 파일과 .ovf 파일. 이 기본값은 현재 디렉터리입니다.

```
SOURCE = /downloads/StorageGRID-Webscale-version/vsphere
```

- * **타겟** *: StorageGRID를 구축할 위치의 VMware vSphere 가상 인프라(vi) URL입니다. 예를 들면 다음과 같습니다.

```
TARGET = vi://vcenter.example.com/vm/sgws
```

- * **GRID_NETWORK_CONFIG** *: 고정 또는 DHCP 중 IP 주소를 획득하는 데 사용되는 방법입니다. 기본값은 정적입니다. 모든 노드 또는 대부분의 노드가 동일한 방법으로 IP 주소를 획득하는 경우 여기에서 해당 방법을 지정할 수 있습니다. 그런 다음 하나 이상의 개별 노드에 대해 다른 설정을 지정하여 전역 설정을 재정의할 수 있습니다. 예를 들면 다음과 같습니다.

```
GRID_NETWORK_CONFIG = STATIC
```

- * **GRID_NETWORK_TARGET** *: 그리드 네트워크에 사용할 기존 VMware 네트워크의 이름입니다. 모든 노드 또는 대부분의 노드가 동일한 네트워크 이름을 사용하는 경우 여기에서 지정할 수 있습니다. 그런 다음 하나 이상의 개별 노드에 대해 다른 설정을 지정하여 전역 설정을 재정의할 수 있습니다. 예를 들면 다음과 같습니다.

```
GRID_NETWORK_TARGET = SG Admin Network
```

- * **grid_network_mask** *: 그리드 네트워크의 네트워크 마스크. 모든 노드 또는 대부분의 노드가 동일한 네트워크 마스크를 사용하는 경우 여기에서 지정할 수 있습니다. 그런 다음 하나 이상의 개별 노드에 대해 다른 설정을 지정하여 전역 설정을 재정의할 수 있습니다. 예를 들면 다음과 같습니다.

```
GRID_NETWORK_MASK = 255.255.255.0
```

- * **grid_network_gateway** *: 그리드 네트워크의 네트워크 게이트웨이. 모든 노드 또는 대부분의 노드가 동일한 네트워크 게이트웨이를 사용하는 경우 여기에서 지정할 수 있습니다. 그런 다음 하나 이상의 개별 노드에 대해 다른

설정을 지정하여 전역 설정을 재정의할 수 있습니다. 예를 들면 다음과 같습니다.

```
GRID_NETWORK_GATEWAY = 10.1.0.1
```

- *GRID_NETWORK_MTU*: 선택 사항. Grid Network의 MTU(Maximum Transmission Unit)입니다. 지정된 경우 값은 1280에서 9216 사이여야 합니다. 예를 들면 다음과 같습니다.

```
GRID_NETWORK_MTU = 9000
```

생략하면 1400이 사용됩니다.

점보 프레임을 사용하려면 MTU를 9000과 같은 점보 프레임에 적합한 값으로 설정합니다. 그렇지 않으면 기본값을 유지합니다.



네트워크의 MTU 값은 노드가 연결된 vSphere의 가상 스위치 포트에 구성된 값과 일치해야 합니다. 그렇지 않으면 네트워크 성능 문제 또는 패킷 손실이 발생할 수 있습니다.



최상의 네트워크 성능을 얻으려면 모든 노드를 그리드 네트워크 인터페이스에서 유사한 MTU 값으로 구성해야 합니다. 개별 노드의 그리드 네트워크에 대한 MTU 설정에 상당한 차이가 있을 경우 * Grid Network MTU mismatch * 경고가 트리거됩니다. MTU 값은 모든 네트워크 유형에 대해 같을 필요는 없습니다.

