



# UbuntuまたはDebianへのインストールの計画 と準備

## StorageGRID software

NetApp  
December 03, 2025

# 目次

UbuntuまたはDebianへのインストールの計画と準備	1
必要な情報と資料	1
必要な情報	1
必要な材料	1
StorageGRIDインストールファイルをダウンロードして解凍します。	2
インストール ファイルを手動で検証する (オプション)	4
UbuntuとDebianのソフトウェア要件	5
テストされたPythonのバージョン	6
テストされたPodmanのバージョン	6
テスト済みのDockerバージョン	6
CPUとRAMの要件	7
ストレージとパフォーマンスの要件	8
パフォーマンス要件	8
NetApp ONTAPストレージを使用するホストの要件	9
必要なホストの数	9
各ホストのストレージボリュームの数	9
ホストの最小ストレージ容量	10
例: ホストのストレージ要件の計算	11
ストレージノードのストレージ要件	11
ノードコンテナの移行要件	13
VMware ライブマイグレーションはサポートされていません	13
一貫したネットワークインターフェース名	14
共有ストレージ	14
ホスト (UbuntuまたはDebian) を準備する	15
インストール中にホスト全体の設定がどのように変化するか	15
Linuxをインストールする	17
AppArmorプロファイルのインストールを理解する	18
ホストネットワークを構成する (UbuntuまたはDebian)	19
ホストストレージを構成する	23
コンテナエンジンのストレージボリュームを構成する	27
Dockerをインストールする	27
StorageGRIDホストサービスをインストールする	28

# UbuntuまたはDebianへのインストールの計画と準備

## 必要な情報と資料

StorageGRIDをインストールする前に、必要な情報と資料を収集して準備してください。

### 必要な情報

#### ネットワーク計画

各StorageGRIDノードに接続する予定のネットワーク。StorageGRIDは、トラフィックの分離、セキュリティ、管理の利便性のために複数のネットワークをサポートします。

StorageGRIDをご覧ください"[ネットワークガイドライン](#)"。

#### ネットワーク情報

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

#### グリッドノード用のサーバー

展開を計画しているStorageGRIDノードの数とタイプをサポートするのに十分なリソースを総合的に提供するサーバーセット(物理、仮想、またはその両方)を特定します。



StorageGRIDインストールで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、仮想ストレージエリアネットワーク(vSAN)、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)

理解する"[ノード移行の要件](#)"サービスを中断せずに物理ホスト上でスケジュールされたメンテナンスを実行する場合。

#### 関連情報

["NetApp Interoperability Matrix Tool"](#)

## 必要な材料

### 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。



テストおよび概念実証グリッドに使用できる非本番環境ライセンスは、StorageGRIDインストールアーカイブに含まれています。

### 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、ファイルを抽出します。"](#)

## サービスラップトップ

StorageGRIDシステムは、サービス ラップトップを通じ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（例：PuTTY）
- ["サポートされているウェブブラウザ"](#)

## StorageGRID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解凍します。

StorageGRIDインストール アーカイブをダウンロードし、必要なファイルを抽出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することもできます。

## 手順

1. に行く ["NetApp StorageGRIDのダウンロードページ"](#)。
2. 最新リリースをダウンロードするための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. NetAppアカウントのユーザー名とパスワードでSign in。
4. 注意/必読の文言が表示された場合は、それを読んでチェックボックスを選択してください。



StorageGRIDリリースをインストールした後、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、["回復およびメンテナンス手順の修正プログラム手順"](#)

5. エンドユーザー使用許諾契約書を読み、チェックボックスをオンにして、[同意して続行] を選択します。
6. \* StorageGRIDのインストール \*列で、Ubuntu または Debian の .tgz または .zip インストール アーカイブを選択します。



選択してください。`.zip`サービス ラップトップで Windows を実行している場合は、ファイルが必要です。

7. インストール アーカイブを保存します。
8. 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次の形式を使用します `StorageGRID_<version-number>_Code_Signature_Verification_Package.tar.gz`、どこ `<version-number>` 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抽出します。

