



# ソフトウェアベースのノードをインストール StorageGRID software

NetApp  
March 10, 2026

# 目次

ソフトウェアベースのノードをインストール	1
ソフトウェアベースのノードにStorageGRIDをインストールするためのクイックスタート	1
インストールを自動化します	1
ソフトウェアベースのノードへのインストールを計画および準備する	2
StorageGRIDに必要な情報と資料	2
StorageGRID インストールファイルをダウンロードして展開します	4
StorageGRIDインストールファイルを確認する	9
StorageGRIDのソフトウェア要件	10
StorageGRIDのCPUとRAMの要件	14
StorageGRIDのストレージとパフォーマンスの要件	15
Linux の StorageGRID ノードコンテナ移行要件	21
ホストを準備する (Linux)	23
ソフトウェアベースのノードのインストールを自動化する	41
Linux上のStorageGRIDホストサービスのインストールとグリッドノードの構成を自動化	41
StorageGRIDでVMware vSphereを使用してグリッドノードの導入を自動化	44
仮想グリッドノードを展開する	59
StorageGRID用のVMware環境に関する情報を収集	59
Linux デプロイメント用の StorageGRID ノード構成ファイルを作成する	60
グリッドノードがStorageGRIDでプライマリ管理ノードを検出する方法	78
VMware vSphereを使用してStorageGRIDの仮想マシンとしてグリッドノードを導入	79
Linux上のStorageGRIDノードの設定ファイルの例	86
Linux での StorageGRID 設定を検証する	88
LinuxでStorageGRIDホストサービスを開始します	89
StorageGRID インストール REST API について学ぶ	90
StorageGRID インストール API	90
StorageGRIDインストールの問題のトラブルシューティング	91
スクリプトの例	93
RHEL上のStorageGRIDで4つの物理インターフェイスを1つのボンドに統合する例	93
Ubuntu および Debian 上の StorageGRID の物理インターフェイスを集約する例	96

# ソフトウェアベースのノードをインストール

## ソフトウェアベースのノードにStorageGRIDをインストールするためのクイックスタート

Linux または VMware StorageGRIDノードをインストールするには、次の大まかな手順に従います。



「Linux」は、RHEL、Ubuntu、または Debian のデプロイメントを指します。サポートされているバージョンのリストについては、"[NetApp Interoperability Matrix Tool \(IMT\)](#)"。

1

### 準備

- 詳細はこちらをご覧ください "[StorageGRID のアーキテクチャとネットワークトポロジ](#)"。
- の詳細については、を "[StorageGRID ネットワーク](#)"参照してください。
- を集めて準備します"[必要な情報と資料](#)"。
- (VMwareのみ) インストールと構成"[VMware vSphereハイパーバイザー、vCenter、およびESXホスト](#)"。
- 必要なを準備します"[CPUおよびRAM](#)"。
- を提供し"[ストレージとパフォーマンスの要件](#)"ます。
- (Linux のみ)"[Linuxサーバの準備](#)" StorageGRIDノードをホストします。

2

### 導入

グリッドノードを導入する。導入したグリッドノードは、StorageGRID システムの一部として作成され、1 つ以上のネットワークに接続されます。

- (Linuxのみ) 手順1で準備したホストにソフトウェアベースのグリッドノードを展開するには、Linuxコマンドラインを使用して"[ノード構成ファイル](#)"。
- (VMwareのみ) VMware vSphere Web Client、.vmdkファイル、および.ovfファイルテンプレートのセットを使用して、"[ソフトウェアベースのノードを仮想マシン \(VM\) として導入](#)"手順 1 で準備したサーバー上で。
- StorageGRIDアプライアンスノードを導入するには、に従って "[ハードウェア設置のクイックスタート](#)"ください。

3

### 構成

すべてのノードを導入したら、Grid Managerを使用してに"[グリッドを設定し、インストールを完了する](#)"移動します。

インストールを自動化します



「Linux」は、RHEL、Ubuntu、または Debian のデプロイメントを指します。サポートされているバージョンのリストについては、"[NetApp Interoperability Matrix Tool \(IMT\)](#)"。

## Linux

時間を節約し、整合性を確保するために、StorageGRIDホストサービスのインストールとグリッドノードの設定を自動化できます。

- Ansible、Puppet、Chefなどの標準的なオーケストレーションフレームワークを使用して自動化：
  - Linuxのインストール
  - ネットワークとストレージの構成
  - コンテナエンジンとStorageGRIDホストサービスのインストール
  - 仮想グリッドノードの導入

見る"[StorageGRID ホストサービスのインストールと設定を自動化する](#)"。

- グリッドノードを展開した後、"[StorageGRIDシステムの設定を自動化](#)"インストール アーカイブに提供されている Python 構成スクリプトを使用します。
- "[アプライアンスグリッドノードのインストールと設定を自動化する](#)"
- StorageGRID環境の高度な開発者は、を使用してグリッドノードのインストールを自動化します"[インストールREST API](#)"。

## VMware

時間を節約し、整合性を確保するために、グリッドノードの導入と設定、およびStorageGRIDシステムの設定を自動化できます。

- "[VMware vSphereを使用してグリッドノードの導入を自動化](#)"です。
- インストールアーカイブに付属のPython設定スクリプトを使用して、グリッドノードを導入したあとに"[StorageGRIDシステムの設定を自動化](#)"実行します。
- "[アプライアンスグリッドノードのインストールと設定を自動化する](#)"
- StorageGRID環境の高度な開発者は、を使用してグリッドノードのインストールを自動化します"[インストールREST API](#)"。

# ソフトウェアベースのノードへのインストールを計画および準備する

## StorageGRIDに必要な情報と資料

StorageGRIDをインストールする前に、必要な情報や資料を収集して準備します。

必要な情報

## ネットワーク計画

各StorageGRIDノードに接続するネットワーク。StorageGRIDは、トラフィックの分離、セキュリティ、および管理上の利便性のために、複数のネットワークをサポートしています。

StorageGRIDを参照してください"[ネットワークのガイドライン](#)"。

## ネットワーク情報

各グリッドノードに割り当てるIPアドレス、およびDNSサーバとNTPサーバのIPアドレス。

## グリッドノードヨウノサーバ

導入予定の StorageGRID ノードの数とタイプに応じて、それらをサポートできる十分なリソースを備えた一連のサーバ（物理、仮想、またはその両方）を特定します。



StorageGRID 環境でStorageGRID アプライアンス（ハードウェア）ストレージノードを使用しない場合は、バッテリーバックアップ式書き込みキャッシュ（BBWC）を備えたハードウェアRAIDストレージを使用する必要があります。StorageGRID は、Virtual Storage Area Network（VSAN;仮想ストレージエリアネットワーク）、ソフトウェアRAID、またはRAID保護なしの使用をサポートしていません。

## ノードの移行（必要な場合、Ubuntu と Debian のみ）

"[ノード移行の要件](#)"物理ホストでサービスを中断せずに定期的なメンテナンスを実行する場合は、を参照してください。

## 関連情報

"[NetApp Interoperability Matrix Tool](#)"

## 前提要件

### NetApp StorageGRID ライセンス

デジタル署名された有効なNetAppライセンスが必要です。



StorageGRIDのインストールアーカイブには、グリッドのテストとコンセプトの実証に使用できる非本番環境のライセンスが含まれています。

### StorageGRID インストールアーカイブ

"[StorageGRIDインストールアーカイブをダウンロードしてファイルを展開する](#)"です。

## サービスラップトップ

StorageGRID システムは、サービスラップトップを介してインストールされます。

サービスラップトップには次のものがが必要です。

- ネットワークポート
- SSH クライアント（PuTTY など）
- "[サポートされている Web ブラウザ](#)"

## StorageGRID のドキュメント

- ["リリースノート"](#)
- ["StorageGRID の管理手順"](#)

## StorageGRID インストールファイルをダウンロードして展開します

StorageGRID インストールアーカイブをダウンロードし、必要なファイルを展開する必要があります。必要に応じて、インストールパッケージ内のファイルを手動で検証できます。

### 手順

1. に進みます ["ネットアップの StorageGRID ダウンロードページ"](#)。
2. 最新のリリースをダウンロードするボタンを選択するか、ドロップダウンメニューから別のバージョンを選択して、「\* Go \*」を選択します。
3. ネットアップアカウントのユーザ名とパスワードを使用してサインインします。
4. Caution/MustRead文が表示された場合は'その文を読み'チェックボックスをオンにします



StorageGRID リリースのインストール後に、必要な修正プログラムを適用する必要があります。詳細については、["リカバリとメンテナンスの手順の Hotfix 手順"](#)

5. [End User License Agreement]を読み、チェックボックスをオンにして、\*[Accept & Continue]\*を選択します。
6. \* StorageGRIDのインストール \*列で、ソフトウェアベースのノード タイプ (RHEL、Ubuntu、Debian、または VMware) に応じた .tgz または .zip インストール アーカイブを選択します。



サービスラップトップでWindowsを実行している場合は、ファイルを使用し`.zip`ます。

7. インストールアーカイブを保存します。
8. Linux ノードではコード署名の検証は手動で行います。オプションで、インストール アーカイブを検証する必要がある場合:
  - a. StorageGRIDコード署名検証パッケージをダウンロードします。このパッケージのファイル名はの形式を使用し`StorageGRID\_<version-number>\_Code\_Signature\_Verification\_Package.tar.gz`ます。  
`<version-number>`はStorageGRIDソフトウェアのバージョンです。
  - b. 手順に従ってください["インストールファイルを手動で検証する"](#)。
9. インストールアーカイブからファイルを展開します。
10. 必要なファイルを選択します。

必要なファイルは、計画したグリッドトポロジおよびStorageGRIDシステムの導入方法によって異なります。



次の表に示すパスは、展開されたインストールアーカイブによってインストールされた最上位ディレクトリに対する相対パスです。

## RHEL

パスとファイル名	製品説明
	StorageGRID ダウンロードファイルに含まれているすべてのファイルについて説明するテキストファイル。
	製品サポートのない無償ライセンス。
	RHELホストにStorageGRIDノードイメージをインストールするためのRPMパッケージ。
	RHELホストにStorageGRIDホストサービスをインストールするためのRPMパッケージ。
導入スクリプトツール	製品説明
	StorageGRID システムの設定を自動化するためのPython スクリプト。
	StorageGRID アプライアンスの設定を自動化するための Python スクリプト。
	スクリプトで使用する構成ファイルの例 configure-storagegrid.py
	シングルサインオンが有効な場合にグリッド管理API にサインインするために使用できる Python スクリプトの例。このスクリプトは、Pingフェデレーション統合にも使用できます。
	スクリプトで使用する空の構成ファイル configure-storagegrid.py
	StorageGRIDコンテナ導入用のRHELホストを設定するためのサンプルのAnsibleのロールとプレイブック。必要に応じて、ロールまたはプレイブックをカスタマイズできます。
	Active DirectoryまたはPingフェデレーションを使用してシングルサインオン（SSO）が有効になっている場合にグリッド管理APIにサインインするために使用できるPythonスクリプトの例。

パスとファイル名	製品説明
	関連するPythonスクリプトによって呼び出され、AzureとのSSO対話を実行するヘルパースクリプト <code>storagegrid-ssoauth-azure.py</code> 。
	StorageGRID の API スキーマ  注：アップグレードを実行する前に、これらのスキーマを使用して、アップグレード互換性テスト用の非本番環境のStorageGRID 環境がない場合、StorageGRID 管理APIを使用するように記述したコードが新しいStorageGRID リリースと互換性があることを確認できます。

### Ubuntu または Debian

パスとファイル名	製品説明
	StorageGRID ダウンロードファイルに含まれているすべてのファイルについて説明するテキストファイル。
	テスト環境やコンセプトの実証環境に使用できる、非本番環境のNetAppライセンスファイル。
	Ubuntu ホストまたは Debian ホストに StorageGRID ノードイメージをインストールするための DEB パッケージ。
	ファイルのMD5チェックサム <code>/debs/storagegrid-webscale-images-version-SHA.deb</code> 。
	Ubuntu ホストまたは Debian ホストに StorageGRID ホストサービスをインストールするための DEB パッケージ。
導入スクリプトツール	製品説明
	StorageGRID システムの設定を自動化するための Python スクリプト。
	StorageGRID アプライアンスの設定を自動化するための Python スクリプト。

パスとファイル名	製品説明
	シングルサインオンが有効な場合にグリッド管理 API にサインインするために使用できる Python スクリプトの例。このスクリプトは、Pingフェデレーション統合にも使用できます。
	スクリプトで使用する構成ファイルの例 configure-storagegrid.py。
	スクリプトで使用する空の構成ファイル configure-storagegrid.py。
	StorageGRID コンテナ導入用の Ubuntu ホストまたは Debian ホストを設定するためのサンプルの Ansible のロールとプレイブック。必要に応じて、ロールまたはプレイブックをカスタマイズできます。
	Active DirectoryまたはPingフェデレーションを使用してシングルサインオン (SSO) が有効になっている場合にグリッド管理APIにサインインするために使用できるPythonスクリプトの例。
	関連するPythonスクリプトによって呼び出され、AzureとのSSO対話を実行するヘルパースクリプト storagegrid-ssoauth-azure.py。
	StorageGRID の API スキーマ  注：アップグレードを実行する前に、これらのスキーマを使用して、アップグレード互換性テスト用の非本番環境のStorageGRID 環境がない場合、StorageGRID 管理APIを使用するように記述したコードが新しいStorageGRID リリースと互換性があることを確認できます。

## VMware

パスとファイル名	製品説明
	StorageGRID ダウンロードファイルに含まれているすべてのファイルについて説明するテキストファイル。
	製品サポートのない無償ライセンス。

パスとファイル名	製品説明
	グリッドノード仮想マシンを作成するためのテンプレートとして使用される仮想マシンディスクファイル。
	(.mf`プライマリ管理ノードを導入するためのOpen Virtualization Formatテンプレートファイル) (.ovfとマニフェストファイル
	テンプレートファイル(.ovf) とマニフェストファイル(.mf) 。非プライマリ管理ノードを導入するためのものです。
	テンプレートファイル(.ovf) とマニフェストファイル(.mf) を使用してゲートウェイノードを導入します。
	(.mf`仮想マシンベースのストレージノードを導入するためのテンプレートファイル(.ovfとマニフェストファイル)
導入スクリプトツール	製品説明
	仮想グリッドノードの導入を自動化するための Bash シェルスクリプト。
	スクリプトで使用する構成ファイルの例 <code>deploy-vsphere-ovftool.sh</code> 。
	StorageGRID システムの設定を自動化するための Python スクリプト。
	StorageGRID アプライアンスの設定を自動化するための Python スクリプト。
	シングルサインオン (SSO) が有効な場合にグリッド管理APIにサインインするために使用できるPythonスクリプトの例。このスクリプトは、Pingフェデレーション統合にも使用できません。
	スクリプトで使用する構成ファイルの例 <code>configure-storagegrid.py</code> 。
	スクリプトで使用する空の構成ファイル <code>configure-storagegrid.py</code> 。

パスとファイル名	製品説明
	Active DirectoryまたはPingフェデレーションを使用してシングルサインオン（SSO）が有効になっている場合にグリッド管理APIにサインインするために使用できるPythonスクリプトの例。
	関連するPythonスクリプトによって呼び出され、AzureとのSSO対話を実行するヘルパースクリプト <code>storagegrid-ssoauth-azure.py</code> 。
	StorageGRID の API スキーマ  注：アップグレードを実行する前に、これらのスキーマを使用して、アップグレード互換性テスト用の非本番環境のStorageGRID 環境がない場合、StorageGRID 管理APIを使用するように記述したコードが新しいStorageGRID リリースと互換性があることを確認できます。

## StorageGRIDインストールファイルを確認する

必要に応じて、StorageGRIDインストールアーカイブ内のファイルを手動で検証できます。

開始する前に

あなたが持っている["検証パッケージをダウンロードしました"](#)から ["ネットアップの StorageGRID ダウンロードページ"](#)。

手順

1. 検証パッケージからアーティファクトを抽出します。

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

2. これらのアーチファクトが抽出されたことを確認します。

- リーフ証明書： `Leaf-Cert.pem`
- 証明書チェーン： `CA-Int-Cert.pem`
- タイムスタンプ応答チェーン： `TS-Cert.pem`
- チェックサムファイル： `sha256sum`
- チェックサム署名： `sha256sum.sig`
- タイムスタンプ応答ファイル： `sha256sum.sig.tsr`

3. チェーンを使用して、リーフ証明書が有効であることを確認します。

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

予想される出力： Leaf-Cert.pem: OK

- リーフ証明書の期限が切れたためにSTEP\_2\_FAILEDが発生した場合は、ファイルを使用して `tsr` 確認します。

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

予想される出力には： Verification: OK

- リーフ証明書から公開鍵ファイルを作成します。

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

予想される出力： *NONE*

- 公開鍵を使用してファイルを `sha256sum.sig` 検証し `sha256sum` ます。

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

予想される出力： Verified OK

- 新しく作成したチェックサムと比較してファイルの内容を確認し `sha256sum` ます。

例： `sha256sum -c sha256sum`

予期される出力: + *<filename>*: OK `<filename>` は、ダウンロードしたアーカイブファイルの名前です。

- "[残りの手順を完了する](#)"インストール アーカイブから適切なファイルを抽出して選択します。

## StorageGRIDのソフトウェア要件

仮想マシンを使用して、あらゆるタイプのStorageGRIDノードをホストできます。グリッドノードごとに仮想マシンが1つ必要です。

## RHEL

RHEL にStorageGRIDをインストールするには、サードパーティのソフトウェア パッケージをいくつかインストールする必要があります。サポートされている一部の Linux ディストリビューションには、これらのパッケージがデフォルトで含まれていません。StorageGRIDインストールがテストされるソフトウェア パッケージのバージョンには、このページに記載されているものが含まれます。

これらのパッケージのいずれかを必要とするLinuxディストリビューションおよびコンテナランタイムインストールオプションを選択し、それらがLinuxディストリビューションによって自動的にインストールされない場合は、プロバイダまたはLinuxディストリビューションのサポートベンダーから入手可能な場合は、ここに記載されているいずれかのバージョンをインストールします。それ以外の場合は、ベンダーが提供しているデフォルトのパッケージバージョンを使用します。

すべてのインストールオプションには、PodmanまたはDockerのいずれかが必要です。両方のパッケージをインストールしないでください。インストールオプションに必要なパッケージのみをインストールします。



ソフトウェアのみの環境のコンテナエンジンとしてのDockerのサポートは廃止されました。Dockerは、今後のリリースで別のコンテナエンジンに置き換えられる予定です。

### テスト済みのPythonバージョン

- 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

### テスト済みのPodmanバージョン

- 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

### テスト済みのDockerバージョン



Dockerのサポートは廃止され、今後のリリースで削除される予定です。

- 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 と Debian

UbuntuまたはDebianにStorageGRIDをインストールするには、サードパーティのソフトウェアパッケージをインストールする必要があります。一部のサポートされているLinuxディストリビューションには、デフォルトでこれらのパッケージが含まれていません。StorageGRIDのインストールがテストされているソフトウェアパッケージのバージョンには、このページに記載されているバージョンも含まれます。

これらのパッケージのいずれかを必要とするLinuxディストリビューションおよびコンテナランタイムインストールオプションを選択し、それらがLinuxディストリビューションによって自動的にインストールされない場合は、プロバイダまたはLinuxディストリビューションのサポートベンダーから入手可能な場合は、ここに記載されているいずれかのバージョンをインストールします。それ以外の場合は、ベンダーが提供しているデフォルトのパッケージバージョンを使用します。

すべてのインストールオプションには、PodmanまたはDockerのいずれかが必要です。両方のパッケージをインストールしないでください。インストールオプションに必要なパッケージのみをインストールします。



ソフトウェアのみの環境のコンテナエンジンとしてのDockerのサポートは廃止されました。Dockerは、今後のリリースで別のコンテナエンジンに置き換えられる予定です。

### テスト済みのPythonバージョン

- 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

### テスト済みのPodmanバージョン

- 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

#### テスト済みのDockerバージョン



Dockerのサポートは廃止され、今後のリリースで削除される予定です。

- 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 ハイパーバイザー

準備が整った物理サーバに VMware vSphere ハイパーバイザーをインストールする必要があります。VMware ソフトウェアをインストールする前に、ハードウェアが正しく設定されている必要があります（ファームウェアバージョンと BIOS 設定を含む）。

- インストールする StorageGRID システムのネットワークをサポートできるように、ハイパーバイザーのネットワークを設定します。

##### "ネットワークのガイドライン"

- データストアが、グリッドノードをホストするために必要な仮想マシンと仮想ディスクに十分な大きさであることを確認します。
- 複数のデータストアを作成する場合は、仮想マシン作成時に各グリッドノードに使用するデータストアを簡単に識別できるように、各データストアに名前を付けます。

#### ESXホストの構成要件



各 ESX ホストでネットワークタイムプロトコル（NTP）を適切に設定する必要があります。ホストの時刻が正しくないと、データ損失などのマイナスの影響が生じる可能性があります。

#### VMware 構成要件

StorageGRIDノードを導入する前に、VMware vSphereとvCenterをインストールして設定する必要があります。

サポートされるVMware vSphere HypervisorおよびVMware vCenter Serverソフトウェアのバージョンについては、を参照してください "[NetApp Interoperability Matrix Tool](#)".