- * admin_network_Config *: IP 주소를 획득하는 데 사용되는 방법으로, 비활성화, 정적 또는 DHCP입니다. 기본값은 사용 안 함으로 설정되어 있습니다. 모든 노드 또는 대부분의 노드가 동일한 방법으로 IP 주소를 획득하는 경우 여기에서 해당 방법을 지정할 수 있습니다. 그런 다음 하나 이상의 개별 노드에 대해 다른 설정을 지정하여 전역 설정을 재정의할 수 있습니다. 예를 들면 다음과 같습니다.

```
ADMIN_NETWORK_CONFIG = STATIC
```

- * admin_network_target *: 관리 네트워크에 사용할 기존 VMware 네트워크의 이름입니다. 이 설정은 관리 네트워크를 사용하지 않는 한 필요합니다. 모든 노드 또는 대부분의 노드가 동일한 네트워크 이름을 사용하는 경우 여기에서 지정할 수 있습니다. 그리드 네트워크와 달리 모든 노드는 동일한 관리 네트워크에 연결할 필요가 없습니다. 그런 다음 하나 이상의 개별 노드에 대해 다른 설정을 지정하여 전역 설정을 재정의할 수 있습니다. 예를 들면 다음과 같습니다.

```
ADMIN_NETWORK_TARGET = SG Admin Network
```

- * admin_network_mask *: 관리 네트워크의 네트워크 마스크입니다. 이 설정은 고정 IP 주소 지정을 사용하는 경우에 필요합니다. 모든 노드 또는 대부분의 노드가 동일한 네트워크 마스크를 사용하는 경우 여기에서 지정할 수 있습니다. 그런 다음 하나 이상의 개별 노드에 대해 다른 설정을 지정하여 전역 설정을 재정의할 수 있습니다. 예를 들면 다음과 같습니다.

```
ADMIN_NETWORK_MASK = 255.255.255.0
```

- * admin_network_gateway *: 관리 네트워크의 네트워크 게이트웨이입니다. 이 설정은 정적 IP 주소 지정을 사용하고 admin_network_ESL 설정에서 외부 서브넷을 지정하는 경우에 필요합니다. (즉, admin_network_esl이 비어 있으면 필요하지 않습니다.) 모든 노드 또는 대부분의 노드가 동일한 네트워크 게이트웨이를 사용하는 경우 여기에서 지정할 수 있습니다. 그런 다음 하나 이상의 개별 노드에 대해 다른 설정을 지정하여 전역 설정을 재정의할 수 있습니다. 예를 들면 다음과 같습니다.

```
ADMIN_NETWORK_GATEWAY = 10.3.0.1
```

- * admin_network_ESL *: 심표로 구분된 CIDR 라우트 대상 목록으로 지정된 관리 네트워크의 외부 서브넷 목록 (라우트). 모든 노드 또는 대부분의 노드가 동일한 외부 서브넷 목록을 사용하는 경우 여기에서 지정할 수 있습니다. 그런 다음 하나 이상의 개별 노드에 대해 다른 설정을 지정하여 전역 설정을 재정의할 수 있습니다. 예를 들면 다음과 같습니다.

```
ADMIN_NETWORK_ESL = 172.16.0.0/21,172.17.0.0/21
```

- * admin_network_mtu *: 선택 사항. 관리자 네트워크의 MTU(Maximum Transmission Unit) admin_network_Config=DHCP인지 지정하지 마십시오. 지정된 경우 값은 1280에서 9216 사이여야 합니다. 생각하면 1400이 사용됩니다. 점보 프레임을 사용하려면 MTU를 9000과 같은 점보 프레임에 적합한 값으로 설정합니다. 그렇지 않으면 기본값을 유지합니다. 모든 노드 또는 대부분의 노드가 Admin Network에 대해 동일한 MTU를 사용하는 경우 여기에서 지정할 수 있습니다. 그런 다음 하나 이상의 개별 노드에 대해 다른 설정을 지정하여 전역 설정을 재정의할 수 있습니다. 예를 들면 다음과 같습니다.

```
ADMIN_NETWORK_MTU = 8192
```

- * CLIENT_NETWORK_CONFIG *: IP 주소를 획득하는 데 사용되는 방법으로, 비활성화, 정적 또는 DHCP 입니다. 기본값은 사용 안 함으로 설정되어 있습니다. 모든 노드 또는 대부분의 노드가 동일한 방법으로 IP 주소를 획득하는 경우 여기에서 해당 방법을 지정할 수 있습니다. 그런 다음 하나 이상의 개별 노드에 대해 다른 설정을 지정하여 전역 설정을 재정의할 수 있습니다. 예를 들면 다음과 같습니다.

```
CLIENT_NETWORK_CONFIG = STATIC
```

- * client_network_target *: 클라이언트 네트워크에 사용할 기존 VMware 네트워크의 이름입니다. 이 설정은 클라이언트 네트워크를 사용하지 않는 경우에만 필요합니다. 모든 노드 또는 대부분의 노드가 동일한 네트워크 이름을 사용하는 경우 여기에서 지정할 수 있습니다. 그리드 네트워크와 달리 모든 노드는 동일한 클라이언트 네트워크에 연결할 필요가 없습니다. 그런 다음 하나 이상의 개별 노드에 대해 다른 설정을 지정하여 전역 설정을 재정의할 수 있습니다. 예를 들면 다음과 같습니다.

```
CLIENT_NETWORK_TARGET = SG Client Network
```

- * client_network_mask *: 클라이언트 네트워크의 네트워크 마스크입니다. 이 설정은 고정 IP 주소 지정을 사용하는 경우에 필요합니다. 모든 노드 또는 대부분의 노드가 동일한 네트워크 마스크를 사용하는 경우 여기에서 지정할 수 있습니다. 그런 다음 하나 이상의 개별 노드에 대해 다른 설정을 지정하여 전역 설정을 재정의할 수 있습니다. 예를 들면 다음과 같습니다.

```
CLIENT_NETWORK_MASK = 255.255.255.0
```

- * client_network_gateway *: 클라이언트 네트워크의 네트워크 게이트웨이입니다. 이 설정은 고정 IP 주소 지정을 사용하는 경우에 필요합니다. 모든 노드 또는 대부분의 노드가 동일한 네트워크 게이트웨이를 사용하는 경우 여기에서 지정할 수 있습니다. 그런 다음 하나 이상의 개별 노드에 대해 다른 설정을 지정하여 전역 설정을 재정의할 수 있습니다. 예를 들면 다음과 같습니다.

```
CLIENT_NETWORK_GATEWAY = 10.4.0.1
```

- * client_network_mtu *: 선택 사항. 클라이언트 네트워크의 MTU(Maximum Transmission Unit) client_network_Config = DHCP인지 지정하지 마십시오. 지정된 경우 값은 1280에서 9216 사이여야 합니다. 생략하면 1400이 사용됩니다. 정보 프레임을 사용하려면 MTU를 9000과 같은 정보 프레임에 적합한 값으로 설정합니다. 그렇지 않으면 기본값을 유지합니다. 모든 노드 또는 대부분의 노드가 클라이언트 네트워크에 동일한 MTU를 사용하는 경우 여기에서 지정할 수 있습니다. 그런 다음 하나 이상의 개별 노드에 대해 다른 설정을 지정하여 전역 설정을 재정의할 수 있습니다. 예를 들면 다음과 같습니다.

```
CLIENT_NETWORK_MTU = 8192
```

- * port_remap *: 내부 그리드 노드 통신 또는 외부 통신을 위해 노드에서 사용하는 포트를 다시 매핑합니다. 엔터프라이즈 네트워킹 정책이 StorageGRID에서 사용하는 하나 이상의 포트를 제한하는 경우 포트를 다시 매핑해야 합니다. StorageGRID에서 사용하는 포트 목록은 의 내부 그리드 노드 통신 및 외부 통신 을 참조하십시오"[네트워킹 지침](#)".



로드 밸런서 엔드포인트를 구성하는 데 사용할 포트를 다시 매핑하지 마십시오.



port_remap 만 설정된 경우 지정하는 매핑이 인바운드 및 아웃바운드 통신 모두에 사용됩니다. port_remap_inbound 도 지정된 경우 port_remap 은 아웃바운드 통신에만 적용됩니다.

사용되는 형식은 다음과 같습니다 *network type/protocol/default port used by grid node/new port*. 여기서 네트워크 유형은 그리드, 관리자 또는 클라이언트이고 프로토콜은 TCP 또는 UDP입니다.

예를 들면 다음과 같습니다.

```
PORT_REMAP = client/tcp/18082/443
```

단독으로 사용하는 경우 이 예제 설정은 그리드 노드에 대한 인바운드 및 아웃바운드 통신을 포트 18082에서 포트 443으로 대칭적으로 매핑합니다. port_remap_inbound 와 함께 사용할 경우 이 예제 설정은 포트 18082에서 포트 443으로 아웃바운드 통신을 매핑합니다.

심표로 구분된 목록을 사용하여 여러 포트를 다시 매핑할 수도 있습니다.