これらのVMware製品をインストールするために必要な手順については、VMwareのドキュメントを参照してください。

## StorageGRIDのCPUとRAMの要件

StorageGRIDソフトウェアをインストールする前に、ハードウェアの確認と設定を行って、StorageGRIDシステムをサポートできる状態にしておきます。

各 StorageGRID ノードに必要な最小リソースは次のとおりです。

- CPU コア：ノードあたり 8 個
- RAM：使用可能なRAMの合計容量と、システムで実行されているStorageGRID以外のソフトウェアの容量によって異なります。
  - 通常、ノードあたり24GB以上、システムRAMの合計より2~16GB少ない
  - 約5,000個のバケットを格納するテナントごとに64GB以上

ソフトウェアベースのメタデータのみノードリソースは、既存のストレージノードリソースと一致している必要があります。例：

- 既存のStorageGRIDサイトでSG6000またはSG6100アプライアンスを使用している場合は、ソフトウェアベースのメタデータのみノードが次の最小要件を満たしている必要があります。
  - 128GBのRAM
  - 8コアCPU
  - 8TB SSDまたはCassandraデータベース用同等のストレージ (rangedb/0)
- 既存のStorageGRIDサイトが 24 GB RAM、8 コア CPU、3 TB または 4 TB のメタデータ ストレージを備えた仮想ストレージ ノードを使用している場合、ソフトウェア ベースのメタデータ専用ノードでは同様のリソース (24 GB RAM、8 コア CPU、4 TB のメタデータ ストレージ (rangedb/0)) を使用する必要があります。

新しいStorageGRIDサイトを追加するときは、新しいサイトの総メタデータ容量が少なくとも既存のStorageGRIDサイトと一致し、新しいサイトのリソースが既存のStorageGRIDサイトのストレージノードと一致している必要があります。



「Linux」は、RHEL、Ubuntu、または Debian のデプロイメントを指します。サポートされているバージョンのリストについては、"[NetApp Interoperability Matrix Tool \(IMT\)](#)"。

## Linux

それぞれの物理ホストまたは仮想ホストで実行する StorageGRID ノードの数が、利用可能な CPU コアや物理 RAM を超えないようにしてください。ホストが StorageGRID 専用でない場合（非推奨）は、他のアプリケーションのリソース要件を考慮してください。

## VMware

VMwareでは、仮想マシンごとに1ノードがサポートされます。StorageGRIDノードが使用可能な物理RAMを超えていないことを確認します。各仮想マシンは、StorageGRIDを実行する専用にする必要があります。



CPU とメモリの使用状況を定期的に監視して、ワークロードに継続的に対応できるようにします。たとえば、仮想ストレージノードの RAM 割り当てと CPU 割り当てを 2 倍にすると、StorageGRID アプライアンスノードの場合と同様のリソースが提供されます。また、ノードあたりのメタデータの量が 500GB を超える場合は、ノードあたりの RAM を 48GB 以上に増やすことを検討してください。オブジェクトメタデータストレージの管理、Metadata Reserved Space設定の拡張、およびCPUとメモリの使用状況の監視については["管理"](#)、["監視"](#)および["アップグレード"](#)StorageGRIDの手順を参照してください。

基盤となる物理ホストでハイパースレッディングが有効である場合は、ノードあたり 8 個の仮想コア（4 個の物理コア）で構成できます。基盤となる物理ホストでハイパースレッディングが有効でない場合は、ノードあたり 8 個の物理コアを用意する必要があります。

仮想マシンをホストとして使用する場合、VM のサイズと数を制御可能であれば、StorageGRID ノードごとに 1 つの VM を使用し、それに応じて VM のサイズを設定する必要があります。

(RHEL、Debian、Ubuntu のみ) 実稼働環境では、同じ物理ストレージ ハードウェアまたは仮想ホスト上で複数のストレージ ノードを実行しないでください。単一のStorageGRID展開内の各ストレージ ノードは、独自の分離された障害ドメインに配置する必要があります。単一のハードウェア障害が単一のストレージ ノードにのみ影響するようにすれば、オブジェクト データの耐久性と可用性を最大限に高めることができます。

も参照してください["ストレージとパフォーマンスの要件"](#)。

## StorageGRIDのストレージとパフォーマンスの要件

初期設定と将来のストレージ拡張に対応できる十分なスペースを確保できるよう、StorageGRID ノードのストレージ要件を把握しておく必要があります。

ストレージとパフォーマンスの要件は、ソフトウェア ベースのノード実装によって異なります。



「Linux」は、RHEL、Ubuntu、または Debian のデプロイメントを指します。サポートされているバージョンのリストについては、["NetApp Interoperability Matrix Tool \(IMT\)"](#)。

### ストレージカテゴリ

StorageGRID ノードに必要なストレージは、3 つの論理カテゴリに分類されます。

- **コンテナプール** \*—ノードコンテナ用のパフォーマンス階層（10K SAS または SSD）ストレージ。StorageGRID ノードをサポートするホストにコンテナエンジンをインストールして設定するときに、コンテナエンジンストレージドライバに割り当てられます。

- \* システムデータ \* — システムデータとトランザクションログのノード単位の永続的ストレージ用のパフォーマンス階層（10K SAS または SSD）ストレージ。StorageGRID ホストサービスで個々のノードにマッピングされて使用されます。
- \* オブジェクトデータ \* — オブジェクトデータとオブジェクトメタデータの永続的なストレージを実現するパフォーマンス階層（10K SAS または SSD）のストレージと大容量階層（NL-SAS / SATA）のストレージ。

カテゴリに関係なく、いずれのストレージにも RAID ベースのブロックデバイスを使用する必要があります。非冗長ディスク、SSD、JBODはサポートされていません。いずれのカテゴリのストレージにも、共有またはローカルのRAIDストレージを使用できます。ただし、StorageGRID のノード移行機能を使用する場合は、システムデータとオブジェクトデータの両方を共有ストレージに格納する必要があります。詳細については、を参照してください "[ノードコンテナの移行要件](#)"。

## パフォーマンス要件

コンテナプールのボリューム、システムデータのボリューム、およびオブジェクトメタデータのボリュームのパフォーマンスは、システム全体のパフォーマンスに大きく影響します。ボリュームのディスクパフォーマンスが、レイテンシ、1秒あたりの入出力操作（IOPS）、スループットの点で適切になるように、それらのボリュームにはパフォーマンス階層（10K SAS または SSD）のストレージを使用します。オブジェクトデータの永続的なストレージには、大容量階層（NL-SAS / SATA）のストレージを使用できます。

コンテナプール、システムデータ、およびオブジェクトデータ用のボリュームでは、ライトバックキャッシュを有効にする必要があります。キャッシュは、保護されたメディアまたは永続的なメディアに配置する必要があります。

## NetApp ONTAPストレージを使用するホストの要件

StorageGRID ノードがNetApp ONTAP システムから割り当てられたストレージを使用している場合は、ボリュームでFabricPool 階層化ポリシーが有効になっていないことを確認してください。StorageGRIDノードで使用するボリュームでFabricPool階層化を無効にすると、トラブルシューティングとストレージの処理が簡単になります。



FabricPoolを使用して、StorageGRIDに関連するデータをStorageGRID自体に階層化しないでください。StorageGRIDデータをStorageGRIDに階層化すると、トラブルシューティングや運用が複雑になります。

## 必要なホストの数

各 StorageGRID サイトに、少なくとも 3 つのストレージノードが必要です。



本番環境では、1つの物理ホストまたは仮想ホストで複数のストレージノードを実行しないでください。各ストレージノードに専用のホストを使用すると、分離された障害ドメインが提供されます。

管理ノードやゲートウェイノードなど、他のタイプのノードは、同じホストに導入するか、必要に応じて独自の専用ホストに導入することができます。



ディスクSnapshotを使用してグリッドノードをリストアすることはできません。代わりに、各タイプのノードの手順を参照して"[グリッドノードのリカバリ](#)"ください。

## 各ノードのストレージボリュームの数

次の表に、ホストに導入するノードの種類別に、各ホストに必要なストレージボリューム（LUN）の数と各LUNに必要な最小サイズを示します。

テストで使用できるLUNの最大サイズは39TBです。



これらはホストごとの数値を示したものであり、グリッド全体の数値ではありません。

LUNの用途	ストレージのカテゴリ	LUN数	LUNあたりの最小サイズ
コンテナエンジンのストレージプール	コンテナプール	1	ノードの総数 × 100GB
`/var/local`ボリューム	システムデータ	このホストのノードごとに1個	100GB
ストレージノード	オブジェクトデータ	このホストのストレージノードごとに3個  注: Linux ソフトウェアベースのストレージノードには、1～48個のストレージボリュームを設定できます。VMware ソフトウェアベースのストレージノードには、1～16個のストレージボリュームを設定できます。少なくとも3つのストレージボリュームが推奨されます。	12 TB (4 TB/LUN、最小)  検証済みの最大 LUN サイズ: 39TB。  <a href="#">見るストレージノードのストレージ要件詳細</a> についてはこちらをご覧ください。
ストレージノード (メタデータのみ)	オブジェクトメタデータ	1	4 TB/LUN (最小)  検証済みの最大 LUN サイズ: 39TB。  <a href="#">見るストレージノードのストレージ要件詳細</a> についてはこちらをご覧ください。  注: メタデータのためのストレージノードに必要な rangedb は1つだけです。
管理ノードの監査ログ	システムデータ	このホストの管理ノードごとに1個	200GB

LUNの用途	ストレージのカテゴリ	LUN数	LUN あたりの最小サイズ
管理ノードのテーブル	システムデータ	このホストの管理ノードごとに 1 個	200GB



設定されている監査レベル、S3 オブジェクトキー名などのユーザー入力のサイズ、保存する必要がある監査ログ データの量に応じて、各管理ノードの監査ログ LUN のサイズを増やす必要がある場合があります。通常、グリッドは S3 操作ごとに約 1 KB の監査データを生成します。つまり、200 GB の LUN は 1 日あたり 7,000 万回の操作、または 2 ~ 3 日間で 1 秒あたり 800 回の操作をサポートすることになります。

## ホストの最小ストレージスペース

次の表に、各タイプのノードに必要な最小ストレージスペースを示します。この表を参照して、ホストに導入するノードの種類に応じて、ストレージカテゴリごとにホストで確保しなければならない最小ストレージ容量を決定できます。



ディスクSnapshotを使用してグリッドノードをリストアすることはできません。代わりに、各タイプのノードの手順を参照して["グリッドノードのリカバリ"](#)ください。

各ノード ホストには、OS 用に 100 GB の LUN が必要です。

ノードのタイプ	コンテナプール	システムデータ	オブジェクトデータ
ストレージノード	100GB	100GB	4,000GB
管理ノード	100GB	500 GB (3 LUN)	_ 該当なし _
ゲートウェイノード	100GB	100GB	_ 該当なし _

## 例: ホストまたは仮想マシンのストレージ要件の計算

同じホストまたは仮想マシンに、ストレージ ノード 1 つ、管理ノード 1 つ、ゲートウェイ ノード 1 つの 3 つのノードを展開する予定であるとします。ホストには少なくとも 9 つのストレージ ボリュームを提供する必要があります。ノード コンテナには最低 300 GB のパフォーマンス層ストレージ、システム データとトランザクション ログには 700 GB のパフォーマンス層ストレージ、オブジェクト データには 12 TB の容量層ストレージが必要です。

### Linuxホストの例

ノードのタイプ	LUNの用途	LUN数	LUNサイズ
ストレージノード	コンテナエンジンのストレージプール	1	300GB (100GB/ノード)
ストレージノード	`/var/local`ボリューム	1	100GB
ストレージノード	オブジェクトデータ	3	12TB (4TB / LUN)
管理ノード	`/var/local`ボリューム	1	100GB
管理ノード	管理ノードの監査ログ	1	200GB
管理ノード	管理ノードのテーブル	1	200GB
ゲートウェイノード	`/var/local`ボリューム	1	100GB
• 合計 *		<b>9</b>	<ul style="list-style-type: none"> <li>• コンテナプール: * 300GB</li> <li>システムデータ: 700 GB</li> <li>• オブジェクトデータ: 12、000GB</li> </ul>

### VMware仮想マシンの例

ノードのタイプ	LUNの用途	LUN数	LUNサイズ
ストレージノード	OSボリューム	1	100GB
ストレージノード	オブジェクトデータ	3	12TB (4TB / LUN)
管理ノード	OSボリューム	1	100GB
管理ノード	管理ノードの監査ログ	1	200GB
管理ノード	管理ノードのテーブル	1	200GB
ゲートウェイノード	OSボリューム	1	100GB

ノードのタイプ	LUNの用途	LUN数	LUNサイズ
・ 合計 *		8	システムデータ: 700 GB ・ オブジェクトデータ : 12、000GB

### ストレージノードの特定のストレージ要件

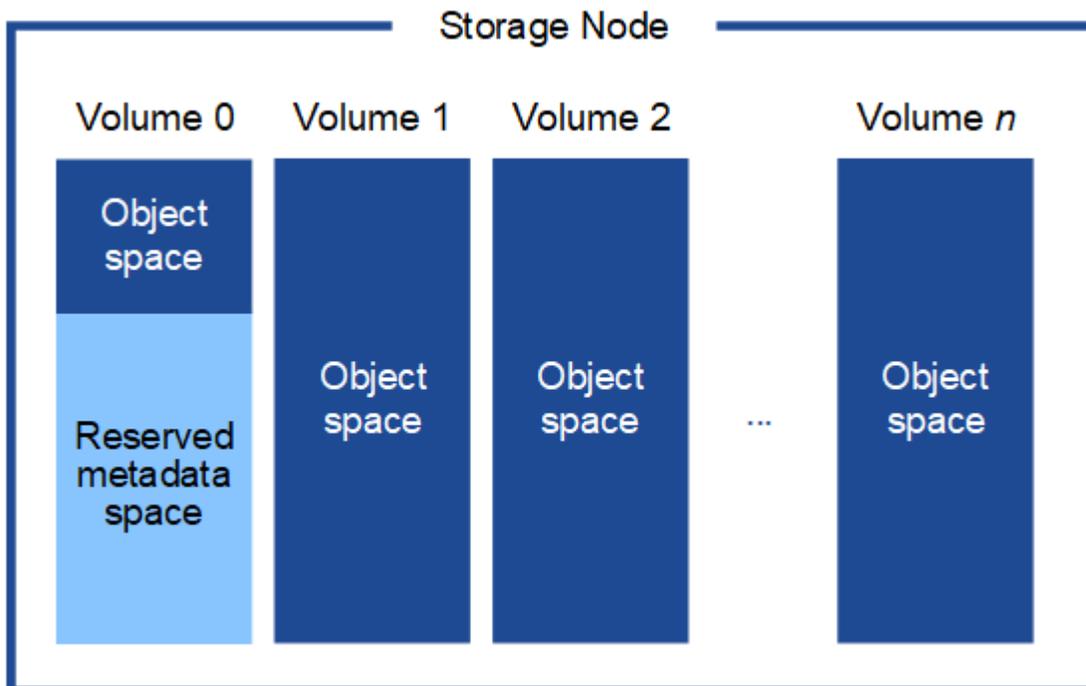
Linux と VMware では、ストレージ ノードに対するストレージ要件が異なります。

- ・ Linuxソフトウェアベースのストレージノードは1~48個のストレージボリュームを持つことができる
- ・ VMwareソフトウェアベースのストレージノードは1~16個のストレージボリュームを持つことができます。
- ・ 3 つ以上のストレージ ボリュームが推奨されます。
- ・ 各ストレージ ボリュームは 4 TB 以上である必要があります。



アプライアンス ストレージ ノードには、最大 48 個のストレージ ボリュームも設定できます。

図に示すように、StorageGRID は各ストレージノードのストレージボリューム 0 にオブジェクトメタデータ用のスペースをリザーブします。ストレージボリューム 0 の残りのスペースとストレージノード内のその他のストレージボリュームは、オブジェクトデータ専用で使用されます。



冗長性を確保し、オブジェクトメタデータを損失から保護するために、StorageGRID は各サイトのシステム内のすべてのオブジェクトにメタデータのコピーを 3 つずつ格納します。オブジェクトメタデータの 3 つのコピーが各サイトのすべてのストレージノードに均等に分散されます。

メタデータのみストレージノードを含むグリッドをインストールする場合は、グリッドにオブジェクトスト

レーズ用のノードの最小数も含まれている必要があります。メタデータ専用ストレージノードの詳細については、[を参照してください](#)"[ストレージノードのタイプ](#)".

- 単一サイトのグリッドの場合は、オブジェクトとメタデータ用に少なくとも2つのストレージノードが設定されます。
- マルチサイトグリッドの場合は、サイトごとに少なくとも1つのストレージノードがオブジェクトとメタデータ用に設定されます。

新しいストレージノードのボリューム 0 にスペースを割り当てる場合は、そのノードのすべてのオブジェクトメタデータの一部に対して十分なスペースを確保する必要があります。

- 少なくとも 4TB をボリューム 0 に割り当てる必要があります。



ストレージノードでストレージボリュームを1つだけ使用していて、そのボリュームに4TB以下を割り当てると、ストレージノードが起動時にストレージ読み取り専用状態になり、オブジェクトメタデータのみが格納される可能性があります。



ボリューム0への割り当てが500GB未満の場合（非本番環境での使用のみ）は、ストレージボリュームの容量の10%がメタデータ用にリザーブされます。

- ソフトウェアベースのメタデータのみノードリソースは、既存のストレージノードリソースと一致している必要があります。例：
  - 既存のStorageGRIDサイトでSG6000またはSG6100アプライアンスを使用している場合は、ソフトウェアベースのメタデータのみノードが次の最小要件を満たしている必要があります。
    - 128GBのRAM
    - 8コアCPU
    - 8TB SSDまたはCassandraデータベース用同等のストレージ (rangedb/0)
  - 既存のStorageGRIDサイトが 24 GB RAM、8 コア CPU、3 TB または 4 TB のメタデータ ストレージを備えた仮想ストレージ ノードを使用している場合、ソフトウェア ベースのメタデータ専用ノードでは同様のリソース (24 GB RAM、8 コア CPU、4 TB のメタデータ ストレージ (rangedb/0)) を使用する必要があります。

新しいStorageGRIDサイトを追加するときは、新しいサイトの総メタデータ容量が少なくとも既存のStorageGRIDサイトと一致し、新しいサイトのリソースが既存のStorageGRIDサイトのストレージノードと一致している必要があります。

- 新しいシステム (StorageGRID 11.6以降) をインストールし、各ストレージノードに128GB以上のRAMがある場合は、8TB以上をボリューム0に割り当てます。ボリューム 0 に大きな値を設定すると、各ストレージノードでメタデータに使用できるスペースが増加する可能性があります。
- サイトに複数のストレージノードを設定する場合は、可能であればボリューム 0 にも同じ設定を使用します。サイトにサイズが異なるストレージノードがある場合、ボリューム 0 が最も小さいストレージノードがそのサイトのメタデータ容量を決定します。

詳細については、[を参照してください](#)"[オブジェクトメタデータストレージを管理する](#)".

## Linux の StorageGRID ノードコンテナ移行要件

ノード移行機能を使用すると、ホスト間でノードを手動で移動できます。通常、両方の

ホストが同じ物理データセンターにあります。



「Linux」は、RHEL、Ubuntu、または Debian のデプロイメントを指します。サポートされているバージョンのリストについては、"[NetApp Interoperability Matrix Tool \(IMT\)](#)"。

ノード移行を使用すると、グリッドの運用を中断せずに物理ホストのメンテナンスを実行できます。物理ホストをオフラインにする前に、すべてのStorageGRID ノードを一度に1つずつ別のホストに移動します。ノードを1つずつ移行するため、それぞれのダウンタイムはごくわずかであり、グリッドサービスの運用や可用性には影響しません。

StorageGRID のノード移行機能を使用する場合は、いくつかの追加の要件を満たす必要があります。

- 単一の物理データセンターのホスト間で一貫したネットワークインターフェイス名を使用する必要があります
- StorageGRID のメタデータとオブジェクトのリポジトリボリューム用に、単一の物理データセンターのすべてのホストからアクセスできる共有ストレージを用意する必要があります。たとえば、NetApp E シリーズストレージアレイなどを使用できます。

仮想ホストを使用していて、基盤となるハイパーバイザーレイヤでVMの移行がサポートされている場合は、StorageGRID のノード移行機能の代わりにこの機能を使用できます。その場合、これらの追加要件は無視してかまいません。

移行またはハイパーバイザーのメンテナンスを実行する前に、ノードを正常にシャットダウンしてください。の手順を参照してください"[グリッドノードをシャットダウンしています](#)"。

**VMware** のライブマイグレーションはサポートされていません

VMware VMでベアメタルインストールを実行する場合、OpenStack Live MigrationとVMwareのライブvMotion原因で仮想マシンのクロック時間がジャンプするため、どのタイプのグリッドノードでもサポートされていません。まれにではありますが、クロック時間が不正確だとデータや設定の更新が失われることがあります。

コールドマイグレーションはサポートされています。コールドマイグレーションでは、StorageGRID ノードをホスト間で移行する前にシャットダウンします。の手順を参照してください"[グリッドノードをシャットダウンしています](#)"。

一貫したネットワークインターフェイス名

ノードを別のホストに移動する場合、StorageGRID ホストサービスでは、ノードが現在の場所で使用している外部ネットワーク接続を新しい場所でも確実に複製できるようにする必要があります。これは、ホスト内で一貫したネットワークインターフェイス名を使用することで実現されます。

たとえば、Host1 で実行されている StorageGRID NodeA で、インターフェイスのマッピングが次のように設定されているとします。

```
eth0  ────>  bond0.1001
eth1  ────>  bond0.1002
eth2  ────>  bond0.1003
```

矢印の左側は、StorageGRID コンテナ内から見た従来のインターフェイス（グリッドネットワーク、管理ネットワーク、およびクライアントネットワークのインターフェイス）です。矢印の右側は、これらのネットワークを提供する実際のホストインターフェイスに対応しています。この3つのVLANインターフェイスは、同じ物理インターフェイスボンドに従属します。

このNodeAをHost2に移行するとします。Host2にbond0.1001、bond0.1002、およびbond0.1003という名前のインターフェイスがある場合、Host2では同じ名前のインターフェイスがHost1と同じ接続を提供すると仮定して、移動が許可されます。Host2に同じ名前のインターフェイスがなければ、移動は許可されません。

複数のホスト間でネットワークインターフェイスの名前を統一するには、さまざまな方法があります。いくつかの例については、を参照してください["ホストネットワークを設定する"](#)。

## 共有ストレージ

オーバーヘッドを抑えて迅速にノードを移行するために、StorageGRID ノード移行機能ではノードのデータの物理的な移動は行いません。代わりに、エクスポート処理とインポート処理を組み合わせ、次のようにノードが移行されます。

- 「ノードのエクスポート」処理で、HostAで実行されているノードコンテナから永続的な状態の少量のデータが抽出され、そのノードのシステムデータボリュームにキャッシュされます。そのあと、HostAのノードコンテナのインスタンス化が解除されます。
- 「ノードのインポート」処理では、HostAと同じネットワークインターフェイスマッピングとブロックストレージマッピングを使用するHostBのノードコンテナがインスタンス化されます。次に、キャッシュされた永続状態データが新しいインスタンスに挿入されます。

この処理では、ノードのすべてのシステムデータボリュームとオブジェクトストレージボリュームにHostAとHostBの両方からアクセスできないと移行は実行できません。また、HostAとHostBで同じLUNを参照するように、同じ名前を使用してノードにマッピングされている必要があります。

次の例は、StorageGRIDストレージノード用のブロックデバイスマッピングの1つのソリューションを示しています。これらのホストではDMマルチパスを使用しており、でaliasフィールドを使用し`/etc/multipath.conf`で、すべてのホストで使用可能な一貫性のあるわかりやすい名前をブロックデバイスに提供しています。

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

## ホストを準備する (Linux)

## StorageGRIDがホスト全体の設定を変更する方法

ベアメタルシステムでは、StorageGRIDによってホスト全体の設定が一部変更され、`sysctl` します。



「Linux」は、RHEL、Ubuntu、または Debian のデプロイメントを指します。サポートされているバージョンのリストについては、"[NetApp Interoperability Matrix Tool \(IMT\)](#)"。