예를 들면 다음과 같습니다.

```
PORT_REMAP = client/tcp/18082/443, client/tcp/18083/80
```

- *port_remap_inbound*: 지정된 포트에 대한 인바운드 통신을 다시 매핑합니다. port_remap_inbound 를 지정하지만 port_remap 의 값을 지정하지 않으면 포트의 아웃바운드 통신이 변경되지 않습니다.



로드 밸런서 엔드포인트를 구성하는 데 사용할 포트를 다시 매핑하지 마십시오.

사용되는 형식은 다음과 같습니다 *network type/protocol/_default port used by grid node /new port*. 여기서 네트워크 유형은 그리드, 관리자 또는 클라이언트이고 프로토콜은 TCP 또는 UDP입니다.

예를 들면 다음과 같습니다.

```
PORT_REMAP_INBOUND = client/tcp/443/18082
```

이 예에서는 포트 443으로 전송된 트래픽을 내부 방화벽을 통과하여 그리드 노드가 S3 요청을 수신하는 포트 18082로 전달합니다.

심표로 구분된 목록을 사용하여 여러 인바운드 포트를 다시 매핑할 수도 있습니다.

예를 들면 다음과 같습니다.

```
PORT_REMAP_INBOUND = grid/tcp/3022/22, admin/tcp/3022/22
```

- *Temporary_password_type*: 노드가 그리드에 합류하기 전에 VM 콘솔이나 StorageGRID 설치 API에 액세스하거나 SSH를 사용할 때 사용되는 임시 설치 암호의 유형입니다.



모든 노드 또는 대부분의 노드가 동일한 유형의 임시 설치 암호를 사용하는 경우 전역 매개 변수 섹션에 형식을 지정합니다. 그런 다음 필요에 따라 개별 노드에 대해 다른 설정을 사용합니다. 예를 들어, *사용자 정의 암호 사용* 을 전역적으로 선택한 경우 * custom_temporary_password=<password> * 를 사용하여 각 노드의 암호를 설정할 수 있습니다.

- Temporary_password_type * 은 다음 중 하나일 수 있습니다.

- *노드 이름 사용*: 노드 이름은 임시 설치 암호로 사용되며 VM 콘솔, StorageGRID 설치 API 및 SSH에 대한 액세스를 제공합니다.
- *비밀번호 비활성화*: 임시 설치 비밀번호가 사용되지 않습니다. 설치 문제를 디버깅하기 위해 VM에 액세스해야 하는 경우 를 참조하십시오"[설치 문제를 해결합니다](#)".
- *사용자 정의 암호 사용*: * custom_temporary_password=<password> * 에 제공된 값은 임시 설치 암호로 사용되며 VM 콘솔, StorageGRID 설치 API 및 SSH에 대한 액세스를 제공합니다.



필요한 경우 * Temporary_password_type * 매개 변수를 생략하고 * custom_Temporary_password=<password> * 만 지정할 수 있습니다.

- *custom_Temporary_password=<password> * 선택 요소입니다. 설치 중에 VM 콘솔, StorageGRID 설치 API 및 SSH에 액세스할 때 사용할 임시 암호입니다. Temporary_password_type * 이 *노드 이름 사용* 또는 * 암호

비활성화 * 로 설정된 경우 무시됩니다.

노드별 매개 변수

각 노드는 구성 파일의 자체 섹션에 있습니다. 각 노드에는 다음과 같은 설정이 필요합니다.

- 섹션 헤드는 그리드 관리자에 표시될 노드 이름을 정의합니다. 노드에 대해 선택 사항인 `node_name` 매개 변수를 지정하여 이 값을 재정의할 수 있습니다.
- * `node_type` *: `VM_Admin_Node`, `VM_Storage_Node` 또는 `VM_API_Gateway_Node`
- * `STORAGE_TYPE` *: 조합, 데이터 또는 메타데이터. 스토리지 노드의 이 선택적 매개 변수는 지정되지 않은 경우 기본적으로 `Combined`(데이터 및 메타데이터)로 설정됩니다. 자세한 내용은 을 "[스토리지 노드 유형](#)"참조하십시오.
- * `GRID_NETWORK_IP` *: 그리드 네트워크의 노드에 대한 IP 주소입니다.
- * `admin_network_ip` *: 관리 네트워크의 노드에 대한 IP 주소입니다. 노드가 Admin Network에 연결되어 있고 `admin_network_Config`가 `static`으로 설정된 경우에만 필요합니다.
- * `client_network_ip` *: 클라이언트 네트워크의 노드에 대한 IP 주소입니다. 노드가 클라이언트 네트워크에 연결되어 있고 이 노드의 `client_network_Config`가 `static`으로 설정된 경우에만 필요합니다.
- * `admin_IP` *: 그리드 네트워크의 기본 관리 노드에 대한 IP 주소입니다. 기본 관리 노드에 대해 `GRID_NETWORK_IP`로 지정하는 값을 사용합니다. 이 매개 변수를 생략하면 노드가 mDNS를 사용하여 운영 관리 노드 IP를 검색합니다. 자세한 내용은 을 "[그리드 노드가 기본 관리자 노드를 검색하는 방법](#)"참조하십시오.