次の変更が行われます。

```
# 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 グリッドホストに **StorageGRID** をインストールします

すべての Linux グリッド ホストにStorageGRIDをインストールする必要があります。サポートされているバージョンの一覧については、NetApp相互運用性マトリックス ツールを使用してください。

開始する前に

お使いのオペレーティングシステムが、以下に示すStorageGRIDのカーネルバージョンの最小要件を満たしていることを確認してください。コマンドを使用し `uname -r` でオペレーティングシステムのカーネルバージョンを確認するか、OSベンダーに問い合わせてください。



「Linux」は、RHEL、Ubuntu、または Debian のデプロイメントを指します。サポートされているバージョンのリストについては、"[NetApp Interoperability Matrix Tool \(IMT\)](#)"。

## RHEL

RHELバージョン	最小カーネルバージョン	カーネルパッケージ名
8.8 (廃止)	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 (廃止)	5.14.0~70.22.1.el9_0.x86_64	kernel-5.14.0-70.22.1.el9_0.x86_64
9.2 (廃止)	5.14.0~284.11.1.el9_2.x86_64	カーネル-5.14.0-284.11.1.el9_2.x86_64
9.4	5.14.0~427.18.1.el9_4.x86_64	カーネル-5.14.0-427.18.1.el9_4.x86_64
9.6	5.14.0-570.18.1.el9_6.x86_64	カーネル-5.14.0-570.18.1.el9_6.x86_64

## Ubuntu

注: Ubuntuバージョン18.04および20.04のサポートは廃止され、今後のリリースで削除される予定です。

Ubuntuバージョン	最小カーネルバージョン	カーネルパッケージ名
22.04.1	5.15.0-47-汎用	linux-image-5.15.0-47-generic/jammy-updates、jammy-security、現在は5.15.0-47.51
24.04	6.8.0-31-汎用	linux-image-6.8.0-31-generic/noble、現在は6.8.0-31.31

## Debian

注意: Debianバージョン11のサポートは非推奨となり、今後のリリースで削除される予定です。

Debianバージョン	最小カーネルバージョン	カーネルパッケージ名
11 (廃止)	5.10.0-18-amd64	linux-image-5.10.0-18-amd64/stable、現在は5.10.150-1
12	6.1.0-9-amd64	linux-image-6.1.0-9-amd64/stable、現在は6.1.27-1

## 手順

1. ディストリビュータの指示または標準の手順に従って、すべての物理グリッドホストまたは仮想グリッドホストに Linux をインストールします。



グラフィカルデスクトップ環境をインストールしないでください。

- RHEL をインストールするときに標準の Linux インストーラーを使用している場合は、「コンピューティング ノード」ソフトウェア構成 (使用可能な場合) または「最小インストール」ベース環境を選択します。
  - Ubuntu をインストールするときは、標準システムユーティリティ\*を選択する必要があります。**Ubuntu** ホストへの **ssh** アクセスを有効にするには、\***OpenSSH** サーバー を選択することをお勧めします。その他のオプションはすべてオフのままにできます。
2. RHEL の Extras チャンネルを含むすべてのホストがパッケージ リポジトリにアクセスできることを確認します。
  3. スワップが有効になっている場合：
    - a. 次のコマンドを実行します。 `$ sudo swapoff --all`
    - b. からすべてのスワップエントリを削除し `/etc/fstab` で、設定を維持します。



スワップを完全に無効にできないと、パフォーマンスが大幅に低下する可能性があります

## Ubuntu および Debian 上の StorageGRID の AppArmor プロファイルについて説明します

自社で導入した Ubuntu 環境を運用し、AppArmor の必須のアクセス制御システムを使用している場合、ベースシステムにインストールするパッケージに関連付けられた AppArmor プロファイルが、StorageGRID と一緒にインストールされた対応するパッケージによってブロックされる可能性があります。

デフォルトでは、AppArmor プロファイルは、ベースのオペレーティングシステムにインストールするパッケージに対してインストールされます。StorageGRID システムコンテナからこれらのパッケージを実行すると、AppArmor プロファイルがブロックされます。DHCP、MySQL、NTP、tcdump のベースパッケージが AppArmor と競合するほか、これら以外のベースパッケージも競合する可能性があります。

AppArmor プロファイルの対処方法としては、次の 2 つの選択肢があります。

- ベースシステムにインストールされたパッケージのうち、StorageGRID システムコンテナに含まれるパッケージと重複するパッケージのプロファイルを個々に無効にする。各プロファイルを無効にすると、StorageGRID ログファイルに AppArmor が有効であることを示すエントリが表示されます。

次のコマンドを使用します。

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

- 例： \*

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

- AppArmor 全体を無効にする。Ubuntu 9.10以降の場合は、Ubuntuオンラインコミュニティの手順に従ってください。"[AppArmor を無効にします](#)"新しいバージョンのUbuntuでは、AppArmorを完全に無効にできない場合があります。

AppArmorを無効にすると、StorageGRIDログファイルにAppArmorが有効であることを示すエントリは表示されません。

## Linux デプロイメント用の StorageGRID ホストネットワークを設定する

ホストへの Linux のインストールの完了後、このあとに導入する StorageGRID ノードにマッピングする一連のネットワークインターフェイスを準備するために、各ホストでいくつかの追加の設定が必要になることがあります。



「Linux」は、RHEL、Ubuntu、または Debian のデプロイメントを指します。サポートされているバージョンのリストについては、"[NetApp Interoperability Matrix Tool \(IMT\)](#)"。

開始する前に

- を確認しておきます"[StorageGRID ネットワークのガイドライン](#)"。
- に関する情報を確認しておき"[ノードコンテナの移行要件](#)"ます。
- 仮想ホストを使用している場合は、[MAC アドレスのクローニングに関する考慮事項と推奨事項](#)ホスト ネットワークを構成する前に。



VM をホストとして使用する場合は、仮想ネットワークアダプタとして VMXNET 3 を選択する必要があります。VMware E1000 ネットワークアダプタは、特定の Linux のディストリビューションで導入された StorageGRID コンテナで接続の問題が発生しました。

タスクの内容

グリッドノードは、グリッドネットワークにアクセスする必要があります。また、管理ネットワークとクライアントネットワークにアクセスすることもできます。このアクセスを確立するには、ホストの物理インターフェイスを各グリッドノードの仮想インターフェイスに関連付けるマッピングを作成します。ホストインターフェイスを作成するときにわかりやすい名前を使用すると、すべてのホストへの導入が簡単になり、移行も可能になります。

ホストと1つ以上のノードで、同じインターフェイスを共有できます。たとえば、ホストアクセス用とノード管理ネットワークアクセス用のインターフェイスに同じものを使用すると、ホストとノードをメンテナンスしやすくなります。ホストと個々のノードで同じインターフェイスを共有できますが、IP アドレスはすべて異なっている必要があります。IPアドレスは、ノード間、またはホストと任意のノード間で共有できません。

グリッドネットワークのインターフェイスについては、ホストのすべての StorageGRID ノードで同じホストネットワークインターフェイスを使用したり、ノードごとに異なるホストネットワークインターフェイスを使用したり、任意のインターフェイスを使用したりできます。ただし、通常は、単一のホストのグリッドネットワークと管理ネットワークの両方のインターフェイス、またはいずれかのノードのグリッドネットワークのインターフェイスと別のホストのクライアントネットワークのインターフェイスに同じホストネットワークインターフェイスを使用することはありません。

このタスクはさまざまな方法で実行できます。たとえば、ホストが仮想マシンで、ホストごとに1つまたは2つのStorageGRID ノードを導入する場合は、ハイパーバイザーで正しい数のネットワークインターフェイスを作成し、1対1のマッピングを使用できます。本番環境用のベアメタルホストに複数のノードを導入する場合は、Linux ネットワークスタックの VLAN と LACP のサポートを利用してフォールトトレランスと帯域幅の共有を実現できます。以降のセクションでは、これら両方の例について詳細なアプローチを紹介します。これらのいずれかの例を使用する必要はありません。ニーズに合ったアプローチを使用できます。



ボンドデバイスやブリッジデバイスをコンテナネットワークインターフェイスとして直接使用しないでください。これにより、カーネル問題 が原因で発生するノードの起動が妨げられ、コンテナ名前空間内のボンドデバイスおよびブリッジデバイスで MACVLAN が使用される可能性があります。代わりに、VLAN ペアや仮想イーサネット（veth）ペアなどの非ボンディングデバイスを使用してください。このデバイスをノード構成ファイルのネットワークインターフェイスとして指定してください。

#### MAC アドレスのクローニングに関する考慮事項と推奨事項

MAC アドレスのクローニングでは、コンテナでホストの MAC アドレスが使用され、ホストでは指定したアドレスまたはランダムに生成されたアドレスの MAC アドレスが使用されます。プロミスキャスモードのネットワーク設定を使用しないようにするには、MAC アドレスのクローニングを使用します。

#### MAC クローニングのイネーブル化

環境によっては、管理ネットワーク、グリッドネットワーク、およびクライアントネットワークに専用の仮想 NIC を使用できるため、MAC アドレスのクローニングによってセキュリティを強化できます。コンテナでホストの専用 NIC の MAC アドレスを使用すると、プロミスキャスモードのネットワーク設定を回避できます。



MAC アドレスクローニングは、仮想サーバ環境で使用するためのものであり、物理アプライアンスのすべての構成で正常に機能しない場合があります。



MAC クローニングのターゲットインターフェイスがビジー状態のためにノードを起動できない場合は、ノードを起動する前にリンクを「停止」に設定しなければならないことがあります。また、リンクが稼働しているときに仮想環境でネットワークインターフェイス上の MAC クローニングが実行されないことがあります。インターフェイスがビジーなためにノードで MAC アドレスの設定が失敗してノードが起動しなかった場合は、問題を修正する前にリンクを「停止」に設定することがあります。

MAC アドレスクローニングは、デフォルトでは無効になっており、ノード設定キーで設定する必要があります。StorageGRID をインストールするときに有効にする必要があります。

ネットワークごとに 1 つのキーがあります。

- ADMIN\_NETWORK\_TARGET\_TYPE\_INTERFACE\_CLONE\_MAC
- GRID\_NETWORK\_TARGET\_TYPE\_INTERFACE\_CLONE\_MAC
- CLIENT\_NETWORK\_TARGET\_TYPE\_INTERFACE\_CLONE\_MAC

キーを「true」に設定すると、コンテナでホストの NIC の MAC アドレスが使用されます。さらに、ホストは指定されたコンテナネットワークの MAC アドレスを使用します。デフォルトでは、コンテナアドレスはランダムに生成されたアドレスですが、ノード構成キーを使用して設定した場合は`\_NETWORK\_MAC`そのアドレスが代わりに使用されます。ホストとコンテナの MAC アドレスは常に異なります。



ハイパーバイザーでプロミスキュースモードも有効にせずに仮想ホストの MAC クローニングを有効にすると、ホストのインターフェイスを使用して原因 Linux ホストのネットワークが停止する可能性があります。

## MAC クローン作成の使用例

MAC クローニングでは、次の 2 つのユースケースを検討します。

- MACクローニングが有効になっていない：ノード構成ファイルのキーが設定されていない場合、または「false」に設定されている場合 `_CLONE_MAC`、ホストはホストNIC MACを使用し、キーでMACが指定されていないかぎり、コンテナはStorageGRIDによって生成されたMACを持ち ``_NETWORK_MAC`` ます。キーにアドレスが設定されている場合、``_NETWORK_MAC`` コンテナはキーで指定されたアドレスを持ち ``_NETWORK_MAC`` ます。このキーの設定では、プロミスキュースモードを使用する必要があります。
- MACクローニングが有効：ノード構成ファイルのキーが「true」に設定されている場合、``_CLONE_MAC`` コンテナはホストNICのMACを使用し、キーでMACが指定されていないかぎり、ホストはStorageGRIDで生成されたMACを使用し ``_NETWORK_MAC`` ます。キーにアドレスが設定されている場合、``_NETWORK_MAC`` ホストは生成されたアドレスではなく、指定されたアドレスを使用します。このキーの設定では、プロミスキュースモードは使用しないでください。



MACアドレスクローニングを使用せず、ハイパーバイザーによって割り当てられたMACアドレス以外のMACアドレスのデータをすべてのインターフェイスで送受信できるようにする場合は、[Promiscuous Mode]、[MAC Address Changes]、および[Forged Transmits]で、仮想スイッチおよびポートグループレベルのセキュリティプロパティが[Accept]に設定されていることを確認します。仮想スイッチに設定された値は、ポートグループレベルの値によって上書きできるため、両方のレベルで設定が同じであることを確認してください。

MACクローニングをイネーブルにするには、を参照してください"[ノード構成ファイルの作成手順](#)"。

## MAC クローニングの例

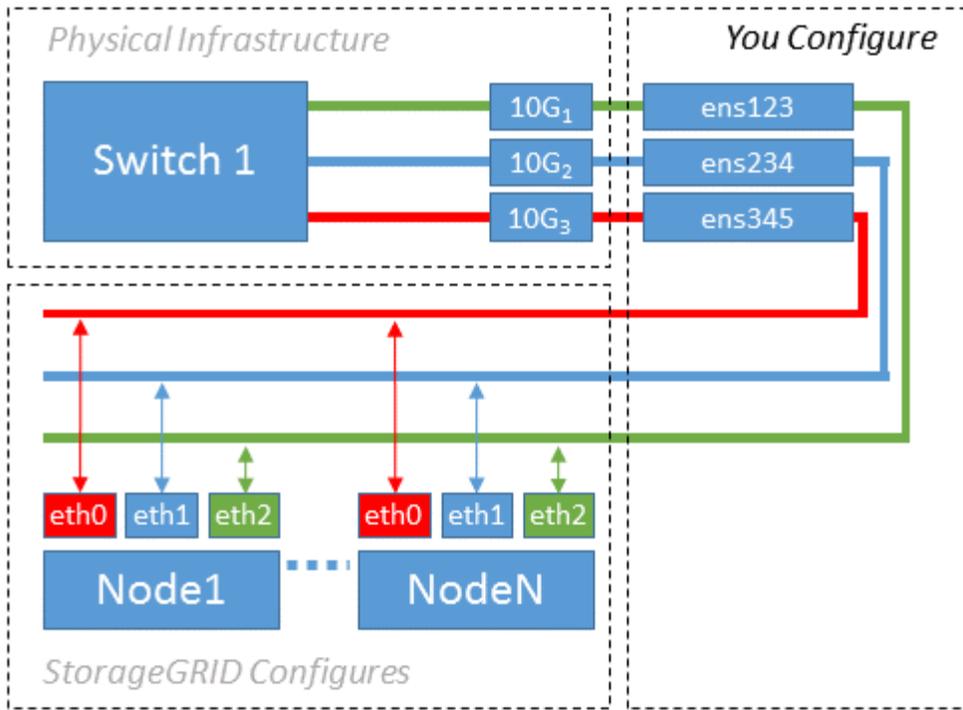
インターフェイスens256およびノード構成ファイルの次のキーに対して、MACアドレス11：22：33：44：55：66のホストでMACクローニングを有効にする例。

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

結果：ens256のホストMACはb2：9c：02：c2：27：10、管理ネットワークMACは11：22：33：44：55：66です。

### 例 1：物理 NIC または仮想 NIC への 1 対 1 のマッピング

例 1 では、ホスト側の設定がほとんどまたはまったく必要ない単純な物理インターフェイスのマッピングについて説明します。



Linuxオペレーティングシステムは、`ensXYZ`インストール時または起動時、あるいはインターフェースがホットアドされたときに、インターフェースを自動的に追加します。起動後にインターフェースが自動的に起動するように設定されていることを確認する以外に、構成は必要ありません。後の構成プロセスで正しいマッピングを提供できるように、どの `ensXYZ` がどのStorageGRIDネットワーク (グリッド、管理、またはクライアント) に対応するかを決定する必要があります。

この図は複数の StorageGRID ノードを示していますが、通常はこの構成をシングルノードの VM に使用します。

スイッチ 1 が物理スイッチの場合は、インターフェイス 10G1 ~ 10G3 に接続されたポートをアクセスモードに設定し、適切な VLAN に配置する必要があります。

## 例 2 : LACP ボンドを使用した VLAN の伝送

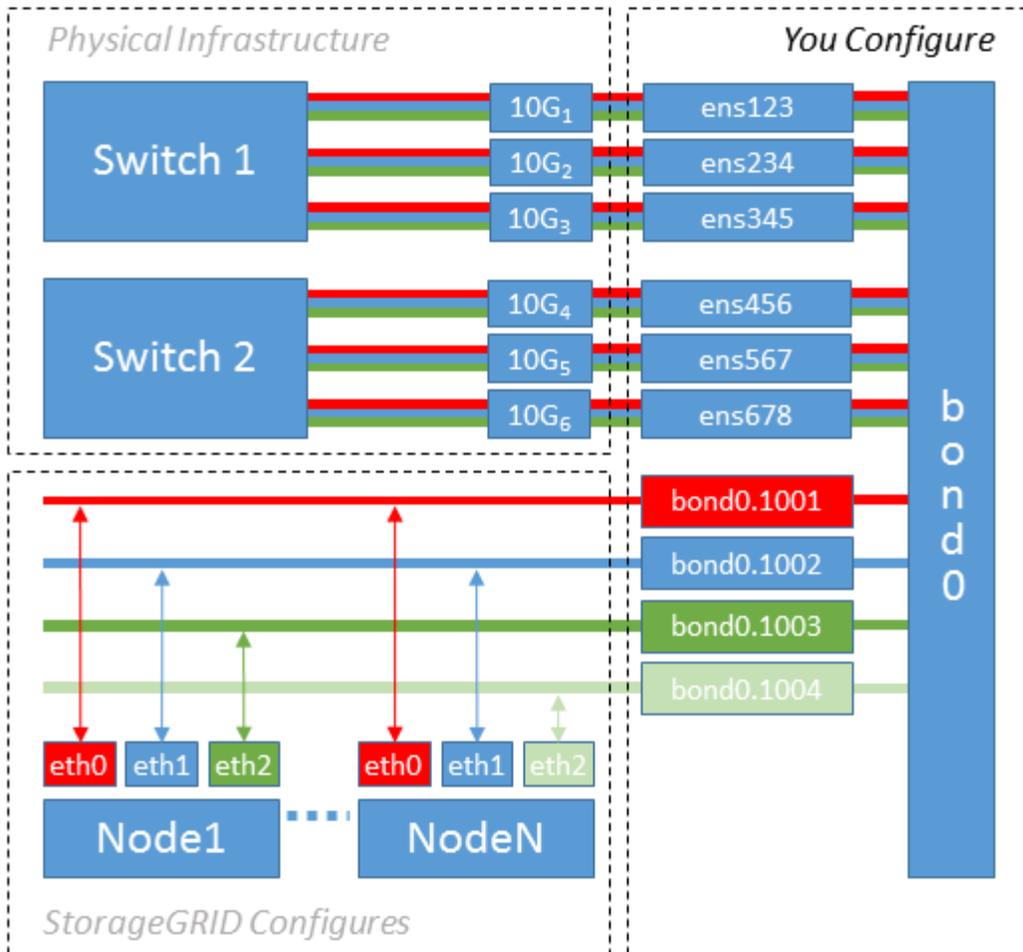
例 2 は、ネットワークインターフェースのボンディングおよび使用している Linux ディストリビューションでの VLAN インターフェースの作成に関する十分な知識があることを前提としています。

### タスクの内容

例 2 では、汎用の柔軟な VLAN ベースのスキームを使用して、使用可能なすべてのネットワーク帯域幅を単一のホスト上のすべてのノードで共有する方法について説明します。この例は、ベアメタルホストに特に該当します。

この例を理解するために、各データセンターにグリッドネットワーク、管理ネットワーク、クライアントネットワーク用に 3 つのサブネットワークがあるとします。サブネットワークは個別の VLAN ( 1001、1002、1003 ) 上にあり、LACP ボンディングされたトランクポート ( bond0 ) でホストに提示されます。この場合、ボンドに bond0.1001、bond0.1002、および bond0.1003 の 3 つの VLAN インターフェースを設定します。

同じホスト上のノードネットワークに別々の VLAN とサブネットワークが必要な場合は、ボンドに VLAN インターフェースを追加してホストにマッピングできます ( 図の bond0.1004 と表示 ) 。



## 手順

1. StorageGRID ネットワークの接続に使用するすべての物理ネットワークインターフェイスを単一の LACP ボンドとしてまとめます。

すべてのホストのボンドに同じ名前を使用します。例: bond0。

2. このボンドを関連する「物理デバイス」として使用するVLANインターフェイスを、標準のVLANインターフェイスの命名規則に従って作成します `physdev-name.VLAN ID`。

手順 1 と 2 のそれぞれについて、ネットワークリンクの反対側の終端にあるエッジスイッチで適切な設定を行う必要があります。エッジスイッチのポートも LACP ポートチャンネルに集約してトランクとして設定し、必要なすべての VLAN を許可する必要があります。

このホストごとのネットワーク構成スキームに使用できるサンプルのインターフェイス構成ファイルが提供されています。

## 関連情報

- ["UbuntuとDebianの/etc/network/interfacesの例"](#)
- ["RHEL の /etc/sysconfig/network-scripts の例"](#)

## Linux デプロイメント用の StorageGRID ホストストレージを設定

各 Linux ホストにブロック ストレージ ボリュームを割り当てる必要があります。

開始する前に

以下のトピックで、このタスクを実行するために必要な情報を確認しておきます。

- ["ストレージとパフォーマンスの要件"](#)
- ["ノードコンテナの移行要件"](#)



「Linux」は、RHEL、Ubuntu、または Debian のデプロイメントを指します。サポートされているバージョンのリストについては、["NetApp Interoperability Matrix Tool \(IMT\)"](#)。

タスクの内容

ブロックストレージボリューム (LUN) をホストに割り当てるときは、「ストレージ要件」の表を使用して次の項目を確認してください。

- 各ホストに必要なボリュームの数 (そのホストに導入するノードの数とタイプに応じて異なる)
- 各ボリュームのストレージのカテゴリ (システムデータまたはオブジェクトデータ)
- 各ボリュームのサイズ

ホストに StorageGRID ノードを導入するときは、この情報に加え、各物理ボリュームに Linux から割り当てられた永続的な名前を使用します。



これらのボリュームをパーティショニング、フォーマット、マウントする必要はありません。ボリュームがホストから認識できることを確認するだけで済みます。



メタデータ専用ストレージノードに必要なオブジェクトデータLUNは1つだけです。

(`/dev/sdb`` ボリューム名のリストを作成するときは、「raw」の特殊なデバイスファイルなどは使用しないでください。これらのファイルはホストのリブート時に変わることがあり、システムの適切な運用に影響します。iSCSI LUNとDevice Mapper Multipathingを使用している場合は、ディレクトリでマルチパスエイリアスを使用することを検討して ``/dev/mapper`` ください。特に、SANトポロジに共有ストレージへの冗長ネットワークパスが含まれている場合は、この方法が有効です。または、システムによって作成されたソフトリンクを永続的なデバイス名に使用することもできます ``/dev/disk/by-path/``。

例：

```
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
```

結果はインストールごとに異なります。

これらのブロックストレージボリュームのそれぞれにわかりやすい名前を割り当てると、StorageGRID の最初のインストールや以降のメンテナンスの手順が簡単になります。共有ストレージボリュームへのアクセスを冗長化するためにデバイスマッパーマルチパスドライバを使用している場合は、ファイルのフィールドを `/etc/multipath.conf`使用できます`alias。`

例：