`admin_ip` 매개 변수는 기본 관리 노드에 대해 무시됩니다.

- 전역적으로 설정되지 않은 모든 매개변수. 예를 들어, 노드가 관리 네트워크에 연결되어 있고 `admin_network` 매개 변수를 전역으로 지정하지 않은 경우 노드에 대해 이러한 매개 변수를 지정해야 합니다.

기본 관리자 노드

기본 관리 노드에는 다음과 같은 추가 설정이 필요합니다.

- * `node_type` *: `vm_Admin_Node`
- * `admin_role` *: 기본

이 예제 항목은 세 네트워크 모두에 있는 기본 관리 노드에 대한 것입니다.

```
[DC1-ADM1]
ADMIN_ROLE = Primary
NODE_TYPE = VM_Admin_Node
TEMPORARY_PASSWORD_TYPE = Use custom password
CUSTOM_TEMPORARY_PASSWORD = Passw0rd

GRID_NETWORK_IP = 10.1.0.2
ADMIN_NETWORK_IP = 10.3.0.2
CLIENT_NETWORK_IP = 10.4.0.2
```

기본 관리 노드에는 다음과 같은 추가 설정이 선택 사항입니다.

- * `디스크` *: 기본적으로 감사 및 데이터베이스 사용을 위해 관리자 노드에 두 개의 추가 200GB 하드 디스크가

할당됩니다. disk 매개 변수를 사용하여 이러한 설정을 늘릴 수 있습니다. 예를 들면 다음과 같습니다.

```
DISK = INSTANCES=2, CAPACITY=300
```



관리 노드의 경우 인스턴스는 항상 2가 되어야 합니다.

스토리지 노드

스토리지 노드에는 다음과 같은 추가 설정이 필요합니다.

- * node_type *: vm_storage_Node

이 예제 항목은 그리드 및 관리 네트워크에 있지만 클라이언트 네트워크에 없는 스토리지 노드에 대한 것입니다. 이 노드는 admin_ip 설정을 사용하여 그리드 네트워크에서 기본 관리 노드의 IP 주소를 지정합니다.

```
[DC1-S1]
NODE_TYPE = VM_Storage_Node

GRID_NETWORK_IP = 10.1.0.3
ADMIN_NETWORK_IP = 10.3.0.3

ADMIN_IP = 10.1.0.2
```

이 두 번째 예제 항목은 고객의 엔터프라이즈 네트워킹 정책에서 포트 80 또는 443을 사용하여 스토리지 노드에 액세스할 수 있다고 명시하는 클라이언트 네트워크의 스토리지 노드에 대한 것입니다. 예제 구성 파일은 port_remap을 사용하여 스토리지 노드가 포트 443에서 S3 메시지를 보내고 받을 수 있도록 합니다.

```
[DC2-S1]
NODE_TYPE = VM_Storage_Node

GRID_NETWORK_IP = 10.1.1.3
CLIENT_NETWORK_IP = 10.4.1.3
PORT_REMAP = client/tcp/18082/443

ADMIN_IP = 10.1.0.2
```

마지막 예에서는 포트 22에서 포트 3022로 ssh 트래픽에 대한 대칭 재매핑을 생성하지만 인바운드 및 아웃바운드 트래픽에 대한 값을 명시적으로 설정합니다.

```
[DC1-S3]
NODE_TYPE = VM_Storage_Node

GRID_NETWORK_IP = 10.1.1.3

PORT_REMAP = grid/tcp/22/3022
PORT_REMAP_INBOUND = grid/tcp/3022/22

ADMIN_IP = 10.1.0.2
```

스토리지 노드에 대한 다음 추가 설정은 선택 사항입니다.