```

multipaths {
    multipath {
        wwid 3600a09800059d6df00005df2573c2c30
        alias docker-storage-volume-hostA
    }
    multipath {
        wwid 3600a09800059d6df00005df3573c2c30
        alias sgws-adml-var-local
    }
    multipath {
        wwid 3600a09800059d6df00005df4573c2c30
        alias sgws-adml-audit-logs
    }
    multipath {
        wwid 3600a09800059d6df00005df5573c2c30
        alias sgws-adml-tables
    }
    multipath {
        wwid 3600a09800059d6df00005df6573c2c30
        alias sgws-gw1-var-local
    }
    multipath {
        wwid 3600a09800059d6df00005df7573c2c30
        alias sgws-sn1-var-local
    }
    multipath {
        wwid 3600a09800059d6df00005df7573c2c30
        alias sgws-sn1-rangedb-0
    }
    ...
}

```

aliasフィールドをこのように使用すると、ホストのディレクトリにブロックデバイスとしてエイリアスが表示される`/dev/mapper`のため、設定やメンテナンスの処理でブロックストレージボリュームを指定する必要があるときに、わかりやすい名前を指定して簡単に検証できます。

StorageGRIDノードの移行とDevice Mapperマルチパスの使用をサポートするために共有ストレージをセットアップする場合は、同じ場所にあるすべてのホストに共通のを作成してインストールできます`/etc/multipath.conf`。各ホストで別のコンテナエンジンのストレージボリュームを使用するだけで済みます。エイリアスを使用し、各コンテナエンジンのストレージボリュームのLUNのエイリアスにターゲットのホスト名を含めると覚えやすいので、この方法で設定することを推奨します。



ソフトウェアのみの環境のコンテナエンジンとしてのDockerのサポートは廃止されました。Dockerは、今後のリリースで別のコンテナエンジンに置き換えられる予定です。

#### 関連情報

- ["コンテナエンジンのストレージボリュームを設定します"](#)

- "ストレージとパフォーマンスの要件"
- "ノードコンテナの移行要件"

Linux上のStorageGRID用にコンテナエンジンのストレージボリュームを設定する

Docker または Podman コンテナ エンジンをインストールする前に、ストレージ ボリュームをフォーマットしてマウントする必要がある場合があります。



ソフトウェアのみの環境のコンテナエンジンとしてのDockerのサポートは廃止されました。Dockerは、今後のリリースで別のコンテナエンジンに置き換えられる予定です。



「Linux」は、RHEL、Ubuntu、または Debian のデプロイメントを指します。サポートされているバージョンのリストについては、"[NetApp Interoperability Matrix Tool \(IMT\)](#)"。

タスクの内容

Docker または Podman ストレージ ボリュームにルート ボリュームを使用する予定で、次のものを含むホストパーティションに十分な空き容量がある場合は、これらの手順をスキップできます。

- ポッドマン: /var/lib/containers
- ドッカー: /var/lib/docker

手順

1. コンテナエンジンのストレージボリュームにファイルシステムを作成します。

#### RHEL

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

#### Ubuntu または Debian

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

2. コンテナエンジンのストレージボリュームをマウントします。

## RHEL

- Docker の場合 :

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

- Podman の場合 :

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

## Ubuntu または Debian

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

- Podman の場合 :

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

### 3. コンテナ ストレージ ボリューム デバイスのエントリを /etc/fstab に追加します。

- RHEL: コンテナストレージボリュームデバイス
- Ubuntu または Debian: docker-storage-volume-device

これにより、ホストのリブート後にストレージボリュームが自動的に再マウントされます。

## Docker をインストールする

StorageGRIDシステムは、コンテナのコレクションとして Linux 上で実行できます。

- Ubuntu または Debian 用のStorageGRIDをインストールする前に、Docker をインストールする必要があります。
- Docker コンテナ エンジンを使用することを選択した場合は、次の手順に従って Docker をインストールします。さもないと、[Podman をインストールします](#)。



ソフトウェアのみの環境のコンテナエンジンとしてのDockerのサポートは廃止されました。Dockerは、今後のリリースで別のコンテナエンジンに置き換えられる予定です。

手順

1. 使用している Linux ディストリビューションの手順に従って Docker をインストールします。



Docker が Linux ディストリビューションに含まれていない場合は、Docker の Web サイトからダウンロードできます。

2. 次の 2 つのコマンドを実行して、Docker が有効化され、起動されたことを確認します。

```
sudo systemctl enable docker
```

```
sudo systemctl start docker
```

3. 次のコマンドを入力して、必要なバージョンの Docker がインストールされたことを確認します。

```
sudo docker version
```

クライアントとサーバのバージョンは1.11.0以降である必要があります。

#### Podman をインストールします

StorageGRIDシステムはコンテナのコレクションとして実行されます。Podman コンテナ エンジンを使用することを選択した場合は、次の手順に従って Podman をインストールします。さもないと、[Docker をインストールする](#)。

#### 手順

1. 使用している Linux ディストリビューションの手順に従って、Podman および Podman-Docker をインストールします。



また、Podman をインストールする際には、Podman-Docker パッケージもインストールする必要があります。

2. 次のように入力して、必要なバージョンの Podman および Podman-Docker がインストールされていることを確認します。

```
sudo docker version
```



Podman-Docker パッケージでは、Docker コマンドを使用できます。

クライアントとサーバのバージョンは3.2.3以降である必要があります。

```
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
```

## 関連情報

["ホストストレージを設定する"](#)

## LinuxへのStorageGRIDホストサービスのインストール

StorageGRIDホスト サービスをインストールするには、オペレーティング システムの種類に応じたStorageGRIDパッケージを使用します。



「Linux」は、RHEL、Ubuntu、または Debian のデプロイメントを指します。サポートされているバージョンのリストについては、"[NetApp Interoperability Matrix Tool \(IMT\)](#)"。

## RHEL

StorageGRID ホストサービスをインストールするには、StorageGRID RPM パッケージを使用します。

### タスクの内容

以下の手順では、RPM パッケージからホストサービスをインストールする方法について説明します。また、インストールアーカイブに含まれている DNF リポジトリメタデータを使用して、RPM パッケージをリモートでインストールすることもできます。使用している Linux オペレーティングシステムの DNF リポジトリに関する手順を参照してください。

### 手順

1. 各ホストに StorageGRID RPM パッケージをコピーするか、共有ストレージに置きます。

たとえば、次の手順のコマンド例を使用できるように、これらのコマンドをディレクトリに配置し `tmp` ます。

2. 各ホストに root アカウントまたは sudo 権限を持つアカウントでログインし、次のコマンドをこの順序で実行します。

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

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



まずイメージパッケージをインストールし、次にサービスパッケージをインストールする必要があります。



パッケージを以外のディレクトリに配置した `tmp` 場合は、使用したパスを反映するようにコマンドを変更します。

## Ubuntu または Debian

StorageGRID DEB パッケージを使用して、Ubuntu または Debian 用の StorageGRID ホスト サービスをインストールします。

### タスクの内容

以下の手順では、DEB パッケージからホストサービスをインストールする方法について説明します。また、インストールアーカイブに含まれている APT リポジトリメタデータを使用して、DEB パッケージをリモートでインストールすることもできます。使用している Linux オペレーティングシステムの APT リポジトリに関する手順を参照してください。

### 手順

1. 各ホストに StorageGRID DEB パッケージをコピーするか、共有ストレージに置きます。

たとえば、次の手順のコマンド例を使用できるように、これらのコマンドをディレクトリに配置し `tmp` ます。

2. 各ホストに root アカウントまたは sudo 権限を持つアカウントでログインし、次のコマンドを実行します。

最初にパッケージをインストールし、`service`次にパッケージをインストールする必要があります。`images`ます。パッケージを以外のディレクトリに配置した`/tmp`場合は、使用したパスを反映するようにコマンドを変更します。

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

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



StorageGRIDパッケージをインストールする前に、Python 3 がインストールされている必要があります。その`sudo dpkg --install /tmp/storagegrid-webscale-images-version-SHA.deb`これを行わない限り、コマンドは失敗します。

## ソフトウェアベースのノードのインストールを自動化する

### Linux上のStorageGRIDホストサービスのインストールとグリッドノードの構成を自動化

StorageGRID ホストサービスのインストールおよびグリッドノードの設定を自動化することができます。



#### タスクの内容

「Linux」は、RHEL、Ubuntu、または Debian のデプロイメントを指します。サポートされているバージョンのリストについては、"[NetApp Interoperability Matrix Tool \(IMT\)](#)"。

導入を自動化すると、次のいずれかの場合に役立ちます。

- 物理ホストや仮想ホストの導入と設定に Ansible、Puppet、Chef などの標準のオーケストレーションフレームワークをすでに使用している場合。
- 複数の StorageGRID インスタンスを導入する場合。
- 大規模で複雑な StorageGRID インスタンスを導入する場合。

StorageGRID ホストサービスはパッケージによってインストールされ、構成ファイルによって制御されます。次のいずれかの方法で構成ファイルを作成できます。

- "[構成ファイルを作成します](#)"手動インストール中に対話的に実行します。
- 構成ファイルを事前に準備し（またはプログラム化して）、この資料で説明するように、標準のオーケストレーションフレームワークを使用した自動インストールを可能にします。

StorageGRIDには、StorageGRIDアプライアンスとStorageGRIDシステム全体（「グリッド」）の設定を

自動化するためのPythonスクリプトがオプションで用意されています。これらのスクリプトを直接使用することも、スクリプトを調べて、独自に開発したグリッド内導入ツールや設定ツールの使用方法を学ぶこともでき["StorageGRID インストール REST API"](#)ます。

## StorageGRID ホストサービスのインストールと設定を自動化する

StorageGRID ホストサービスのインストールは、Ansible、Puppet、Chef、Fabric、SaltStack などの標準のオーケストレーションフレームワークを使用して自動化できます。

StorageGRIDホスト サービスは、DEB (Ubuntu または Debian) または RPM (RHEL) にパッケージ化されており、自動インストールを有効にするために事前に (またはプログラムによって) 準備できる構成ファイルによって駆動されます。すでに標準のオーケストレーション フレームワークを使用して Linux デプロイメントをインストールおよび構成している場合は、プレイブックまたはレシピにStorageGRIDを追加するのは簡単です。

ホストの準備と仮想グリッドノードの導入の手順をすべて自動化することができます。

サンプルの **Ansible** のロールとプレイブック

サンプルのAnsibleのロールとプレイブックは、インストールアーカイブのフォルダにあります `/extras`。Ansibleプレイブックは、ロールでホストを準備してStorageGRIDをターゲットサーバにインストールする方法を示しています `storagegrid`。必要に応じて、ロールまたはプレイブックをカスタマイズできます。



サンプルのプレイブックには、StorageGRID ホストサービスを開始する前にネットワークデバイスを作成するために必要な手順は含まれていません。これらの手順は、最終的な確認と使用の前に追加してください。

### RHEL

RHELの場合、提供されているインストールタスクは `storagegrid` 役割の例を使用する `ansible.builtin.dnf` ローカル RPM ファイルまたはリモート Yum リポジトリからインストールを実行するモジュール。モジュールが利用できないかサポートされていない場合は、次のファイル内の適切なAnsibleタスクを編集して、`yum` または `ansible.builtin.yum` モジュール:

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

### Ubuntu または Debian

UbuntuまたはDebianの場合、提供されているインストールタスクは `storagegrid` 役割の例を使用する `ansible.builtin.apt` ローカル DEB ファイルまたはリモート apt リポジトリからインストールを実行するモジュール。モジュールが利用できないかサポートされていない場合は、次のファイル内の適切なAnsibleタスクを編集して、`ansible.builtin.apt` モジュール:

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

## StorageGRID の設定を自動化

グリッドノードを導入したら、StorageGRID システムの設定を自動化できます。

## 開始する前に

- インストールアーカイブにある次のファイルの場所を確認しておきます。

ファイル名	製品説明
configure-storagegrid.py	設定を自動化するための Python スクリプト
storagegrid-sample.json を設定します	スクリプトで使用する構成ファイルの例
storagegrid-bank.json を設定する	スクリプトで使用する空の構成ファイルです

- 構成ファイルを作成しておき `configure-storagegrid.json` ます。このファイルを作成するには (`configure-storagegrid.sample.json`、サンプル構成ファイル) または空の構成ファイル (`configure-storagegrid.blank.json` を変更します)。



変更したファイルのパスワードセクションから管理パスワードとプロビジョニングパスワードを保存します。`configure-storagegrid.json` 構成ファイルを安全な場所に保管します。これらのパスワードは、インストール、拡張、およびメンテナンスの手順に必要です。変更したファイルもバックアップする必要があります `configure-storagegrid.json` 構成ファイルを作成して安全な場所に保存します。

## タスクの内容

Python スクリプトと `configure-storagegrid.json` 構成ファイルを使用して、StorageGRID システムの設定を自動化できます `configure-storagegrid.py`。



また、Grid Manager またはインストール API を使用してシステムを設定することもできます。

## 手順

1. Python スクリプトを実行するために使用する Linux マシンにログインします。
2. インストールアーカイブを展開したディレクトリに移動します。

例：

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

```
`platform` は `rpms`、または `vsphere` です `debs`。
```

3. Python スクリプトを実行し、作成した構成ファイルを使用します。

例：

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

## 結果

設定プロセス中にリカバリパッケージ`.zip`ファイルが生成され、インストールおよび設定プロセスを実行するディレクトリにダウンロードされます。グリッドノードで障害が発生した場合に StorageGRID システムをリカバリできるようにするために、リカバリパッケージファイルをバックアップする必要があります。たとえば、バックアップされたセキュアなネットワーク上の場所や、安全なクラウドストレージ上の場所にコピーします。



リカバリパッケージファイルには StorageGRID システムからデータを取得するための暗号キーとパスワードが含まれているため、安全に保管する必要があります。

ランダムパスワードを生成するように指定した場合は、ファイルを開き Passwords.txt、StorageGRID システムへのアクセスに必要なパスワードを探します。

```
#####  
##### 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.           #####  
#####
```

StorageGRID システムがインストールおよび設定されると、確認メッセージが表示されます。

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

## StorageGRIDでVMware vSphereを使用してグリッドノードの導入を自動化

VMware OVF Toolを使用すると、グリッドノードの導入を自動化できます。StorageGRID の設定を自動化することもできます。

### グリッドノードの導入を自動化

VMware OVF Toolを使用すると、グリッドノードの導入を自動化できます。

#### 開始する前に

- Bash 3.2 以降が搭載された Linux / UNIX システムにアクセスできるようにしておきます。
- VMware vSphereとvCenterを使用している場合
- VMware OVF ツールがインストールされ、正しく構成されています。
- OVFツールを使用してVMware vSphereにアクセスするためのユーザ名とパスワードを確認しておきます。
- OVFファイルからVMを導入して電源をオンにするための十分な権限と、VMに接続するための追加ポリシーを作成するための権限が必要です。詳細については、のドキュメントを参照してください  
ovftool。
- StorageGRID 仮想マシンを導入する vSphere 内の場所の仮想インフラ (VI) URL を確認しておきます。この URL は通常、vApp またはリソースプールです。例：

vi://vcenter.example.com/vi/sgws



この値は、VMwareユーティリティを使用して確認でき `ovftool``ます（詳細についてはのドキュメントを参照してください ``ovftool`）。



vApp に導入する場合、初回は仮想マシンが自動的に起動されないため、手動で電源をオンにする必要があります。

- 導入構成ファイルに必要なすべての情報を収集しておきます。詳細は、を参照してください"[導入環境に関する情報を収集します](#)"。
- VMware 用インストールアーカイブに含まれている次のファイルに StorageGRID からアクセスできるようにしておきます。

ファイル名	製品説明
NetApp-SG-version-sha.vmdk	グリッドノード仮想マシンを作成するためのテンプレートとして使用される仮想マシンディスクファイル。  *注：*このファイルは、ファイルおよび <code>.mf`</code> ファイルと同じフォルダにある必要があります <code>`ovf</code> 。
vsphere-primary-admin.ovf vsphere-primary-admin.mf	( <code>.mf`</code> プライマリ管理ノードを導入するためのOpen Virtualization Formatテンプレートファイル) ( <code>.ovf</code> とマニフェストファイル)
vsphere-non-primary-admin.ovf vsphere-non-primary-admin.mf	テンプレートファイル( <code>.ovf</code> ) とマニフェストファイル( <code>.mf</code> )。非プライマリ管理ノードを導入するためのものです。
vsphere-gateway.ovf vsphere-gateway.mf	テンプレートファイル( <code>.ovf</code> ) とマニフェストファイル( <code>.mf</code> ) を使用してゲートウェイノードを導入します。
vsphere-storage.ovf vsphere-storage.mf	( <code>.mf`</code> 仮想マシンベースのストレージノードを導入するためのテンプレートファイル ( <code>.ovf</code> とマニフェストファイル))
deploy-vsphere-ovftool.sh	仮想グリッドノードの導入を自動化するための Bash シェルスクリプト。
deploy-vsphere-ovftool-sample.ini	スクリプトで使用する構成ファイルの例を示します <code>deploy-vsphere-ovftool.sh</code> 。

導入環境に応じた構成ファイルを定義します

StorageGRIDの仮想グリッドノードを導入するために必要な情報を構成ファイルで指定します。このファイルは、Bashスクリプトで使用され ``deploy-vsphere-ovftool.sh``ます。サンプル構成ファイルを変更して、ファイ

ルを最初から作成する必要がないようにすることができます。

## 手順

1. サンプルコンフィギュレーションファイルのコピーを作成し(`deploy-vsphere-ovftool.sample.ini`です)。新しいファイルをとこの名前で同じディレクトリに `deploy-vsphere-ovftool.sh` 保存します `deploy-vsphere-ovftool.ini`。
2. `deploy-vsphere-ovftool.ini`を開きます。
3. VMware 仮想グリッドノードを導入するために必要なすべての情報を入力します。

詳細は、を参照してください[構成ファイルの設定](#)。

4. 必要な情報をすべて入力して確認したら、ファイルを保存して閉じます。

## 構成ファイルの設定

```
`deploy-vsphere-ovftool.ini`構成ファイルには、仮想グリッドノードの導入に必要な設定が含まれています。
```

構成ファイルでは、最初にグローバルパラメータがリストされ、そのあとにノード名で定義されるセクションにノード固有のパラメータがリストされます。ファイルの使用状況：

- *Global parameters* は、すべてのグリッドノードに適用されます。
- `_Node-specific parameters_override` グローバルパラメータ。

## グローバルパラメータ

グローバルパラメータは、個々のセクションの設定で上書きされないかぎり、すべてのグリッドノードに適用されます。複数のノードに適用するパラメータをグローバルパラメータセクションに配置し、個々のノードのセクションで必要に応じてこれらの設定を上書きします。

- `* OVFTOOL_ARGUMENTS *` : `OVFTOOL_ARGUMENTS` をグローバル設定として指定するか、または特定のノードに個別に引数を適用できます。例：

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

オプションと `--overwrite` オプションを使用して、既存の仮想マシンをシャットダウンして交換できます `--powerOffTarget`。



ノードを別々のデータストアに導入し、`OVFTOOL_ARGUMENTS` をグローバルに指定するのではなくノードごとに指定する必要があります。

- `* source *` : StorageGRID仮想マシンテンプレート(`.vmdk`ファイルと個々のグリッドノードのファイル`.ovf`と`.mf`ファイルのパス。デフォルトでは現在のディレクトリに設定されます。

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

- \* target \* : StorageGRID の導入先となる VMware vSphere 仮想インフラ (vi) の URL。例:

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

- \* GRID\_NETWORK\_CONFIG \* : 静的または DHCP のいずれかの IP アドレスの取得に使用される方法。デフォルトは STATIC です。全ノードまたはほとんどのノードが IP アドレスの取得に同じ方法を使用する場合は、ここでその方法を指定できます。その後、個々のノードで別々の設定を指定してグローバル設定を上書きできます。例:

```
GRID_NETWORK_CONFIG = STATIC
```

- \* GRID\_NETWORK\_TARGET \* : グリッドネットワークに使用される既存の VMware ネットワークの名前。全ノードまたはほとんどのノードが同じネットワーク名を使用する場合は、ここでその名前を指定できます。その後、個々のノードで別々の設定を指定してグローバル設定を上書きできます。例:

```
GRID_NETWORK_TARGET = SG Admin Network
```

- \* GRID\_NETWORK\_MASK \* : グリッドネットワークのネットワークマスク。全ノードまたはほとんどのノードが同じネットワークマスクを使用する場合は、ここでそのネットワークマスクを指定できます。その後、個々のノードで別々の設定を指定してグローバル設定を上書きできます。例:

```
GRID_NETWORK_MASK = 255.255.255.0
```

- \* GRID\_NETWORK\_GATEWAY \* : グリッドネットワークのネットワークゲートウェイ。全ノードまたはほとんどのノードが同じネットワークゲートウェイを使用する場合は、ここでそのネットワークゲートウェイを指定できます。その後、個々のノードで別々の設定を指定してグローバル設定を上書きできます。例:

```
GRID_NETWORK_GATEWAY = 10.1.0.1
```

- \* GRID\_NETWORK\_MTU \* : オプション。グリッドネットワークでの最大伝送ユニット (MTU) です。この値を指定する場合、1280 ~ 9216 の範囲で指定する必要があります。例:

```
GRID_NETWORK_MTU = 9000
```

省略すると、1400 が使用されます。

ジャンボフレームを使用する場合は、MTU を 9000 などのジャンボフレームに適した値に設定します。それ以外の場合は、デフォルト値のままにします。



ネットワークのMTU値は、ノードの接続先であるvSphereの仮想スイッチポートに設定されている値と同じである必要があります。そうしないと、ネットワークパフォーマンスの問題やパケット損失が発生する可能性があります。



ネットワークのパフォーマンスを最大限に高めるには、すべてのノードのグリッドネットワークインターフェイスで MTU 値がほぼ同じになるように設定する必要があります。個々のノードのグリッドネットワークの MTU 設定に大きな違いがある場合は、\* Grid Network MTU mismatch \* アラートがトリガーされます。MTU値はすべてのネットワークタイプで同じである必要はありません。

- \* ADMIN\_NETWORK\_CONFIG \* : IP アドレスの取得に使用された方法。無効、静的、または DHCP のいずれかです。デフォルトはdisabledです。全ノードまたはほとんどのノードが IP アドレスの取得に同じ方法を使用する場合は、ここでその方法を指定できます。その後、個々のノードで別々の設定を指定してグローバル設定を上書きできます。例：

```
ADMIN_NETWORK_CONFIG = STATIC
```

- \* ADMIN\_NETWORK\_TARGET \* : 管理ネットワークに使用する既存の VMware ネットワークの名前。この設定は、管理ネットワークが無効になっていない場合に必要となります。全ノードまたはほとんどのノードが同じネットワーク名を使用する場合は、ここでその名前を指定できます。グリッドネットワークとは異なり、すべてのノードを同じ管理ネットワークに接続する必要はありません。その後、個々のノードで別々の設定を指定してグローバル設定を上書きできます。例：

```
ADMIN_NETWORK_TARGET = SG Admin Network
```

- \* ADMIN\_NETWORK\_MASK \* : 管理ネットワークのネットワークマスク。この設定は、静的 IP アドレスを使用する場合に必要となります。全ノードまたはほとんどのノードが同じネットワークマスクを使用する場合は、ここでそのネットワークマスクを指定できます。その後、個々のノードで別々の設定を指定してグローバル設定を上書きできます。例：

```
ADMIN_NETWORK_MASK = 255.255.255.0
```

- \* ADMIN\_NETWORK\_GATEWAY \* : 管理ネットワークのネットワークゲートウェイ。この設定は、IP アドレスを静的に指定し、かつ ADMIN\_NETWORK\_ESL 設定で外部サブネットを指定する場合に必要となります（つまり、ADMIN\_NETWORK\_ESLが空の場合は必要ありません）。全ノードまたはほとんどのノードが同じネットワークゲートウェイを使用する場合は、ここでそのネットワークゲートウェイを指定できます。その後、個々のノードで別々の設定を指定してグローバル設定を上書きできます。例：

```
ADMIN_NETWORK_GATEWAY = 10.3.0.1
```

- \* ADMIN\_NETWORK\_ESL \* : 管理ネットワークの外部サブネットリスト（ルート）。CIDR ルートのデスティネーションをカンマで区切ったリストとして指定します。全ノードまたはほとんどのノードが同じ外部サブネットリストを使用する場合は、ここでそのリストを指定できます。その後、個々のノードで別々の設定を指定してグローバル設定を上書きできます。例：

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

- \* ADMIN\_NETWORK\_MTU \* : オプション。管理ネットワークでの最大伝送ユニット (MTU) です。ADMIN\_NETWORK\_CONFIG = DHCPの場合は指定しないでください。この値を指定する場合、1280 ~ 9216 の範囲で指定する必要があります。省略すると、1400 が使用されます。ジャンボフレームを使用する場合は、MTU を 9000 などのジャンボフレームに適した値に設定します。それ以外の場合は、デフォルト値のままにします。全ノードまたはほとんどのノードが管理ネットワークに同じ MTU を使用する場合は、ここでその MTU を指定できます。その後、個々のノードで別々の設定を指定してグローバル設定を上書きできます。例：

```
ADMIN_NETWORK_MTU = 8192
```

- \* CLIENT\_NETWORK\_CONFIG \* : IP アドレスの取得に使用する方法。無効、静的、または DHCP のいずれかになります。デフォルトはdisabledです。全ノードまたはほとんどのノードが IP アドレスの取得に同じ方法を使用する場合は、ここでその方法を指定できます。その後、個々のノードで別々の設定を指定してグローバル設定を上書きできます。例：

```
CLIENT_NETWORK_CONFIG = STATIC
```

- \* client\_network\_target \* : クライアントネットワークに使用する既存の VMware ネットワークの名前。この設定は、クライアントネットワークが無効になっていない場合に必要となります。全ノードまたはほとんどのノードが同じネットワーク名を使用する場合は、ここでその名前を指定できます。グリッドネットワークとは異なり、すべてのノードを同じクライアントネットワークに接続する必要はありません。その後、個々のノードで別々の設定を指定してグローバル設定を上書きできます。例：

```
CLIENT_NETWORK_TARGET = SG Client Network
```

- \* CLIENT\_NETWORK\_MASK \* : クライアントネットワークのネットワークマスク。この設定は、静的 IP アドレスを使用する場合に必要となります。全ノードまたはほとんどのノードが同じネットワークマスクを使用する場合は、ここでそのネットワークマスクを指定できます。その後、個々のノードで別々の設定を指定してグローバル設定を上書きできます。例：

```
CLIENT_NETWORK_MASK = 255.255.255.0
```

- \* client\_network\_gateway \* : クライアントネットワークのネットワークゲートウェイ。この設定は、静的 IP アドレスを使用する場合に必要となります。全ノードまたはほとんどのノードが同じネットワークゲートウェイを使用する場合は、ここでそのネットワークゲートウェイを指定できます。その後、個々のノードで別々の設定を指定してグローバル設定を上書きできます。例：

```
CLIENT_NETWORK_GATEWAY = 10.4.0.1
```

- \* CLIENT\_NETWORK\_MTU \* : オプション。クライアントネットワークでの最大伝送ユニット (MTU) です。CLIENT\_NETWORK\_CONFIG = DHCPの場合は指定しないでください。この値を指定する場合、

1280 ~ 9216 の範囲で指定する必要があります。省略すると、1400 が使用されます。ジャンボフレームを使用する場合は、MTU を 9000 などのジャンボフレームに適した値に設定します。それ以外の場合は、デフォルト値のままにします。全ノードまたはほとんどのノードがクライアントネットワークに同じ MTU を使用する場合は、ここでその MTU を指定できます。その後、個々のノードで別々の設定を指定してグローバル設定を上書きできます。例：

```
CLIENT_NETWORK_MTU = 8192
```

- \* PORT\_REMAP \* : ノードが内部でのグリッドノードの通信または外部との通信に使用するポートを再マッピングします。StorageGRID で使用される 1 つ以上のポートがエンタープライズネットワークポリシーによって制限される場合は、ポートの再マッピングが必要です。StorageGRID で使用されるポートのリストについては、の内部でのグリッドノードの通信と外部との通信を参照してください"[ネットワークのガイドライン](#)"。



ロードバランサエンドポイントの設定に使用する予定のポートは再マッピングしないでください。



PORT\_REMAP のみを設定すると、指定したマッピングがインバウンド通信とアウトバウンド通信の両方に使用されます。PORT\_REMAP\_INBOUND を併せて指定した場合は、PORT\_REMAP がアウトバウンド通信のみに適用されます。

使用される形式は、`network type/protocol/default port used by grid node/new port`。ネットワークタイプは `grid`、`admin`、または `client`、`protocol` は `tcp` または `udp` です。

例：

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

この例の設定だけを使用した場合は、グリッドノードのインバウンド通信とアウトバウンド通信の両方が、ポート 18082 からポート 443 へと対称的にマッピングされます。この例の設定を PORT\_REMAP\_INBOUND とともに使用した場合は、アウトバウンド通信がポート 18082 からポート 443 にマッピングされます。

カンマで区切ったリストを使用して複数のポートを再マッピングすることもできます。

例：

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

- \* port\_remap\_inbound \* : 指定したポートのインバウンド通信を再マッピングします。PORT\_REMAP\_INBOUND を指定し、PORT\_REMAP に値を指定しなかった場合、ポートのアウトバウンド通信は変更されません。



ロードバランサエンドポイントの設定に使用する予定のポートは再マッピングしないでください。

使用される形式は、です `network type/protocol/_default port used by grid node/new port`。ネットワークタイプはgrid、admin、またはclient、protocolはtcpまたはudpです。

例：

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

次の例は、ポート 443 に送信されたトラフィックを内部ファイアウォールを通過させ、グリッドノードが S3 要求をリスンしているポート 18082 に転送します。

カンマで区切った複数のインバウンドポートを再マッピングすることもできます。

例：

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

- `* temporary_password_type *`：ノードがグリッドに参加する前に、VMコンソールやStorageGRIDインストールAPIにアクセスする場合、またはSSHを使用してアクセスする場合に使用する一時インストールパスワードのタイプ。



すべてのノードまたはほとんどのノードで同じタイプの一時インストールパスワードを使用する場合は、グローバルパラメータセクションでタイプを指定します。その後、必要に応じて個々のノードに別の設定を使用します。たとえば、`[カスタムパスワードを使用]*`をグローバルに選択した場合は、`custom_temporary_password =<password>*`を使用して各ノードのパスワードを設定できます。

- `temporary_password_type *`には、次のいずれかを指定できます。
  - ノード名を使用：ノード名は一時的なインストールパスワードとして使用され、VMコンソール、StorageGRIDインストールAPI、およびSSHへのアクセスを提供します。
  - パスワードを無効にする：一時的なインストールパスワードは使用されません。インストールの問題をデバッグするためにVMにアクセスする必要がある場合は、[を参照してください"インストールに関する問題のトラブルシューティング"](#)。
  - カスタムパスワードを使用：`* custom_temporary_password =<password>*`で指定した値は、一時的なインストールパスワードとして使用され、VMコンソール、StorageGRIDインストールAPI、およびSSHへのアクセスを提供します。



必要に応じて、`* temporary_password_type` パラメータを省略し、`custom_temporary_password=<password>*`のみを指定できます。

- `* custom_temporary_password =<password>*`オプション。インストール時にVMコンソール、StorageGRIDインストールAPI、およびSSHにアクセスする際に使用する一時パスワード。TEMPORARY\_PASSWORD\_TYPE が Use node name または Disable password \*に設定されている場合は無視されます。

ノード固有のパラメータ

構成ファイルには、各ノード専用のセクションがあります。各ノードには次の設定が必要です。

- セクションヘッドでは、Grid Manager に表示されるノード名を定義します。この値を無視するには、ノードに対してオプションの `node_name` パラメータを指定します。
- \* `NODE_TYPE` \* : `VM_Admin_Node`、`VM_Storage_Node`、または `VM_API_Gateway_Node`
- \* `storage_type` \* : 組み合わせたデータ、またはメタデータ。(オプション) ストレージノードのこのパラメータは、データとメタデータの組み合わせが指定されていない場合はデフォルトで設定されます。詳細については、を参照してください "[ストレージノードのタイプ](#)"。
- \* `GRID_NETWORK_IP` \* : グリッドネットワークでのノードの IP アドレス。
- \* `ADMIN_NETWORK_IP` \* : 管理ネットワークでのノードの IP アドレス。ノードが管理ネットワークに接続され、かつ `ADMIN_NETWORK_CONFIG` が `STATIC` に設定されている場合にのみ必要です。
- \* `client_network_ip` \* : クライアントネットワーク上のノードの IP アドレス。ノードがクライアントネットワークに接続され、かつノードの `CLIENT_NETWORK_CONFIG` が `STATIC` に設定されている場合にのみ必要です。
- \* `ADMIN_IP` \* : グリッドネットワークでのプライマリ管理ノードの IP アドレス。プライマリ管理ノードの `GRID_NETWORK_IP` で指定した値を使用します。このパラメータを省略すると、ノードは mDNS を使用してプライマリ管理ノードの IP を検出しようとします。詳細については、を参照してください "[グリッドノードによるプライマリ管理ノードの検出](#)"。



プライマリ管理ノードでは `ADMIN_IP` パラメータが無視されます。

- グローバルに設定されていないすべてのパラメータ。たとえば、ノードが管理ネットワークに接続されていて、`ADMIN_NETWORK_NETWORK` パラメータをグローバルに指定していない場合は、ノードに対してそれらのパラメータを指定する必要があります。

#### プライマリ管理ノード

プライマリ管理ノードには次の設定を追加する必要があります。

- \* `node_type` \* : `VM_Admin_Node`
- \* `Admin_role` \* : プライマリ

次のエントリ例は、プライマリ管理ノードが 3 つのネットワークすべてに接続される場合を示しています。

```
[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
```

プライマリ管理ノードにオプションで追加できる設定は次のとおりです。

- \* `DISK` \* : デフォルトでは、管理ノードに対して監査用とデータベース用の 2 つの 200GB ハードディスクが追加で割り当てられます。DISK パラメータを使用して、この容量を増やすことができます。例：

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



管理ノードの場合は、INSTANCES を必ず 2 にする必要があります。

ストレージノード

ストレージノードには次の設定を追加する必要があります。

- \* node\_name \* : VM\_Storage\_Node

次のエントリ例は、ストレージノードがグリッドネットワークと管理ネットワークに接続され、クライアントネットワークに接続されない場合を示しています。このノードでは、ADMIN\_IP 設定を使用してグリッドネットワークでのプライマリ管理ノードの IP アドレスを指定しています。

```
[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
```

2 番目のエントリ例は、ストレージノードがクライアントネットワークに接続される場合を示しています。ここでは、S3 クライアントアプリケーションがストレージノードへのアクセスに使用できるポートが、ユーザのエンタープライズネットワークポリシーによって 80 または 443 に制限されています。この例の構成ファイルでは、PORT\_REMAP を使用して、ストレージノードがポート 443 で S3 メッセージを送受信できるようにしています。

```
[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
```

最後の例では、ssh トラフィックに対してポート 22 からポート 3022 への対称的な再マッピングが作成されますが、インバウンドとアウトバウンドの両方のトラフィックに明示的に値が設定されます。

```
[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
```

ストレージノードにオプションで追加できる設定は次のとおりです。

- **\* DISK \*** : デフォルトでは、ストレージノードに対して RangeDB 用に 3 つの 4TB ディスクが割り当てられます。DISK パラメータを使用して、この容量を増やすことができます。例 :

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

- **\* storage\_type \*** : すべての新しいストレージノードは、オブジェクトデータとメタデータの両方を格納するようにデフォルトで設定されます ( `_combined_storage Node` )。storage\_type パラメータを使用して、データまたはメタデータのみを格納するようにストレージノードのタイプを変更できます。例 :

```
STORAGE_TYPE = data
```

#### ゲートウェイノード

ゲートウェイノードには次の設定を追加する必要があります。

- **\* node\_name \*** : VM\_API\_Gateway

次のエントリ例は、ゲートウェイノードが 3 つのネットワークすべてに接続される場合を示しています。この例では、構成ファイルのグローバルセクションでクライアントネットワークのパラメータが指定されていないため、ノードに対してそれらのパラメータを指定する必要があります。

```
[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
```

#### 非プライマリ管理ノード

非プライマリ管理ノードには次の設定を追加する必要があります。

- \* node\_type \* : VM\_Admin\_Node
- \* Admin\_role \* : 非プライマリ

次のエントリ例は、非プライマリ管理ノードがクライアントネットワークに接続されない場合を示しています。

```
[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
```

非プライマリ管理ノードにオプションで追加できる設定は次のとおりです。

- \* DISK \* : デフォルトでは、管理ノードに対して監査用とデータベース用の 2 つの 200GB ハードディスクが追加で割り当てられます。DISK パラメータを使用して、この容量を増やすことができます。例：

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



管理ノードの場合は、INSTANCES を必ず 2 にする必要があります。

## Bash スクリプトを実行します

VMware vSphereへのStorageGRIDノードの導入を自動化するために、Bashスクリプトと変更したdeploy-vsphere-ovftool.ini構成ファイルを使用できます deploy-vsphere-ovftool.sh。

開始する前に

環境に対応した deploy-vsphere-ovftool.ini 構成ファイルを作成しておきます。

Bashスクリプトのヘルプを使用するには、helpコマンドを入力し(`-h/--help`ます)。例：

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

または

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

手順

1. Bash スクリプトの実行に使用する Linux マシンにログインします。
2. インストールアーカイブを展開したディレクトリに移動します。

例：

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

3. グリッドノードをすべて導入する場合は、使用する環境に適したオプションを指定して Bash スクリプトを実行します。

例：

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

4. エラーのために導入できなかったグリッドノードがある場合は、エラーを解決し、そのノードだけを対象に Bash スクリプトを再実行します。

例：

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

各ノードのステータスが「PASSED」になると、導入は完了です。

## 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

## StorageGRID の設定を自動化

グリッドノードを導入したら、StorageGRID システムの設定を自動化できます。

開始する前に

- インストールアーカイブにある次のファイルの場所を確認しておきます。

ファイル名	製品説明
configure-storagegrid.py	設定を自動化するための Python スクリプト
storagegrid-sample.json を設定します	スクリプトで使用する構成ファイルの例
storagegrid-bank.json を設定する	スクリプトで使用する空の構成ファイルです

- 構成ファイルを作成しておき `configure-storagegrid.json` ます。このファイルを作成するには (`configure-storagegrid.sample.json`、サンプル構成ファイル) または空の構成ファイル) (`configure-storagegrid.blank.json` を変更します。



変更したファイルのパスワードセクションから管理パスワードとプロビジョニングパスワードを保存します。`configure-storagegrid.json` 構成ファイルを安全な場所に保管します。これらのパスワードは、インストール、拡張、およびメンテナンスの手順に必要です。変更したファイルもバックアップする必要があります `configure-storagegrid.json` 構成ファイルを作成して安全な場所に保存します。

タスクの内容

Python スクリプトと `configure-storagegrid.json` グリッド構成ファイルを使用して、StorageGRID システムの設定を自動化できます `configure-storagegrid.py`。



また、Grid Manager またはインストール API を使用してシステムを設定することもできます。

手順

- Python スクリプトを実行するために使用する Linux マシンにログインします。

2. インストールアーカイブを展開したディレクトリに移動します。

例：

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

`platform`は、debs、rpm、またはvsphereです。

3. Python スクリプトを実行し、作成した構成ファイルを使用します。

例：

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

## 結果

設定プロセス中にリカバリパッケージ`.zip`ファイルが生成され、インストールおよび設定プロセスを実行するディレクトリにダウンロードされます。グリッドノードで障害が発生した場合に StorageGRID システムをリカバリできるようにするために、リカバリパッケージファイルをバックアップする必要があります。たとえば、バックアップされたセキュアなネットワーク上の場所や、安全なクラウドストレージ上の場所にコピーします。



リカバリパッケージファイルには StorageGRID システムからデータを取得するための暗号キーとパスワードが含まれているため、安全に保管する必要があります。

ランダムパスワードを生成するように指定した場合は、ファイルを開き Passwords.txt、StorageGRID システムへのアクセスに必要なパスワードを探します。

```
#####  
##### 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. #####  
#####
```

StorageGRID システムがインストールおよび設定されると、確認メッセージが表示されます。

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

## 関連情報

- ["Grid Manager に移動します"](#)
- ["インストールREST API"](#)

# 仮想グリッドノードを展開する

## StorageGRID用のVMware環境に関する情報を収集

グリッドノードを導入する前に、ネットワーク設定と VMware 環境に関する情報を収集する必要があります。



一部のノードだけを先にインストールしてから、一部のノードだけをインストールするよりも、すべてのノードを1つのインストールの方が効率的です。

### VMware の情報

導入環境にアクセスし、VMware 環境に関する情報、グリッドネットワーク、管理ネットワーク、クライアントネットワーク用に作成されたネットワークに関する情報、およびストレージノードで使用する予定のストレージボリュームタイプに関する情報を収集する必要があります。

VMware 環境に関する次の情報を収集する必要があります。

- 導入を完了するための適切な権限を持つ VMware vSphere アカウントのユーザ名とパスワード。
- 各StorageGRIDノード仮想マシンのホスト、データストア、およびネットワーク構成の情報。



VMware のライブ vMotion を使用すると仮想マシンのクロック時間が急に進むため、この機能はどのタイプのグリッドノードでもサポートされていません。まれにはありますが、クロック時間が不正確だとデータや設定の更新が失われることがあります。

### グリッドネットワークの情報

StorageGRID グリッドネットワーク（必須）用に作成された VMware ネットワークに関する次の情報を収集する必要があります。

- ネットワーク名。
- 静的または DHCP のいずれかの IP アドレスの割り当てに使用する方法。
  - 静的 IP アドレスを使用する場合は、各グリッドノードのネットワークに関する必須の詳細情報（IP アドレス、ゲートウェイ、ネットワークマスク）。
  - DHCPを使用している場合は、グリッドネットワークでのプライマリ管理ノードのIPアドレス。詳細については、[を参照してください "グリッドノードによるプライマリ管理ノードの検出"](#)。

### 管理ネットワークの情報

ノードがオプションの StorageGRID 管理ネットワークに接続される場合は、このネットワーク用に作成された VMware ネットワークに関する次の情報を収集する必要があります。

- ネットワーク名。
- 静的または DHCP のいずれかの IP アドレスの割り当てに使用する方法。
  - 静的 IP アドレスを使用する場合は、各グリッドノードのネットワークに関する必須の詳細情報（IP アドレス、ゲートウェイ、ネットワークマスク）。

- DHCPを使用している場合は、グリッドネットワークでのプライマリ管理ノードのIPアドレス。詳細については、[を参照してください "グリッドノードによるプライマリ管理ノードの検出"](#)。

- 管理ネットワークの外部サブネットリスト（ESL）。

### クライアントネットワークの情報

ノードがオプションの StorageGRID クライアントネットワークに接続される場合は、このネットワーク用に作成された VMware ネットワークに関する次の情報を収集する必要があります。

- ネットワーク名。
- 静的または DHCP のいずれかの IP アドレスの割り当てに使用する方法。
- 静的 IP アドレスを使用する場合は、各グリッドノードのネットワークに関する必須の詳細情報（IP アドレス、ゲートウェイ、ネットワークマスク）。

### 追加のインターフェイスに関する情報

ノードのインストール後に、vCenter で VM にトランクインターフェイスまたはアクセスインターフェイスを追加することもできます。たとえば、管理ノードまたはゲートウェイノードにトランクインターフェイスを追加して、VLAN インターフェイスを使用して複数のアプリケーションまたはテナントに属するトラフィックを分離できます。または、ハイアベイラビリティ（HA）グループで使用するアクセスインターフェイスを追加することもできます。

追加したインターフェイスは、VLAN インターフェイスのページおよび Grid Manager の HA グループのページに表示されます。

- トランクインターフェイスを追加する場合は、新しい親インターフェイスごとに 1 つ以上の VLAN インターフェイスを設定します。[を参照して "VLAN インターフェイスを設定します"](#)
- アクセスインターフェイスを追加した場合は、HA グループに直接追加する必要があります。[を参照して "ハイアベイラビリティグループを設定する"](#)

### 仮想ストレージノードのストレージボリューム

仮想マシンベースのストレージノードに関する次の情報を収集する必要があります。

- 追加するストレージボリューム（ストレージLUN）の数とサイズ。[を参照してください。"ストレージとパフォーマンスの要件"](#)

### グリッドの設定情報

グリッドを設定するための情報を収集する必要があります。

- Grid ライセンス
- Network Time Protocol（NTP；ネットワークタイムプロトコル）サーバの IP アドレス
- DNSサーバのIPアドレス

## Linux デプロイメント用の StorageGRID ノード構成ファイルを作成する

ノード構成ファイルは、ノードを起動して適切なネットワークおよびブロックストレージ

グリッドに接続するために StorageGRID ホストサービスで必要となる情報をまとめた小さいテキストファイルです。ノード構成ファイルは仮想ノードに使用され、アプリケーションノードには使用されません。



「Linux」は、RHEL、Ubuntu、または Debian のデプロイメントを指します。サポートされているバージョンのリストについては、"[NetApp Interoperability Matrix Tool \(IMT\)](#)"。

## ノード構成ファイルの場所

各 StorageGRID ノードの構成ファイルを、そのノードを実行するホストのディレクトリに配置し `/etc/storagegrid/nodes`` ます。たとえば、HostA で管理ノード、ゲートウェイノード、およびストレージノードを1つずつ実行する場合は、3つのノード構成ファイルを HostA の `/etc/storagegrid/nodes`` に配置する必要があります。

構成ファイルは、vim や nano などのテキストエディタを使用して各ホストで直接作成することも、別の場所で作成して各ホストに移動することもできます。

## ノード構成ファイルの命名

構成ファイルの名前は、`node-name.conf`` の形式はです。はノードに割り当てる名前です。`node-name` の名前は StorageGRID インストーラに表示され、ノード移行などのノードのメンテナンス処理で使用されません。

ノード名は次のルールに従って付ける必要があります。

- 一意でなければなりません
- 1文字目はアルファベットにする必要があります
- A~Z と a~z のアルファベットを使用できます
- 0~9 の数字を使用できます
- 1つまたは複数のハイフン (-) を含めることができます。
- 拡張子は含めず、32文字以下にする必要があります。 `.conf``

これらの命名規則に従わない内のファイルは、`/etc/storagegrid/nodes`` ホストサービスによって解析されません。

グリッドでマルチサイトトポロジを使用する場合の一般的なノード名は次のようになります。