- * disk *: 기본적으로 스토리지 노드에는 RangeDB 사용을 위해 3 개의 4TB 디스크가 할당됩니다. 디스크 매개 변수를 사용하여 이러한 설정을 늘릴 수 있습니다. 예를 들면 다음과 같습니다.

```
DISK = INSTANCES=16, CAPACITY=4096
```

- * STORAGE_TYPE *: 기본적으로 모든 새 스토리지 노드는 오브젝트 데이터와 메타데이터를 모두 저장하도록 구성되어 있습니다. storage_type 매개 변수를 사용하여 데이터나 메타데이터만 저장하도록 스토리지 노드 유형을 변경할 수 있습니다. 예를 들면 다음과 같습니다.

```
STORAGE_TYPE = data
```

게이트웨이 노드

게이트웨이 노드에는 다음과 같은 추가 설정이 필요합니다.

- * node_type *: vm_api_Gateway

이 예제 항목은 세 네트워크 모듈에서 게이트웨이 노드의 예입니다. 이 예에서는 구성 파일의 전역 섹션에 클라이언트 네트워크 매개 변수가 지정되지 않아 노드에 대해 지정해야 합니다.

```
[DC1-G1]
NODE_TYPE = VM_API_Gateway

GRID_NETWORK_IP = 10.1.0.5
ADMIN_NETWORK_IP = 10.3.0.5

CLIENT_NETWORK_CONFIG = STATIC
CLIENT_NETWORK_TARGET = SG Client Network
CLIENT_NETWORK_MASK = 255.255.255.0
CLIENT_NETWORK_GATEWAY = 10.4.0.1
CLIENT_NETWORK_IP = 10.4.0.5

ADMIN_IP = 10.1.0.2
```

운영 관리자 노드가 아닌 노드

운영 관리자 노드가 아닌 경우 다음과 같은 추가 설정이 필요합니다.

- * node_type *: vm_Admin_Node
- * admin_role *: Non-Primary

이 예제 항목은 클라이언트 네트워크에 없는 비 기본 관리 노드에 대한 것입니다.

```
[DC2-ADM1]
ADMIN_ROLE = Non-Primary
NODE_TYPE = VM_Admin_Node

GRID_NETWORK_TARGET = SG Grid Network
GRID_NETWORK_IP = 10.1.0.6
ADMIN_NETWORK_IP = 10.3.0.6

ADMIN_IP = 10.1.0.2
```

다음 추가 설정은 운영 관리자 노드가 아닌 경우 선택 사항입니다.

- * 디스크 *: 기본적으로 감사 및 데이터베이스 사용을 위해 관리자 노드에 두 개의 추가 200GB 하드 디스크가 할당됩니다. disk 매개 변수를 사용하여 이러한 설정을 늘릴 수 있습니다. 예를 들면 다음과 같습니다.

```
DISK = INSTANCES=2, CAPACITY=300
```



관리 노드의 경우 인스턴스는 항상 2가 되어야 합니다.

Bash 스크립트를 실행합니다

VMware vSphere에서 StorageGRID 노드의 배포를 자동화하기 위해 수정한 Bash 스크립트와 deploy-vsphere-ovftool.ini 구성 파일을 사용할 수 deploy-vsphere-ovftool.sh 있습니다.

시작하기 전에

사용자 환경에 대한 deploy-vsphere-ovftool.ini 구성 파일을 만들었습니다.

도움말 명령을 입력하여 Bash 스크립트에서 사용할 수 있는 도움말을 사용할 수 (`-h/--help` 있습니다.) 예를 들면 다음과 같습니다.

```
./deploy-vsphere-ovftool.sh -h
```

또는

```
./deploy-vsphere-ovftool.sh --help
```

단계

1. Bash 스크립트를 실행하기 위해 사용 중인 Linux 시스템에 로그인합니다.
2. 설치 아카이브를 추출한 디렉토리로 변경합니다.

예를 들면 다음과 같습니다.

```
cd StorageGRID-Webscale-version/vsphere
```

3. 모든 그리드 노드를 배포하려면 환경에 적합한 옵션을 사용하여 Bash 스크립트를 실행합니다.

예를 들면 다음과 같습니다.

```
./deploy-vsphere-ovftool.sh --username=user --password=pwd ./deploy-vsphere-ovftool.ini
```

4. 오류로 인해 그리드 노드를 배포하지 못한 경우 오류를 해결하고 해당 노드에 대해서만 Bash 스크립트를 다시 실행합니다.

예를 들면 다음과 같습니다.

```
./deploy-vsphere-ovftool.sh --username=user --password=pwd --single -node="DC1-S3" ./deploy-vsphere-ovftool.ini
```

각 노드의 상태가 "Passed"이면 배포가 완료됩니다.

Deployment Summary

node	attempts	status
DC1-ADM1	1	Passed
DC1-G1	1	Passed
DC1-S1	1	Passed
DC1-S2	1	Passed
DC1-S3	1	Passed

StorageGRID의 구성을 자동화합니다

그리드 노드를 구축한 후 StorageGRID 시스템 구성을 자동화할 수 있습니다.

시작하기 전에

- 설치 아카이브에서 다음 파일의 위치를 알고 있습니다.

파일 이름	설명
configure-storagegrid.py	구성을 자동화하는 데 사용되는 Python 스크립트입니다
configure -StorageGrid.sample.json	스크립트와 함께 사용할 예제 구성 파일
configure -StorageGrid.blank.json을 지정합니다	스크립트에 사용할 빈 구성 파일입니다

- `configure-storagegrid.json`` 구성 파일을 만들었습니다. 이 파일을 작성하려면 예제 구성 파일 (`configure-storagegrid.sample.json`)이나 빈 구성 파일을 수정할 수 (`configure-storagegrid.blank.json``) 있습니다.



수정된 암호 섹션에서 관리 암호와 프로비저닝 암호를 저장합니다. `configure-storagegrid.json` 구성 파일을 안전한 위치에 보관하세요. 이러한 비밀번호는 설치, 확장 및 유지 관리 절차에 필요합니다. 수정된 내용도 백업해야 합니다. `configure-storagegrid.json` 구성 파일을 만들어 안전한 위치에 저장하세요.

이 작업에 대해

Python 스크립트와 `configure-storagegrid.json` 그리드 구성 파일을 사용하여 StorageGRID 시스템의 구성을 자동화할 수 `configure-storagegrid.py` 있습니다.



그리드 관리자 또는 설치 API를 사용하여 시스템을 구성할 수도 있습니다.

단계

1. Python 스크립트를 실행하기 위해 사용 중인 Linux 시스템에 로그인합니다.
2. 설치 아카이브를 추출한 디렉토리로 변경합니다.

예를 들면 다음과 같습니다.

```
cd StorageGRID-Webscale-version/platform
```

여기서 platform 는 debs, rpms 또는 vSphere입니다.

3. Python 스크립트를 실행하고 생성한 구성 파일을 사용합니다.

예를 들면 다음과 같습니다.

```
./configure-storagegrid.py ./configure-storagegrid.json --start-install
```

결과

복구 패키지 .zip 파일은 구성 프로세스 중에 생성되며 설치 및 구성 프로세스를 실행 중인 디렉터리에 다운로드됩니다. 하나 이상의 그리드 노드에 장애가 발생할 경우 StorageGRID 시스템을 복구할 수 있도록 복구 패키지 파일을 백업해야 합니다. 예를 들어, 안전한 백업 네트워크 위치 및 안전한 클라우드 저장소 위치에 복사합니다.



복구 패키지 파일은 StorageGRID 시스템에서 데이터를 가져오는 데 사용할 수 있는 암호화 키와 암호가 포함되어 있으므로 보안을 유지해야 합니다.

임의의 암호를 생성하도록 지정한 경우 파일을 열고 Passwords.txt StorageGRID 시스템에 액세스하는 데 필요한 암호를 찾습니다.

```
#####
##### The StorageGRID "Recovery Package" has been downloaded as: #####
#####      ./sgws-recovery-package-994078-rev1.zip      #####
##### Safeguard this file as it will be needed in case of a #####
#####      StorageGRID node recovery.      #####
#####
```

확인 메시지가 표시되면 StorageGRID 시스템이 설치 및 구성됩니다.

```
StorageGRID has been configured and installed.
```

관련 정보

- ["그리드 관리자로 이동합니다"](#)
- ["REST API 설치"](#)

저작권 정보

Copyright © 2026 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.