```
site-nodetype-nodenummer.conf
```

たとえば、データセンター1の最初の管理ノードに `dc1-adm1.conf`` を使用し、データセンター2の3番目のストレージノードに `dc2-sn3.conf`` を使用できます。ただし、すべてのノード名がルールに従っていれば、別の名前にしてもかまいません。

## ノード構成ファイルの内容

構成ファイルには、1行に1つのキーと1つの値を持つキーと値のペアが含まれています。キーと値のペアごとに、次のルールに従ってください。

- キーと値は等号で区切る必要があります( `=` ます) 、およびオプションの空白文字で区切る必要があります。

- キーにスペースを含めることはできません。
- 値にはスペースを含めることができます。
- 先頭または末尾の空白は無視されます。

次の表に、サポートされているすべてのキーの値を示します。各キーには、次のいずれかの指定があります。

- 必須：すべてのノードまたは指定したノードタイプに必須
- ベストプラクティス：オプション（推奨されますが）
- オプション：すべてのノードでオプション

#### 管理ネットワークキー

##### ADMIN\_IP を指定します

値	名称
Linux ベースのノードをインストールするために使用する管理ノードのグリッド ネットワーク IPv4 アドレス。リカバリの場合、プライマリ管理ノードの IP が使用可能な場合はそれを使用し、そうでない場合は非プライマリ管理ノードの IP を使用します。このパラメータを省略すると、ノードは mDNS を使用してプライマリ管理ノードを検出しようとします。  "グリッドノードによるプライマリ管理ノードの検出"  • 注 *：この値は無視されます。また、プライマリ管理ノードでは禁止される場合があります。	ベストプラクティス

##### ADMIN\_NETWORK\_CONFIG

値	名称
DHCP、STATIC、または DISABLED	オプション

##### ADMIN\_NETWORK\_ESL

値	名称
このノードが管理ネットワークゲートウェイを使用して通信するサブネット（CIDR表記）をカンマで区切ったリスト。  例：172.16.0.0/21,172.17.0.0/21	オプション

##### ADMIN\_NETWORK\_GATEWAY

値	名称
<p>このノードのローカルの管理ネットワークゲートウェイの IPv4 アドレス。ADMIN_NETWORK_IP および ADMIN_NETWORK_MASK で定義されるサブネットに属している必要があります。この値は、DHCP によって設定されたネットワークでは無視されます。</p> <p>例：</p> <p>1.1.1.1</p> <p>10.224.4.81</p>	<p>を指定した場合は必須 `ADMIN_NETWORK_ESL` です。それ以外の場合はオプション。</p>

### ADMIN\_NETWORK\_IP

値	名称
<p>このノードの管理ネットワークにおける IPv4 アドレス。このキーが必要なのは、ADMIN_NETWORK_CONFIG = STATIC の場合だけです。それ以外の値の場合は指定しないでください。</p> <p>例：</p> <p>1.1.1.1</p> <p>10.224.4.81</p>	<p>ADMIN_NETWORK_CONFIG = STATIC の場合に必要です。</p> <p>それ以外の場合はオプション。</p>

### ADMIN\_NETWORK\_MAC

値	名称
<p>コンテナ内の管理ネットワークインターフェイスの MAC アドレス。</p> <p>このフィールドはオプションです。省略すると、MAC アドレスが自動的に生成されます。</p> <p>6 つの 16 進数値をコロンで区切って指定する必要があります。</p> <p>例： b2:9c:02:c2:27:10</p>	<p>オプション</p>

### ADMIN\_NETWORK\_MASK

値	名称
<p>このノードの管理ネットワークにおける IPv4 ネットマスク。ADMIN_NETWORK_CONFIG = STATICの場合はこのキーを指定します。それ以外の値の場合は指定しないでください。</p> <p>例：</p> <p>255.255.255.0</p> <p>255.255.248.0</p>	<p>ADMIN_NETWORK_IPを指定し、ADMIN_NETWORK_CONFIG = STATICの場合は必須です。</p> <p>それ以外の場合はオプション。</p>

### ADMIN\_NETWORK\_MTU を指定します

値	名称
<p>このノードの管理ネットワークでの最大伝送ユニット（MTU）。ADMIN_NETWORK_CONFIG = DHCPの場合は指定しないでください。この値を指定する場合、1280 ~ 9216 の範囲で指定する必要があります。省略すると、1500 が使用されます。</p> <p>ジャンボフレームを使用する場合は、MTU を 9000 などのジャンボフレームに適した値に設定します。それ以外の場合は、デフォルト値のままにします。</p> <ul style="list-style-type: none"> <li>• <b>重要 *</b>：ネットワークの MTU 値は、ノードが接続されているスイッチポートに設定された値と一致する必要があります。そうしないと、ネットワークパフォーマンスの問題やパケット損失が発生する可能性があります。</li> </ul> <p>例：</p> <p>1500</p> <p>8192</p>	<p>オプション</p>

### ADMIN\_NETWORK\_TARGET

値	名称
<p>StorageGRID ノードで管理ネットワークのアクセスに使用するホストデバイスの名前。ネットワークインターフェイス名のみがサポートされています。通常、GRID_NETWORK_TARGET または CLIENT_NETWORK_TARGET に指定したインターフェイス名とは別のインターフェイス名を使用します。</p> <p>注：ボンドデバイスやブリッジデバイスをネットワークターゲットとして使用しないでください。ボンドデバイスの上に VLAN（または他の仮想インターフェイス）を設定するか、ブリッジと仮想イーサネット（veth）のペアを使用します。</p> <ul style="list-style-type: none"> <li>• ベストプラクティス*：管理ネットワークの IP アドレスは、このノードで最初は使用しない場合でも値を指定します。そうすることで、ホストでノードの設定を再度行わなくても、管理ネットワークの IP アドレスをあとから追加することができます。</li> </ul> <p>例：</p> <p>bond0.1002</p> <p>ens256</p>	ベストプラクティス

#### ADMIN\_NETWORK\_TARGET タイプ

値	名称
interface（サポートされている値はこれだけです）	オプション

#### ADMIN\_NETWORK\_TARGET\_TYPE\_interface\_clone\_MAC

値	名称
<p>正しいか間違っているか</p> <p>StorageGRID コンテナで管理ネットワークのホストターゲットインターフェイスの MAC アドレスを使用するには、キーを「true」に設定して原因に設定します。</p> <ul style="list-style-type: none"> <li>• ベストプラクティス：プロミスキャスモードが必要なネットワークでは、「ADMIN_NETWORK_TARGET_TYPE_interface_clone_MAC」キーを使用してください。</li> </ul> <p>LinuxのMACクローニングの詳細については、以下を参照してください。<a href="#">"MAC アドレスのクローニングに関する考慮事項と推奨事項"</a></p>	ベストプラクティス

## ADMIN\_NETWORK\_ROLE

値	名称
プライマリまたは非プライマリ  このキーが必要なのは、NODE_TYPE = VM_ADMIN_Nodeの場合のみです。それ以外のタイプのノードの場合は指定しないでください。	NODE_TYPE = VM_Admin_Nodeの場合は必須  それ以外の場合はオプション。

ブロックデバイスキー

## BLOBK\_DEVICE\_AUDIT\_logs

値	名称
このノードで監査ログの永続的なストレージに使用するブロックデバイススペシャルファイルのパスと名前。  例：  <code>/dev/disk/by-path/pci-0000:03:00.0-scsi-0:0:0:0</code>  <code>/dev/disk/by-id/wwn-0x600a09800059d6df000060d757b475fd</code>  <code>/dev/mapper/sgws-adml-audit-logs</code>	NODE_TYPE = VM_Admin_Nodeのノードに必要です。他のノードタイプの場合は指定しないでください。

## block\_device\_rangedb\_nnn

値	名称
<p>このノードでオブジェクトの永続的なストレージに使用するブロックデバイススペシャルファイルのパスと名前。このキーが必要なのは、NODE_TYPE = VM_Storage_Nodeのノードだけです。それ以外のタイプのノードの場合は指定しないでください。</p> <p>BLOCK_DEVICE_RANGEDB_000のみが必須で、それ以外は省略可能です。BLOCK_DEVICE_RANGEDB_000に指定するブロックデバイスは4TB以上である必要があります。それ以外は4TB未満でもかまいません。</p> <p>隙間を空けてはいけません。BLOCK_DEVICE_RANGEDB_005を指定する場合は、BLOCK_DEVICE_RANGEDB_004も指定されている必要があります。</p> <ul style="list-style-type: none"> <li>注 * : 既存の環境との互換性を確保するため、アップグレードされたノードでは2桁のキーがサポートされています。</li> </ul> <p>例 :</p> <pre>/dev/disk/by-path/pci-0000:03:00.0-scsi-0:0:0:0</pre> <pre>/dev/disk/by-id/wwn-0x600a09800059d6df000060d757b475fd</pre> <pre>/dev/mapper/sgws-sn1-rangedb-000</pre>	<p>必須 :</p> <p>BLOCK_DEVICE_RANGEDB_000</p> <p>オプション :</p> <p>BLOCK_DEVICE_RANGEDB_001</p> <p>BLOCK_DEVICE_RANGEDB_002</p> <p>BLOCK_DEVICE_RANGEDB_003</p> <p>BLOCK_DEVICE_RANGEDB_004</p> <p>BLOCK_DEVICE_RANGEDB_005</p> <p>BLOCK_DEVICE_RANGEDB_006</p> <p>BLOCK_DEVICE_RANGEDB_007</p> <p>BLOCK_DEVICE_RANGEDB_008</p> <p>BLOCK_DEVICE_RANGEDB_009</p> <p>BLOCK_DEVICE_RANGEDB_010</p> <p>BLOCK_DEVICE_RANGEDB_011</p> <p>BLOCK_DEVICE_RANGEDB_012</p> <p>BLOCK_DEVICE_RANGEDB_013</p> <p>BLOCK_DEVICE_RANGEDB_014</p> <p>BLOCK_DEVICE_RANGEDB_015</p>

## BLOBK\_DEVICE\_tables

値	名称
<p>このノードでデータベーステーブルの永続的なストレージに使用するブロックデバイススペシャルファイルのパスと名前。このキーが必要なのは、<code>NODE_TYPE = VM_ADMIN_Node</code>のノードだけです。それ以外のタイプのノードの場合は指定しないでください。</p> <p>例：</p> <pre>/dev/disk/by-path/pci-0000:03:00.0-scsi-0:0:0:0</pre> <pre>/dev/disk/by-id/wwn-0x600a09800059d6df000060d757b475fd</pre> <pre>/dev/mapper/sgws-adm1-tables</pre>	必須

### **BLOBK\_DEVICE\_VAR\_LOCAL** です

値	名称
<p>このノードの永続的ストレージに使用するブロックデバイススペシャルファイルのパスと名前 <code>/var/local</code>。</p> <p>例：</p> <pre>/dev/disk/by-path/pci-0000:03:00.0-scsi-0:0:0:0</pre> <pre>/dev/disk/by-id/wwn-0x600a09800059d6df000060d757b475fd</pre> <pre>/dev/mapper/sgws-sn1-var-local</pre>	必須

クライアントネットワークキー

### **CLIENT\_NETWORK\_CONFIG**

値	名称
DHCP、STATIC、または DISABLED	オプション

### **CLIENT\_NETWORK\_GATEWAY**

値	名称

<p>このノードのローカルのクライアントネットワークゲートウェイの IPv4 アドレス。 CLIENT_NETWORK_IP および CLIENT_NETWORK_MASK で定義されるサブネットに属している必要があります。この値は、DHCP によって設定されたネットワークでは無視されます。</p> <p>例：</p> <p>1.1.1.1</p> <p>10.224.4.81</p>	オプション
--	-------

## CLIENT\_NETWORK\_IP

値	名称
<p>このノードのクライアントネットワークにおける IPv4 アドレス。</p> <p>このキーが必要なのは、CLIENT_NETWORK_CONFIG = STATICの場合だけです。それ以外の値の場合は指定しないでください。</p> <p>例：</p> <p>1.1.1.1</p> <p>10.224.4.81</p>	<p>client_network_config = staticの場合に必要</p> <p>それ以外の場合はオプション。</p>

## CLIENT\_NETWORK\_MAC

値	名称
<p>コンテナ内のクライアントネットワークインターフェイスの MAC アドレス。</p> <p>このフィールドはオプションです。省略すると、MAC アドレスが自動的に生成されます。</p> <p>6 つの 16 進数値をコロンで区切って指定する必要があります。</p> <p>例： b2:9c:02:c2:27:20</p>	オプション

## CLIENT\_NETWORK\_MASK

値	名称
<p>このノードのクライアントネットワークにおける IPv4 ネットマスク。</p> <p>CLIENT_NETWORK_CONFIG = STATICの場合にこのキーを指定します。他の値の場合は指定しないでください。</p> <p>例：</p> <p>255.255.255.0</p> <p>255.255.248.0</p>	<p>CLIENT_NETWORK_IPを指定し、CLIENT_NETWORK_CONFIG = STATICの場合は必須</p> <p>それ以外の場合はオプション。</p>

## CLIENT\_NETWORK\_MTU

値	名称
<p>このノードのクライアントネットワークでの最大伝送ユニット（MTU）。CLIENT_NETWORK_CONFIG = DHCPの場合は指定しないでください。この値を指定する場合、1280 ~ 9216 の範囲で指定する必要があります。省略すると、1500 が使用されます。</p> <p>ジャンボフレームを使用する場合は、MTU を 9000 などのジャンボフレームに適した値に設定します。それ以外の場合は、デフォルト値のままにします。</p> <ul style="list-style-type: none"> <li>重要*：ネットワークの MTU 値は、ノードが接続されているスイッチポートに設定された値と一致する必要があります。そうしないと、ネットワークパフォーマンスの問題やパケット損失が発生する可能性があります。</li> </ul> <p>例：</p> <p>1500</p> <p>8192</p>	<p>オプション</p>

client\_network\_target です

値	名称
<p>StorageGRID ノードでクライアントネットワークのアクセスに使用するホストデバイスの名前。ネットワークインターフェイス名のみがサポートされています。通常、GRID_NETWORK_TARGET または ADMIN_NETWORK_TARGET に指定したインターフェイス名とは別のインターフェイス名を使用します。</p> <p>注：ボンドデバイスやブリッジデバイスをネットワークターゲットとして使用しないでください。ボンドデバイスの上に VLAN（または他の仮想インターフェイス）を設定するか、ブリッジと仮想イーサネット（veth）のペアを使用します。</p> <ul style="list-style-type: none"> <li>• ベストプラクティス：* クライアントネットワークの IP アドレスは、このノードで最初には使用しない場合でも値を指定してください。そうすることで、ホストでノードの設定を再度行わなくても、クライアントネットワークの IP アドレスをあとから追加することができます。</li> </ul> <p>例：</p> <p>bond0.1003</p> <p>ens423</p>	<p>ベストプラクティス</p>

#### client\_network\_target\_type

値	名称
interface（サポートされている値のみ）	オプション

#### client\_network\_target\_type\_interface\_clone\_MAC

値	名称
<p>正しいか間違っているか</p> <p>クライアントネットワークでホストターゲットインターフェイスの MAC アドレスを使用するには、キーを「true」に設定して StorageGRID コンテナを原因します。</p> <ul style="list-style-type: none"> <li>• ベストプラクティス：プロミスキャスモードが必要なネットワークでは、client_network_target_type_interface_clone_MAC キーを使用してください。</li> </ul> <p>LinuxのMACクローニングの詳細については、以下を参照してください。<a href="#">"MAC アドレスのクローニングに関する考慮事項と推奨事項"</a></p>	<p>ベストプラクティス</p>

### GRID\_NETWORK\_CONFIG

値	名称
<p>STATIC または DHCP</p> <p>指定しない場合のデフォルトはstaticです。</p>	<p>ベストプラクティス</p>

### GRID\_NETWORK\_GATEWAY

値	名称
<p>このノードのローカルのグリッドネットワークゲートウェイの IPv4 アドレス。 GRID_NETWORK_IP および GRID_NETWORK_MASK で定義されるサブネットに属している必要があります。この値は、 DHCP によって設定されたネットワークでは無視されます。</p> <p>グリッドネットワークのサブネットが 1 つだけでゲートウェイがない場合は、サブネットの標準のゲートウェイアドレス (X.Y.Z.1) か、このノードの GRID_NETWORK_IP の値を使用します。このどちらかの値にしておけば、以降にグリッドネットワークを拡張するときに処理が簡単になります。</p>	<p>必須</p>

### GRID\_NETWORK\_IP

値	名称
<p>このノードのグリッドネットワークにおける IPv4 アドレス。このキーが必要なのは、 GRID_NETWORK_CONFIG = STATIC の場合のみです。それ以外の値の場合は指定しないでください。</p> <p>例：</p> <p>1.1.1.1</p> <p>10.224.4.81</p>	<p>GRID_NETWORK_CONFIG = STATIC の場合は必須</p> <p>それ以外の場合はオプション。</p>

### GRID\_NETWORK\_MAC

値	名称
<p>コンテナ内のグリッドネットワークインターフェイスの MAC アドレス。</p> <p>6 つの 16 進数値をコロンで区切って指定する必要があります。</p> <p>例： b2:9c:02:c2:27:30</p>	<p>オプション</p> <p>省略すると、MAC アドレスが自動的に生成されます。</p>

## GRID\_NETWORK\_MASK

値	名称
<p>このノードのグリッドネットワークにおける IPv4 ネットマスク。GRID_NETWORK_CONFIG = STATICの場合はこのキーを指定します。それ以外の値の場合は指定しないでください。</p> <p>例：</p> <p>255.255.255.0</p> <p>255.255.248.0</p>	<p>GRID_NETWORK_IPを指定し、GRID_NETWORK_CONFIG = STATICを指定した場合に必要です。</p> <p>それ以外の場合はオプション。</p>

## GRID\_NETWORK\_MTU

値	名称
<p>このノードのグリッドネットワークでの最大伝送ユニット（MTU）。GRID_NETWORK_CONFIG = DHCPの場合は指定しないでください。この値を指定する場合、1280 ~ 9216 の範囲で指定する必要があります。省略すると、1500 が使用されます。</p> <p>ジャンボフレームを使用する場合は、MTU を 9000 などのジャンボフレームに適した値に設定します。それ以外の場合は、デフォルト値のままにします。</p> <ul style="list-style-type: none"><li>重要*：ネットワークの MTU 値は、ノードが接続されているスイッチポートに設定された値と一致する必要があります。そうしないと、ネットワークパフォーマンスの問題やパケット損失が発生する可能性があります。</li><li>重要*：ネットワークパフォーマンスを最大限に高めるには、すべてのノードのグリッドネットワークインターフェイスで MTU 値がほぼ同じになるように設定する必要があります。個々のノードのグリッドネットワークの MTU 設定に大きな違いがある場合は、* Grid Network MTU mismatch * アラートがトリガーされます。MTU 値はすべてのネットワークタイプで同じである必要はありません。</li></ul> <p>例：</p> <p>1500</p> <p>8192</p>	<p>オプション</p>

## GRID\_NETWORK\_TARGET

値	名称
<p>StorageGRID ノードでグリッドネットワークのアクセスに使用するホストデバイスの名前。ネットワークインターフェイス名のみがサポートされています。通常、ADMIN_NETWORK_TARGET または ADMIN_NETWORK_TARGET に指定したインターフェイス名とは別のインターフェイス名を使用します。</p> <p>注：ボンドデバイスやブリッジデバイスをネットワークターゲットとして使用しないでください。ボンドデバイスの上に VLAN（または他の仮想インターフェイス）を設定するか、ブリッジと仮想イーサネット（veth）のペアを使用します。</p> <p>例：</p> <p>bond0.1001</p> <p>ens192</p>	必須

### GRID\_NETWORK\_TARGET タイプ

値	名称
interface（サポートされている値はこれだけです）	オプション

### GRID\_NETWORK\_TARGET\_TYPE\_interface\_clone\_MAC

値	名称
<p>正しいか間違っているか</p> <p>グリッドネットワーク上のホストターゲットインターフェイスの MAC アドレスを使用するには、キーの値を「true」に設定して StorageGRID コンテナを原因 に設定します。</p> <ul style="list-style-type: none"> <li>• ベストプラクティス：プロミスキャスモードが必要なネットワークでは、GRID_NETWORK_TARGET_TYPE_interface_clone_MAC キーを使用してください。</li> </ul> <p>LinuxのMACクローニングの詳細については、以下を参照してください。<a href="#">"MAC アドレスのクローニングに関する考慮事項と推奨事項"</a></p>	ベストプラクティス

インストールパスワードキー（一時）

### custom\_temporary\_password\_hash

値	名称
<p>プライマリ管理ノードの場合は、インストール時にStorageGRIDインストールAPIのデフォルトの一時パスワードを設定します。</p> <p>注：インストールパスワードはプライマリ管理ノードにのみ設定します。別のタイプのノードでパスワードを設定しようとすると、ノード構成ファイルの検証に失敗します。</p> <p>この値を設定しても、インストールが完了しても効果はありません。</p> <p>このキーを省略すると、デフォルトでは一時パスワードは設定されません。または、StorageGRIDインストールAPIを使用して一時パスワードを設定することもできます。</p> <p>8文字以上32文字以下のパスワードの形式のSHA-512パスワードハッシュで <code>\$6\$&lt;salt&gt;\$&lt;password hash&gt;</code> する必要があります `crypt()`。</p> <p>このハッシュは、SHA-512モードのコマンドなどのCLIツールを使用して生成できます <code>openssl passwd</code>。</p>	ベストプラクティス

#### interfacesキー

#### interface\_target\_nnnn

値	名称
<p>このノードに追加するインターフェイスの名前とオプションの概要。各ノードに複数のインターフェイスを追加できます。</p> <p><code>_nnnn_</code>には、追加する各interface_targetエントリに一意的番号を指定します。</p> <p>値には、ベアメタルホスト上の物理インターフェイスの名前を指定します。その後、必要に応じて、カンマを追加してインターフェイスの概要を指定します。このインターフェイスは、VLAN インターフェイスのページと HA グループのページに表示されます。</p> <p>例： <code>INTERFACE_TARGET_0001=ens256, Trunk</code></p> <p>トランクインターフェイスを追加する場合は、StorageGRIDでVLANインターフェイスを設定する必要があります。アクセスインターフェイスを追加する場合は、そのインターフェイスをHAグループに直接追加できます。VLANインターフェイスを設定する必要はありません。</p>	オプション

#### 最大RAMキー

#### MAXIMUM\_RAM

値	名称
<p>このノードに使用を許可する RAM の最大容量。このキーを省略した場合、ノードでメモリは制限されません。本番用のノードについて設定するときは、システム RAM の合計容量よりも 24GB 以上、16~32GB 以上小さい値を指定してください。</p> <ul style="list-style-type: none"> <li>注*：RAM 値は、ノードの実際のメタデータ用リザーブスペースに影響します。を参照してください"<a href="#">Metadata Reserved Spaceとは何かの概要</a>"。</li> </ul> <p>このフィールドの形式は <code>numberunit</code>。 <code>unit`</code> には、 <code>`k</code>、 <code>m</code>、または <code>g`</code> を指定できます <code>`b</code>。</p> <p>例：</p> <p>24g</p> <p>38654705664b</p> <ul style="list-style-type: none"> <li>注：このオプションを使用する場合は、 <code>memory cgroups</code> のカーネルサポートを有効にする必要があります。</li> </ul>	オプション

#### ノードタイプキー

**Node\_type** のように指定します

値	名称
<p>ノードのタイプ：</p> <ul style="list-style-type: none"> <li>VM_Admin_Nodeの略</li> <li>VM_Storage_Nodeの略</li> <li>VM_Archive_Nodeの略</li> <li>VM_API_Gateway</li> </ul>	必須

#### ストレージタイプ

値	名称
<p>ストレージノードに含まれるオブジェクトのタイプを定義。詳細については、<a href="#">を参照してください "ストレージノードのタイプ"</a>。このキーが必要なのは、NODE_TYPE = VM_Storage_Nodeのノードだけです。それ以外のタイプのノードの場合は指定しないでください。ストレージタイプ:</p> <ul style="list-style-type: none"> <li>• 組み合わせ ( Combined )</li> <li>• データ</li> <li>• メタデータ</li> </ul> <p>注: storage_typeを指定しない場合、ストレージノードタイプはデフォルトで組み合わせ (データとメタデータ) に設定されます。</p>	オプション

#### ポートの再マッピングキー



ポート再マッピングのサポートは非推奨であり、将来のリリースでは削除される予定です。再マップされたポートを削除するには、["ベアメタルホストでのポートの再マッピングを削除します"](#)。

#### PORT\_REMAP を参照してください

値	名称
<p>ノードが内部でのグリッドノードの通信または外部との通信に使用するポートを再マッピングします。ポートの再マッピングが必要になるのは、またはの説明に従って、StorageGRIDで使用される1つ以上のポートがエンタープライズネットワークポリシーによって制限されている場合です。"<a href="#">内部でのグリッドノードの通信</a>" "<a href="#">外部との通信</a>"</p> <p>重要: ロードバランサエンドポイントの設定に使用する予定のポートを再マッピングしないでください。</p> <ul style="list-style-type: none"> <li>• 注: PORT_REMAP のみを設定すると、指定したマッピングがインバウンド通信とアウトバウンド通信の両方に使用されません。PORT_REMAP_INBOUND を併せて指定した場合は、PORT_REMAP がアウトバウンド通信のみに適用されます。</li> </ul> <p>使用される形式は、`network type/protocol/default port used by grid node/new port` です。`network type` は grid、admin、または client、`protocol` は tcp または udp です。</p> <p>例: PORT_REMAP = client/tcp/18082/443</p> <p>カンマで区切ったリストを使用して複数のポートを再マッピングすることもできます。</p> <p>例: PORT_REMAP = client/tcp/18082/443, client/tcp/18083/80</p>	オプション

## PORT\_REMAP\_INBOUND

値	名称
<p>指定したポートのインバウンド通信を再マッピングします。PORT_REMAP_INBOUNDを指定し、PORT_REMAPに値を指定しなかった場合、ポートのアウトバウンド通信は変更されません。</p> <p>重要：ロードバランサエンドポイントの設定に使用する予定のポートを再マッピングしないでください。</p> <p>使用される形式は、`network type/protocol/remapped port/default port used by grid node`です。`network type`はgrid、admin、またはclient、`protocol`はtcpまたはudpです。</p> <p>例：PORT_REMAP_INBOUND = grid/tcp/3022/22</p> <p>カンマで区切った複数のインバウンドポートを再マッピングすることもできます。</p> <p>例：PORT_REMAP_INBOUND = grid/tcp/3022/22, admin/tcp/3022/22</p>	オプション

### グリッドノードがStorageGRIDでプライマリ管理ノードを検出する方法

グリッドノードは、設定や管理のためにプライマリ管理ノードと通信します。各グリッドノードがグリッドネットワーク上のプライマリ管理ノードのIPアドレスを認識している必要があります。

グリッドノードからプライマリ管理ノードにアクセスできるようにするために、ノードを導入する際に次のいずれかを実行します。

- ADMIN\_IP パラメータを使用して、プライマリ管理ノードのIPアドレスを手動で入力します。
- ADMIN\_IP パラメータを省略して、グリッドノードで自動的に値が検出されるようにします。自動検出は、グリッドネットワークでDHCPを使用してプライマリ管理ノードにIPアドレスを割り当てる場合に特に便利です。

プライマリ管理ノードの自動検出は、マルチキャストドメインネームシステム (mDNS) を使用して実行されます。プライマリ管理ノードは、最初に起動されるときに、mDNS を使用してそのノードのIPアドレスを公開します。同じサブネット上の他のノードは、このIPアドレスを自動的に照会して取得します。ただし、通常、マルチキャストIPトラフィックはサブネット間でルーティングできないため、他のサブネット上のノードはプライマリ管理ノードのIPアドレスを直接取得できません。

自動検出を使用する場合：



- プライマリ管理ノードが直接接続されていないサブネットの少なくとも 1 つのグリッドノードで、ADMIN\_IP 設定を指定する必要があります。このグリッドノードがプライマリ管理ノードの IP アドレスを公開することで、サブネット上の他のノードが mDNS を使用して IP アドレスを検出できるようになります。
- ネットワークインフラがサブネット内のマルチキャスト IP トラフィックの転送をサポートしていることを確認します。

## VMware vSphere を使用して StorageGRID の仮想マシンとしてグリッドノードを導入

VMware vSphere Web Client を使用して、各グリッドノードを仮想マシンとして導入します。導入時に、各グリッドノードが作成されて、1 つ以上の StorageGRID ネットワークに接続されます。

StorageGRID アプライアンスストレージノードを導入する必要がある場合は、を参照してください ["アプライアンスストレージノードを導入する"](#)。

必要に応じて、ノードポートを再マッピングしたり、ノードの CPU やメモリの設定を増やしたりして、電源をオンにすることができます。

開始する前に

- 方法を確認し ["設置を計画して準備"](#)、ソフトウェア、CPU と RAM、ストレージとパフォーマンスの要件を把握しておく必要があります。
- VMware vSphere ハイパーバイザーについて理解し、この環境で仮想マシンの導入を経験している必要があります。



この `open-vm-tools` パッケージは、VMware Tools に似たオープンソースの実装であり、StorageGRID 仮想マシンに含まれています。VMware Tools を手動でインストールする必要はありません。

- VMware 用の正しいバージョンの StorageGRID インストールアーカイブをダウンロードして展開しておきます。



拡張またはリカバリ処理の一環として新しいノードを導入する場合は、グリッドで現在実行されているバージョンの StorageGRID を使用する必要があります。

- StorageGRID 仮想マシンディスク (`.vmdk`) ファイルが必要です。

```
NetApp-SG-version-SHA.vmdk
```

- 導入するグリッドノードのタイプごとにファイルと `.mf` ファイルを用意しておき `.ovf` ます。

ファイル名	製品説明
vsphere-primary-admin.ovf vsphere-primary-admin.mf	プライマリ管理ノードのテンプレートファイルとマニフェストファイル。
vsphere-non-primary-admin.ovf vsphere-non-primary-admin.mf	非プライマリ管理ノードのテンプレートファイルとマニフェストファイル。
vsphere-storage.ovf vsphere-storage.mf	ストレージノードのテンプレートファイルとマニフェストファイル。
vsphere-gateway.ovf vsphere-gateway.mf	ゲートウェイノードのテンプレートファイルとマニフェストファイル。

- .vdmk、.ovf、および`.mf`のファイルはすべて同じディレクトリにあります。
- 障害ドメインを最小限に抑えるための計画が必要です。たとえば、すべてのゲートウェイノードを単一のvSphere ESXiホストに導入することは避けてください。



本番環境では、1台の仮想マシンで複数のストレージノードを実行しないでください。許容できない障害ドメインの問題が発生する場合は、同じESXiホストで複数の仮想マシンを実行しないでください。

- 拡張またはリカバリ処理でノードを導入する場合は、またはが必要"[StorageGRID システムの拡張手順](#)"[リカバリとメンテナンスの手順](#)"です。
- NetApp ONTAP システムからストレージが割り当てられた仮想マシンとしてStorageGRID ノードを導入する場合は、ボリュームでFabricPool 階層化ポリシーが有効になっていないことを確認しておきます。たとえば、StorageGRIDノードがVMwareホストで仮想マシンとして実行されている場合は、そのノードのデータストアを作成するボリュームでFabricPool階層化ポリシーが有効になっていないことを確認してください。StorageGRIDノードで使用するボリュームでFabricPool階層化を無効にすると、トラブルシューティングとストレージの処理が簡単になります。



FabricPoolを使用して、StorageGRIDに関連するデータをStorageGRID自体に階層化しないでください。StorageGRIDデータをStorageGRIDに階層化すると、トラブルシューティングや運用が複雑になります。

## タスクの内容

最初に VMware ノードを導入するとき、拡張時に新しい VMware ノードを追加するとき、またはリカバリ処理の一環として VMware ノードを交換するときは、次の手順に従います。手順に記載されている場合を除き、ノードの導入手順は、管理ノード、ストレージノード、ゲートウェイノードを含むすべてのタイプのノードで同じです。

新しい StorageGRID システムを設置する場合は、次の手順を実行します。

- ノードは任意の順序で導入できます。
- 各仮想マシンがグリッドネットワーク経由でプライマリ管理ノードに接続できることを確認する必要があります。
- グリッドを設定する前に、すべてのグリッドノードを導入する必要があります。

拡張またはリカバリ処理を実行する場合は、次の手順を実行します。

- 新しい仮想マシンがグリッドネットワーク経由で他のすべてのノードに接続できることを確認する必要があります。

ノードのポートを再マッピングする必要がある場合は、ポートの再マッピングの設定が完了するまで新しいノードの電源をオンにしないでください。



ポート再マッピングのサポートは非推奨であり、将来のリリースでは削除される予定です。再マップされたポートを削除するには、"[ベアメタルホストでのポートの再マッピングを削除します](#)"。

## 手順

### 1. vCenter を使用して OVF テンプレートを導入

URL を指定する場合は、次のファイルを含むフォルダを指定します。それ以外の場合は、ローカルディレクトリから各ファイルを選択します。

```
NetApp-SG-version-SHA.vmdk  
vsphere-node.ovf  
vsphere-node.mf
```

たとえば、導入する最初のノードがこのファイルに含まれている場合は、次のファイルを使用して StorageGRID システムのプライマリ管理ノードを導入します。

```
NetApp-SG-version-SHA.vmdk  
vsphere-primary-admin.ovf  
vsphere-primary-admin.mf
```

### 2. 仮想マシンの名前を指定します。

標準的には、仮想マシンとグリッドノードに同じ名前を使用します。

### 3. 仮想マシンを適切な vApp またはリソースプールに配置します。

### 4. プライマリ管理ノードを導入する場合は、エンドユーザライセンス契約を読んで同意します。

vCenter のバージョンによっては、使用する手順の順序は、エンドユーザライセンス契約を承諾し、仮想マシンの名前を指定し、データストアを選択する場合とで異なります。

### 5. 仮想マシンのストレージを選択します。

リカバリ処理の一環としてノードを導入する場合は、の手順に従って、[ストレージリカバリ手順](#)新しい仮想ディスクの追加、障害が発生したグリッドノードからの仮想ハードディスクの再接続、またはその両方を行います。

ストレージノードを導入する際は、ストレージボリュームを 3 個以上使用し、各ストレージボリュームのサイズを 4TB 以上にします。ボリューム 0 に少なくとも 4TB 割り当てる必要があります。



ストレージノードの .ovf ファイルは、ストレージ用の複数の VMDK を定義します。これらの VMDK がストレージ要件を満たしていない場合は、ノードの電源を入れる前に、それらの VMDK を削除し、ストレージに適切な VMDK または RDM を割り当てる必要があります。VMware 環境で一般に使用され、管理も容易であるのは VMDK ですが、大きなオブジェクトサイズ（たとえば 100MB 超）を使用するワークロードのパフォーマンスは RDM の方が高くなります。



一部の StorageGRID 環境では、一般的な仮想ワークロードよりも大容量のアクティブなストレージボリュームを使用する場合があります。パフォーマンスを最適化するために、などの一部のハイパーバイザーパラメータの調整が必要になる場合があります MaxAddressableSpaceTB。パフォーマンスが低下する場合は、仮想化のサポートリソースに問い合わせ、ワークロード固有の構成調整によって環境がメリットを受けるかどうかを確認してください。

## 6. ネットワークを選択します。

各ソースネットワークのデスティネーションネットワークを選択して、ノードで使用する StorageGRID ネットワークを決定します。

- グリッドネットワークは必須です。vSphere 環境でデスティネーションネットワークを選択する必要があります。+グリッドネットワークは、すべての内部StorageGRIDトラフィックに使用されます。グリッド内のすべてのノードが、すべてのサイトとサブネットにわたって接続されます。グリッドネットワーク上のすべてのノードが他のすべてのノードと通信できる必要があります。
- 管理ネットワークを使用する場合は、vSphere 環境で別のデスティネーションネットワークを選択します。管理ネットワークを使用しない場合は、グリッドネットワークに対して選択したデスティネーションと同じデスティネーションを選択します。
- クライアントネットワークを使用する場合は、vSphere 環境で別のデスティネーションネットワークを選択します。クライアントネットワークを使用しない場合は、グリッドネットワークに対して選択したデスティネーションと同じデスティネーションを選択します。
- 管理ネットワークまたはクライアントネットワークを使用する場合は、ノードが同じ管理ネットワークまたはクライアントネットワーク上にある必要はありません。

## 7. [テンプレートのカスタマイズ]\*で、必要なStorageGRIDノードプロパティを構成します。

### a. ノード名 \* を入力します。



グリッドノードをリカバリする場合は、リカバリするノードの名前を入力する必要があります。

- ### b. 新しいノードがグリッドに追加される前にVMコンソールまたはStorageGRIDインストールAPIにアクセスしたり、SSHを使用したりできるように、\*[Temporary installation password]\*ドロップダウンを使用して一時的なインストールパスワードを指定します。



一時インストールパスワードは、ノードのインストール時にのみ使用されます。グリッドに追加されたノードに"**ノードのコンソールパスワード**"は、リカバリパッケージのファイルに含まれているを使用してアクセスできます。 Passwords.txt

- ノード名を使用：\*ノード名\*フィールドに入力した値は、一時的なインストールパスワードとして使用されます。

- カスタムパスワードを使用：カスタムパスワードを一時的なインストールパスワードとして使用します。
  - パスワードを無効にする：一時的なインストールパスワードは使用されません。インストールの問題をデバッグするためにVMにアクセスする必要がある場合は、[を参照してください"インストールに関する問題のトラブルシューティング"](#)。
- c. \*カスタムパスワードを使用\*を選択した場合は、\*カスタムパスワード\*フィールドで使用する一時インストールパスワードを指定します。
- d. \*グリッドネットワーク (eth0) \*セクションで、\*グリッドネットワーク IP 設定\*に静的またはDHCPを選択します。
- 静的を選択した場合は、\*グリッドネットワーク IP\*、\*グリッドネットワークマスク\*、\*グリッドネットワークゲートウェイ\*、\*グリッドネットワーク MTU\*を入力します。
  - DHCPを選択した場合は、\*グリッドネットワーク IP\*、\*グリッドネットワークマスク\*、\*グリッドネットワークゲートウェイ\*が自動的に割り当てられます。
- e. 「\* Primary Admin IP \*」フィールドに、グリッドネットワークのプライマリ管理ノードの IP アドレスを入力します。



この手順は、導入するノードがプライマリ管理ノードの場合は必要ありません。

プライマリ管理ノードの IP アドレスを省略すると、プライマリ管理ノードまたは ADMIN\_IP が設定された少なくとも 1 つのグリッドノードが同じサブネットにある場合は、IP アドレスが自動的に検出されます。ただし、ここでプライマリ管理ノードの IP アドレスを設定することを推奨します。

- a. 「\* Admin Network (eth1) \*」セクションで、「\* Admin network IP configuration \*」に対して「static」、「dhcp」、または「disabled」を選択します。
- 管理ネットワークを使用しない場合は、[DISABLED]を選択し、[Admin Network IP]に「\* 0.0.0.0 \*」と入力します。他のフィールドは空白のままにすることができます。
  - 静的を選択した場合は、\*管理ネットワーク IP\*、\*管理ネットワークマスク\*、\*管理ネットワークゲートウェイ\*、\*管理ネットワーク MTU\*を入力します。
  - 静的を選択した場合は、\*管理ネットワークの外部サブネットリスト\*を入力します。ゲートウェイも設定する必要があります。
  - DHCPを選択した場合は、\*管理ネットワーク IP\*、\*管理ネットワークマスク\*、および\*管理ネットワークゲートウェイ\*が自動的に割り当てられます。
- b. クライアントネットワーク (eth2) \*セクションで、\*クライアントネットワーク IP 構成\*の静的、DHCP、または無効を選択します。
- クライアントネットワークを使用しない場合は、[DISABLED]を選択し、[Client Network IP]に「\* 0.0.0.0 \*」と入力します。他のフィールドは空白のままにすることができます。
  - 静的を選択した場合は、\*クライアントネットワーク IP\*、\*クライアントネットワークマスク\*、\*クライアントネットワークゲートウェイ\*、および\*クライアントネットワーク MTU\*を入力します。
  - DHCPを選択した場合は、\*クライアントネットワーク IP\*、\*クライアントネットワークマスク\*、および\*クライアントネットワークゲートウェイ\*が自動的に割り当てられます。
8. 仮想マシンの設定を確認し、必要な変更を行います。
9. 完了する準備ができたなら、[完了]を選択して仮想マシンのアップロードを開始します。

10. [[step\_recovery\_storage] - リカバリ処理の一環としてこのノードを導入し、フルノードリカバリではない場合は、導入の完了後に次の手順を実行します。
  - a. 仮想マシンを右クリックし、\* 設定の編集 \* を選択します。
  - b. ストレージに指定されている各デフォルト仮想ハードディスクを選択し、\* 削除 \* を選択します。
  - c. データリカバリの状況に応じて、ストレージ要件に従って新しい仮想ディスクを追加し、以前に削除した障害グリッドノードから保存した仮想ハードディスクを再接続するか、またはその両方を実行します。

次の重要なガイドラインに注意してください。

- 新しいディスクを追加する場合は、ノードのリカバリ前に使用していたものと同じタイプのストレージデバイスを使用する必要があります。
- ストレージノードの .ovf ファイルは、ストレージ用の複数の VMDK を定義します。これらの VMDK がストレージ要件を満たしていない場合は、ノードの電源を入れる前に、それらの VMDK を削除し、ストレージに適切な VMDK または RDM を割り当てる必要があります。VMware 環境で一般に使用され、管理も容易であるのは VMDK ですが、大きなオブジェクトサイズ（たとえば 100MB 超）を使用するワークロードのパフォーマンスは RDM の方が高くなります。

11. このノードで使用するポートを再マッピングする必要がある場合は、次の手順を実行します。

ポートの再マッピングが必要となるのは、StorageGRID で使用される 1 つ以上のポートへのアクセスがエンタープライズネットワークポリシーによって制限される場合です。StorageGRID で使用されるポートについては、を参照してください["ネットワークのガイドライン"](#)。



ロードバランサエンドポイントで使用されるポートは再マッピングしないでください。

- a. 新しい VM を選択します。
- b. [構成] タブで、[\* 設定 \* > \* vApp オプション \*] を選択します。vapp Options \* の場所は、vCenter のバージョンによって異なります。
- c. プロパティ \* テーブルで、PORT\_REMAP\_INBOUND および PORT\_REMAP を確認します。
- d. ポートのインバウンド通信とアウトバウンド通信の両方を対称的にマッピングするには、\* PORT\_REMAP \* を選択します。



ポート再マッピングのサポートは非推奨であり、将来のリリースでは削除される予定です。再マップされたポートを削除するには、["ベアメタルホストでのポートの再マッピングを削除します"](#)。



PORT\_REMAP のみを設定すると、インバウンド通信とアウトバウンド通信の両方で環境を指定したマッピングが適用されます。PORT\_REMAP\_INBOUND を併せて指定した場合は、PORT\_REMAP がアウトバウンド通信のみに適用されます。

- i. 「\* 値の設定 \*」を選択します。
- ii. ポートマッピングを入力します。

```
<network type>/<protocol>/<default port used by grid node>/<new port>
```

`<network type>`はgrid、admin、またはclientで、`<protocol>`はtcpまたはudpです。

たとえば、ssh トラフィックをポート 22 からポート 3022 に再マッピングするには、次のように入力します。

```
client/tcp/22/3022
```

カンマで区切ったリストを使用して複数のポートを再マッピングできます。

例：

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

- i. 「\* OK \*」を選択します。
- e. ノードへのインバウンド通信に使用するポートを指定するには、\* port\_remap\_inbound \* を選択します。



PORT\_REMAP\_INBOUNDを指定し、PORT\_REMAPに値を指定しなかった場合、ポートのアウトバウンド通信は変更されません。

- i. 「\* 値の設定 \*」を選択します。
- ii. ポートマッピングを入力します。

```
<network type>/<protocol>/<remapped inbound port>/<default inbound port used by grid node>
```

`<network type>`はgrid、admin、またはclientで、`<protocol>`はtcpまたはudpです。

たとえば、ポート 3022 に送信されるインバウンドの SSH トラフィックを再マッピングしてグリッドノードがポート 22 で受信するようにするには、次のように入力します。

```
client/tcp/3022/22
```

カンマで区切った複数のインバウンドポートを再マッピングできます。

例：

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

- i. 「\* OK」を選択します
12. ノードの CPU またはメモリをデフォルトの設定から増やす場合は、次の手順を実行します。
  - a. 仮想マシンを右クリックし、\* 設定の編集 \* を選択します。
  - b. CPU の数またはメモリの容量を必要に応じて変更します。  
  
[メモリ予約 \*] を、仮想マシンに割り当てられた \* メモリ \* と同じサイズに設定します。
  - c. 「\* OK \*」を選択します。
13. 仮想マシンの電源をオンにします。

終了後

このノードを拡張またはリカバリ手順の一部として導入した場合は、その手順に戻って手順を完了しま

す。

## Linux上のStorageGRIDノードの設定ファイルの例

ここでは、StorageGRID システムで使用するノード構成ファイルを設定する際の参考として、すべてのタイプのグリッドノードのノード構成ファイルの例を示します。



「Linux」は、RHEL、Ubuntu、または Debian のデプロイメントを指します。サポートされているバージョンのリストについては、"[NetApp Interoperability Matrix Tool \(IMT\)](#)"。

ほとんどのノードについては、Grid Manager またはインストール API を使用してグリッドを設定するときに、管理ネットワークとクライアントネットワークのアドレス情報（IP、マスク、ゲートウェイなど）を追加できます。ただし、プライマリ管理ノードは例外です。グリッドの設定を行うためにプライマリ管理ノードの管理ネットワークの IP を参照する必要がある場合（グリッドネットワークがルーティングされていない場合など）は、プライマリ管理ノードのノード構成ファイルで管理ネットワーク接続を設定する必要があります。次の例を参照してください。



ここに示す例では、クライアントネットワークがデフォルトで無効になっていても、クライアントネットワークターゲットがベストプラクティスとして設定されています。

### プライマリ管理ノードの例

ファイル名の例： /etc/storagegrid/nodes/dc1-adm1.conf

• ファイルの内容の例： \*

```
NODE_TYPE = VM_Admin_Node
ADMIN_ROLE = Primary
TEMPORARY_PASSWORD_TYPE = Use custom password
CUSTOM_TEMPORARY_PASSWORD = Passw0rd
BLOCK_DEVICE_VAR_LOCAL = /dev/mapper/dc1-adm1-var-local
BLOCK_DEVICE_AUDIT_LOGS = /dev/mapper/dc1-adm1-audit-logs
BLOCK_DEVICE_TABLES = /dev/mapper/dc1-adm1-tables
GRID_NETWORK_TARGET = bond0.1001
ADMIN_NETWORK_TARGET = bond0.1002
CLIENT_NETWORK_TARGET = bond0.1003

GRID_NETWORK_IP = 10.1.0.2
GRID_NETWORK_MASK = 255.255.255.0
GRID_NETWORK_GATEWAY = 10.1.0.1

ADMIN_NETWORK_CONFIG = STATIC
ADMIN_NETWORK_IP = 192.168.100.2
ADMIN_NETWORK_MASK = 255.255.248.0
ADMIN_NETWORK_GATEWAY = 192.168.100.1
ADMIN_NETWORK_ESL = 192.168.100.0/21,172.16.0.0/21,172.17.0.0/21
```

## ストレージノードの例

ファイル名の例： /etc/storagegrid/nodes/dc1-sn1.conf

- ファイルの内容の例： \*

```
NODE_TYPE = VM_Storage_Node
ADMIN_IP = 10.1.0.2
BLOCK_DEVICE_VAR_LOCAL = /dev/mapper/dc1-sn1-var-local
BLOCK_DEVICE_RANGEDB_00 = /dev/mapper/dc1-sn1-rangedb-0
BLOCK_DEVICE_RANGEDB_01 = /dev/mapper/dc1-sn1-rangedb-1
BLOCK_DEVICE_RANGEDB_02 = /dev/mapper/dc1-sn1-rangedb-2
BLOCK_DEVICE_RANGEDB_03 = /dev/mapper/dc1-sn1-rangedb-3
GRID_NETWORK_TARGET = bond0.1001
ADMIN_NETWORK_TARGET = bond0.1002
CLIENT_NETWORK_TARGET = bond0.1003

GRID_NETWORK_IP = 10.1.0.3
GRID_NETWORK_MASK = 255.255.255.0
GRID_NETWORK_GATEWAY = 10.1.0.1
```

## ゲートウェイノードの例

ファイル名の例： /etc/storagegrid/nodes/dc1-gw1.conf

- ファイルの内容の例： \*

```
NODE_TYPE = VM_API_Gateway
ADMIN_IP = 10.1.0.2
BLOCK_DEVICE_VAR_LOCAL = /dev/mapper/dc1-gw1-var-local
GRID_NETWORK_TARGET = bond0.1001
ADMIN_NETWORK_TARGET = bond0.1002
CLIENT_NETWORK_TARGET = bond0.1003
GRID_NETWORK_IP = 10.1.0.5
GRID_NETWORK_MASK = 255.255.255.0
GRID_NETWORK_GATEWAY = 10.1.0.1
```

## 非プライマリ管理ノードの例

ファイル名の例： /etc/storagegrid/nodes/dc1-adm2.conf

- ファイルの内容の例： \*

```
NODE_TYPE = VM_Admin_Node
ADMIN_ROLE = Non-Primary
ADMIN_IP = 10.1.0.2
BLOCK_DEVICE_VAR_LOCAL = /dev/mapper/dc1-adm2-var-local
BLOCK_DEVICE_AUDIT_LOGS = /dev/mapper/dc1-adm2-audit-logs
BLOCK_DEVICE_TABLES = /dev/mapper/dc1-adm2-tables
GRID_NETWORK_TARGET = bond0.1001
ADMIN_NETWORK_TARGET = bond0.1002
CLIENT_NETWORK_TARGET = bond0.1003

GRID_NETWORK_IP = 10.1.0.6
GRID_NETWORK_MASK = 255.255.255.0
GRID_NETWORK_GATEWAY = 10.1.0.1
```

## Linux での StorageGRID 設定を検証する

StorageGRID ノードごとに構成ファイルをに作成したら /etc/storagegrid/nodes、それらのファイルの内容を検証する必要があります。



「Linux」は、RHEL、Ubuntu、または Debian のデプロイメントを指します。サポートされているバージョンのリストについては、"[NetApp Interoperability Matrix Tool \(IMT\)](#)"。

構成ファイルの内容を検証するには、各ホストで次のコマンドを実行します。

```
sudo storagegrid node validate all
```

ファイルが正しい場合は、次の例に示すように、各構成ファイルの出力に \* PASSED \* と表示されます。



メタデータのみノードでLUNを1つだけ使用している場合は、警告メッセージが表示されても無視してかまいません。

```
Checking for misnamed node configuration files... PASSED
Checking configuration file for node dc1-adm1... PASSED
Checking configuration file for node dc1-gw1... PASSED
Checking configuration file for node dc1-sn1... PASSED
Checking configuration file for node dc1-sn2... PASSED
Checking configuration file for node dc1-sn3... PASSED
Checking for duplication of unique values between nodes... PASSED
```



自動インストールの場合は、コマンドのまたは --quiet オプション `storagegrid (など) storagegrid --quiet...` を使用して、この出力を抑制できます。`-q`。出力を抑制した場合、構成で警告またはエラーが検出されたときはゼロ以外の終了値が返されます。

構成ファイルが正しくない場合、次の例に示すように、問題は \*WARNING\* および \*ERROR\* として表示されます。構成エラーが見つかった場合は、インストールを続行する前に修正する必要があります。

```
Checking for misnamed node configuration files...
WARNING: ignoring /etc/storagegrid/nodes/dc1-adml
WARNING: ignoring /etc/storagegrid/nodes/dc1-sn2.conf.keep
WARNING: ignoring /etc/storagegrid/nodes/my-file.txt
Checking configuration file for node dc1-adml...
ERROR: NODE_TYPE = VM_Foo_Node
      VM_Foo_Node is not a valid node type.  See *.conf.sample
ERROR: ADMIN_ROLE = Foo
      Foo is not a valid admin role.  See *.conf.sample
ERROR: BLOCK_DEVICE_VAR_LOCAL = /dev/mapper/sgws-gw1-var-local
      /dev/mapper/sgws-gw1-var-local is not a valid block device
Checking configuration file for node dc1-gw1...
ERROR: GRID_NETWORK_TARGET = bond0.1001
      bond0.1001 is not a valid interface.  See `ip link show`
ERROR: GRID_NETWORK_IP = 10.1.3
      10.1.3 is not a valid IPv4 address
ERROR: GRID_NETWORK_MASK = 255.248.255.0
      255.248.255.0 is not a valid IPv4 subnet mask
Checking configuration file for node dc1-sn1...
ERROR: GRID_NETWORK_GATEWAY = 10.2.0.1
      10.2.0.1 is not on the local subnet
ERROR: ADMIN_NETWORK_ESL = 192.168.100.0/21,172.16.0foo
      Could not parse subnet list
Checking configuration file for node dc1-sn2... PASSED
Checking configuration file for node dc1-sn3... PASSED
Checking for duplication of unique values between nodes...
ERROR: GRID_NETWORK_IP = 10.1.0.4
      dc1-sn2 and dc1-sn3 have the same GRID_NETWORK_IP
ERROR: BLOCK_DEVICE_VAR_LOCAL = /dev/mapper/sgws-sn2-var-local
      dc1-sn2 and dc1-sn3 have the same BLOCK_DEVICE_VAR_LOCAL
ERROR: BLOCK_DEVICE_RANGEDB_00 = /dev/mapper/sgws-sn2-rangedb-0
      dc1-sn2 and dc1-sn3 have the same BLOCK_DEVICE_RANGEDB_00
```

## LinuxでStorageGRIDホストサービスを開始します

StorageGRID ノードを起動し、ホストのリブート後もノードが再起動されるようにするには、StorageGRID ホストサービスを有効にして開始する必要があります。



「Linux」は、RHEL、Ubuntu、または Debian のデプロイメントを指します。サポートされているバージョンのリストについては、"[NetApp Interoperability Matrix Tool \(IMT\)](#)"。

手順

1. 各ホストで次のコマンドを実行します。

```
sudo systemctl enable storagegrid
sudo systemctl start storagegrid
```

2. 次のコマンドを実行して、導入の進行状況を確認します。

```
sudo storagegrid node status node-name
```

3. いずれかのノードのステータスが「Not Running」または「Stopped」になった場合は、次のコマンドを実行します。

```
sudo storagegrid node start node-name
```

4. StorageGRID ホストサービスを以前に有効にして開始している場合（またはサービスを有効にして開始したかどうか分からない場合）は、次のコマンドも実行します。

```
sudo systemctl reload-or-restart storagegrid
```

## StorageGRID インストール REST API について学ぶ

StorageGRID には、インストールタスクを実行するための StorageGRID インストール API が用意されています。

API のドキュメントは、Swagger オープンソース API プラットフォームで提供されています。Swagger では、ユーザインターフェイスを使用してパラメータやオプションを変更した場合の API の動作を確認しながら、API の開発を進めることができます。このドキュメントは、標準的な Web テクノロジーと JSON データ形式に精通していることを前提としています。



APIドキュメントWebページで実行するAPI処理はすべてライブ処理です。設定データやその他のデータを誤って作成、更新、または削除しないように注意してください。

各 REST API コマンドは、API の URL、HTTP アクション、必須またはオプションの URL パラメータ、および想定される API 応答で構成されます。

### StorageGRID インストール API

StorageGRID インストールAPIは、StorageGRID システムを最初に設定するとき、およびプライマリ管理ノードのリカバリを実行する必要がある場合にのみ使用できます。インストール API には、Grid Manager から HTTPS 経由でアクセスできます。

APIドキュメントにアクセスするには、プライマリ管理ノードでインストールWebページに移動し、メニューバーから\*>[APIドキュメント]\*を選択します。

StorageGRID インストール API には次のセクションがあります。

- `*config *` -- API の製品リリースとバージョンに関連する操作。製品リリースバージョンおよびそのリリースでサポートされる API のメジャーバージョンを一覧表示できます。
- `*grid *` -- グリッドレベルの設定操作。グリッドの詳細、グリッドネットワークのサブネット、グリッドパスワード、NTP および DNS サーバの IP アドレスなど、グリッド設定を取得および更新できます。
- `*nodes *` -- ノードレベルの設定操作。グリッドノードのリストを取得できるほか、グリッドノードの削除、設定、表示、およびグリッドノードの設定のリセットを行うことができます。
- `*provision *` -- プロビジョニング操作。プロビジョニング処理を開始し、プロビジョニング処理のステータスを表示できます。
- `*recovery *` - プライマリ管理ノードのリカバリ処理。情報のリセット、リカバリパッケージのアップロード、リカバリの開始、およびリカバリ処理のステータスの表示を行うことができます。
- `*recovery-package *` -- リカバリパッケージをダウンロードする処理。
- `*sites *` -- サイトレベルの設定操作。サイトを作成、表示、削除、および変更できます。
- `*temporary-password *` -- インストール中に `mgmt-api` を保護するための一時パスワードに対する操作。

## StorageGRID インストールの問題のトラブルシューティング

StorageGRID システムのインストール中に問題が発生した場合は、インストールログファイルにアクセスできます。テクニカルサポートが問題を解決するためにインストールログファイルを使用することもあります。



「Linux」は、RHEL、Ubuntu、または Debian のデプロイメントを指します。サポートされているバージョンのリストについては、"[NetApp Interoperability Matrix Tool \(IMT\)](#)"。

## Linux

次のインストールログファイルは、各ノードを実行しているコンテナからアクセスできます。

- /var/local/log/install.log (すべてのグリッドノードに存在)
- /var/local/log/gdu-server.log (プライマリ管理ノードにあります)

次のインストールログファイルは、ホストからアクセスできます。

- /var/log/storagegrid/daemon.log
- /var/log/storagegrid/nodes/node-name.log

ログファイルへのアクセス方法については、[を参照してください"ログファイルとシステムデータを収集"](#)。

## VMware

次のファイルは、テクニカルサポートが問題の解決に必要とする場合があるメインのインストールログファイルです。

- /var/local/log/install.log (すべてのグリッドノードに存在)
- /var/local/log/gdu-server.log (プライマリ管理ノードにあります)

### 仮想マシンのリソース予約の調整が必要

OVF ファイルでは、各グリッドノードが十分な RAM と CPU を確保して効率よく動作できるようにするためのリソースリザーベーションが設定されています。これらのOVFファイルをVMwareに導入して仮想マシンを作成し、事前定義された数のリソースを使用できない場合、仮想マシンは起動しません。

### タスクの内容

VM ホストに各グリッドノード用の十分なリソースがあることがわかっている場合は、各仮想マシンに割り当てられているリソースを手動で調整し、仮想マシンの起動を試みます。

### 手順

1. VMware vSphere ハイパーバイザーのクライアントツリーで、起動されていない仮想マシンを選択します。
2. 仮想マシンを右クリックし、\* 設定の編集 \* を選択します。
3. [仮想マシンのプロパティ] ウィンドウで、[\* リソース \*] タブを選択します。
4. 仮想マシンに割り当てられているリソースを調整します。
  - a. [CPU] を選択し、[予約] スライダを使用して、この仮想マシン用に予約されている MHz を調整します。
  - b. [\* Memory] を選択し、[Reservation (予約)] スライダを使用してこの仮想マシン用に予約されている MB を調整します。
5. [OK]\*をクリックします。
6. 必要に応じて、同じ VM ホストでホストされている他の仮想マシンに対して同じ手順を繰り返します。

## 一時インストールパスワードが無効になりました

VMwareノードを導入するときに、必要に応じて一時的なインストールパスワードを指定できます。新しいノードがグリッドに追加される前にVMコンソールにアクセスするかSSHを使用するには、このパスワードが必要です。

一時インストールパスワードを無効にした場合は、インストールの問題をデバッグするために追加の手順を実行する必要があります。

次のいずれかを実行できます。

- コンソールにアクセスしたり、SSHを使用してインストールの問題をデバッグできるように、VMを再導入します。ただし、一時的なインストールパスワードを指定してください。
- vCenterを使用してパスワードを設定します。
  - a. VMの電源をオフにします。
  - b. に移動し、[設定]タブを選択して[vApp Options]\*を選択します。
  - c. 設定する一時インストールパスワードのタイプを指定します。
    - カスタム一時パスワードを設定するには、\* custom\_temporary\_password \*を選択します。
    - ノード名を一時パスワードとして使用する場合は、\* temporary\_password\_type \*を選択します。
  - d. 「\* 値の設定 \*」を選択します。
  - e. 一時パスワードを設定します。
    - custom\_temporary\_password \*をカスタムのパスワード値に変更します。
    - temporary\_password\_type を use node name \*の値で更新します。
  - f. VMを再起動して新しいパスワードを適用します。

## 関連情報

- ログファイルへのアクセス方法については、を参照してください"[ログファイル参照](#)"。
- "[StorageGRID システムのトラブルシューティングを行う](#)"
- サポートが必要な場合は、にお問い合わせください "[NetAppサポート](#)"。

## スクリプトの例

### RHEL上のStorageGRIDで4つの物理インターフェイスを1つのボンドに統合する例

以下のサンプルファイルを使用して、4つのLinux物理インターフェイスを1つのLACPボンドにまとめ、3つのVLANインターフェイスを確立して、StorageGRIDのグリッドネットワーク、管理ネットワーク、およびクライアントネットワークのインターフェイス用にボンドを分割します。

## 物理インターフェイス

リンクの反対側のスイッチでも、4つのポートを1つのLACP トランクまたはポートチャネルとして扱い、少なくともタグで参照された3つのVLAN を通過させる必要があります。

### **/etc/sysconfig/network-scripts/ifcfg-ens160**

```
TYPE=Ethernet
NAME=ens160
UUID=011b17dd-642a-4bb9-acae-d71f7e6c8720
DEVICE=ens160
ONBOOT=yes
MASTER=bond0
SLAVE=yes
```

### **/etc/sysconfig/network-scripts/ifcfg-ens192**

```
TYPE=Ethernet
NAME=ens192
UUID=e28eb15f-76de-4e5f-9a01-c9200b58d19c
DEVICE=ens192
ONBOOT=yes
MASTER=bond0
SLAVE=yes
```

### **/etc/sysconfig/network-scripts/ifcfg-ens224**

```
TYPE=Ethernet
NAME=ens224
UUID=b0e3d3ef-7472-4cde-902c-ef4f3248044b
DEVICE=ens224
ONBOOT=yes
MASTER=bond0
SLAVE=yes
```

### **/etc/sysconfig/network-scripts/ifcfg-ens256**

```
TYPE=Ethernet
NAME=ens256
UUID=7cf7aabc-3e4b-43d0-809a-1e2378faa4cd
DEVICE=ens256
ONBOOT=yes
MASTER=bond0
SLAVE=yes
```

## ボンドインターフェイス

**/etc/sysconfig/network-scripts/ifcfg-bond0**

```
DEVICE=bond0
TYPE=Bond
BONDING_MASTER=yes
NAME=bond0
ONBOOT=yes
BONDING_OPTS=mode=802.3ad
```

## VLANインターフェイス

**/etc/sysconfig/network-scripts/ifcfg-bond0.1001**

```
VLAN=yes
TYPE=Vlan
DEVICE=bond0.1001
PHYSDEV=bond0
VLAN_ID=1001
REORDER_HDR=0
BOOTPROTO=none
UUID=296435de-8282-413b-8d33-c4dd40fca24a
ONBOOT=yes
```

**/etc/sysconfig/network-scripts/ifcfg-bond0.1002**

```
VLAN=yes
TYPE=Vlan
DEVICE=bond0.1002
PHYSDEV=bond0
VLAN_ID=1002
REORDER_HDR=0
BOOTPROTO=none
UUID=dbaaec72-0690-491c-973a-57b7dd00c581
ONBOOT=yes
```

### **/etc/sysconfig/network-scripts/ifcfg-bond0.1003**

```
VLAN=yes
TYPE=Vlan
DEVICE=bond0.1003
PHYSDEV=bond0
VLAN_ID=1003
REORDER_HDR=0
BOOTPROTO=none
UUID=d1af4b30-32f5-40b4-8bb9-71a2fbf809a1
ONBOOT=yes
```

## **Ubuntu および Debian 上の StorageGRID の物理インターフェースを集約する例**

この `/etc/network/interfaces` ファイルは、物理インターフェース、ボンドインターフェース、および VLAN インターフェースを定義する 3 つのセクションで構成されています。以下の 3 つのセクションサンプルを 1 つのファイルに統合すれば、4 つの Linux 物理インターフェースを 1 つの LACP ボンドにまとめ、そのボンドを StorageGRID のグリッドネットワーク、管理ネットワーク、およびクライアントネットワークのインターフェースとして使用するための 3 つの VLAN インターフェースを確立できます。

### **物理インターフェース**

リンクの反対側のスイッチでも、4 つのポートを 1 つの LACP トランクまたはポートチャネルとして扱い、少なくともタグで参照された 3 つの VLAN を通過させる必要があります。

```
# loopback interface
auto lo
iface lo inet loopback

# ens160 interface
auto ens160
iface ens160 inet manual
    bond-master bond0
    bond-primary en160

# ens192 interface
auto ens192
iface ens192 inet manual
    bond-master bond0

# ens224 interface
auto ens224
iface ens224 inet manual
    bond-master bond0

# ens256 interface
auto ens256
iface ens256 inet manual
    bond-master bond0
```

## ボンドインターフェイス

```
# bond0 interface
auto bond0
iface bond0 inet manual
    bond-mode 4
    bond-miimon 100
    bond-slaves ens160 ens192 ens224 ens256
```

## VLANインターフェイス

```
# 1001 vlan
auto bond0.1001
iface bond0.1001 inet manual
vlan-raw-device bond0

# 1002 vlan
auto bond0.1002
iface bond0.1002 inet manual
vlan-raw-device bond0

# 1003 vlan
auto bond0.1003
iface bond0.1003 inet manual
vlan-raw-device bond0
```

## 著作権に関する情報

Copyright © 2026 NetApp, Inc. All Rights Reserved. Printed in the U.S.このドキュメントは著作権によって保護されています。著作権所有者の書面による事前承諾がある場合を除き、画像媒体、電子媒体、および写真複写、記録媒体、テープ媒体、電子検索システムへの組み込みを含む機械媒体など、いかなる形式および方法による複製も禁止します。

ネットアップの著作物から派生したソフトウェアは、次に示す使用許諾条項および免責条項の対象となります。

このソフトウェアは、ネットアップによって「現状のまま」提供されています。ネットアップは明示的な保証、または商品性および特定目的に対する適合性の暗示的保証を含み、かつこれに限定されないいかなる暗示的な保証も行いません。ネットアップは、代替品または代替サービスの調達、使用不能、データ損失、利益損失、業務中断を含み、かつこれに限定されない、このソフトウェアの使用により生じたすべての直接的損害、間接的損害、偶発的損害、特別損害、懲罰的損害、必然的損害の発生に対して、損失の発生の可能性が通知されていたとしても、その発生理由、根拠とする責任論、契約の有無、厳格責任、不法行為（過失またはそうでない場合を含む）にかかわらず、一切の責任を負いません。

ネットアップは、ここに記載されているすべての製品に対する変更を随時、予告なく行う権利を保有します。ネットアップによる明示的な書面による合意がある場合を除き、ここに記載されている製品の使用により生じる責任および義務に対して、ネットアップは責任を負いません。この製品の使用または購入は、ネットアップの特許権、商標権、または他の知的所有権に基づくライセンスの供与とはみなされません。

このマニュアルに記載されている製品は、1つ以上の米国特許、その他の国の特許、および出願中の特許によって保護されている場合があります。

権利の制限について：政府による使用、複製、開示は、DFARS 252.227-7013（2014年2月）およびFAR 5252.227-19（2007年12月）のRights in Technical Data -Noncommercial Items（技術データ - 非商用品目に関する諸権利）条項の(b)(3)項、に規定された制限が適用されます。

本書に含まれるデータは商用製品および/または商用サービス（FAR 2.101の定義に基づく）に関係し、データの所有権はNetApp, Inc.にあります。本契約に基づき提供されるすべてのネットアップの技術データおよびコンピュータソフトウェアは、商用目的であり、私費のみで開発されたものです。米国政府は本データに対し、非独占的かつ移転およびサブライセンス不可で、全世界を対象とする取り消し不能の制限付き使用权を有し、本データの提供の根拠となった米国政府契約に関連し、当該契約の裏付けとする場合にのみ本データを使用できます。前述の場合を除き、NetApp, Inc.の書面による許可を事前に得ることなく、本データを使用、開示、転載、改変するほか、上演または展示することはできません。国防総省にかかる米国政府のデータ使用权については、DFARS 252.227-7015(b)項（2014年2月）で定められた権利のみが認められます。

## 商標に関する情報

NetApp、NetAppのロゴ、<http://www.netapp.com/TM>に記載されているマークは、NetApp, Inc.の商標です。その他の会社名と製品名は、それを所有する各社の商標である場合があります。