10. 必要なファイルを選択してください。

必要なファイルは、計画しているグリッド トポロジとStorageGRIDシステムの展開方法によって異なります。



表にリストされているパスは、抽出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を基準としています。

パスとファイル名	説明
	StorageGRIDダウンロード ファイルに含まれ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。
	テスト環境および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 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 /debs/storagegrid-webscale-images-version-SHA.deb。
	Ubuntu 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。
デプロイメントスクリプトツール	説明
	StorageGRIDシステムの構成を自動化するために使用される Python スクリプト。
	StorageGRIDアプライアンスの構成を自動化するために使用される Python スクリプト。
	シングル サインオンが有効な場合に Grid Management 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 Federate の統合にも使用できます。
	で使用するための設定ファイルの例 `configure-storagegrid.py` スクリプト。

パスとファイル名	説明
	で使用するための空の設定ファイル `configure-storagegrid.py` スクリプト。
	StorageGRIDコンテナのデプロイメント用に Ubuntu または Debian ホストを構成するための Ansible ロールとプレイブックの例。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 Federate 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、Grid Management API にサインインするために使用できるサンプル Python スクリプト。
	コンパニオンによって呼び出されるヘルパー スクリプト `storagegrid-ssoauth-azure.py` Azure との SSO 対話を実行する Python スクリプト。
	StorageGRIDの API スキーマ。  注: アップグレードを実行する前に、アップグレード互換性テスト用の非本番環境のStorageGRID環境がない場合は、これらのスキーマを使用して、StorageGRID管理 API を使用するために作成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。

## インストール ファイルを手動で検証する (オプション)

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 "[NetApp StorageGRIDのダウンロードページ](#)"。

手順

1. 検証パッケージから成果物を抽出します。

```
tar -xf StorageGRID_11.9.0_Code_Signature_Verification_Package.tar.gz
```

2. 次の成果物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`

期待される出力: Leaf-Cert.pem: OK

4. 期限切れのリーフ証明書のためにステップ\_2\_が失敗した場合は、`tsr` 検証するファイル。

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

期待される出力には以下が含まれます: Verification: OK

5. 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`

期待される出力: なし

6. 公開鍵を使用して検証する sha256sum` 訴訟を起こす `sha256sum.sig。

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

期待される出力: Verified OK

7. 確認する `sha256sum` ファイルの内容を新しく作成されたチェックサムと照合します。

例: `sha256sum -c sha256sum`

期待される出力: <filename>: OK

`<filename>` ダウンロードしたアーカイブ ファイルの名前です。

8. "残りの手順を完了する"適切なインストール ファイルを抽出して選択します。

## UbuntuとDebian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1つの仮想マシンが必要です。

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 ランタイムの

インストール オプションを選択した場合に、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、プロバイダーまたは Linux ディストリビューションのサポート ベンダーから入手できる場合は、ここにリストされているバージョンのいずれかをインストールしてください。それ以外の場合は、ベンダーから入手可能な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デプロイメント用のコンテナ エンジンとしての Docker のサポートは非推奨になりました。Docker は将来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。

## テストされたPythonのバージョン

- 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

## テストされたPodmanのバージョン

- 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 サポートは非推奨であり、将来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

## CPUとRAMの要件

StorageGRIDソフトウェアをインストールする前に、ハードウェアを確認して構成し、StorageGRIDシステムをサポートできる状態にしておきます。

各StorageGRIDノードには、次の最小リソースが必要です。

- CPUコア数: ノードあたり8個
- RAM: 利用可能なRAMの合計容量とシステム上で実行されているStorageGRID以外のソフトウェアの量によって異なります。
  - 通常、ノードあたり少なくとも24 GB、システム全体のRAMより2~16 GB少ない
  - 約5,000個のバケットを持つテナントごとに最低64 GB

ソフトウェアベースのメタデータのみのノード リソースは、既存のストレージ ノード リソースと一致する必要があります。例えば：

- 既存のStorageGRIDサイトが SG6000 または SG6100 アプライアンスを使用している場合、ソフトウェアベースのメタデータのみのノードは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
  - Cassandra データベース用の 8 TB SSD または同等のストレージ (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サイトを追加する場合、新しいサイトの合計メタデータ容量は少なくとも既存のStorageGRIDサイトと一致し、新しいサイトのリソースは既存のStorageGRIDサイトのストレージ ノードと一致する必要があります。

各物理ホストまたは仮想ホストで実行する予定のStorageGRIDノードの数が、使用可能な CPU コア数または物理 RAM を超えないようにしてください。ホストがStorageGRID の実行専用でない場合は (非推奨)、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。



CPU とメモリの使用状況を定期的に監視し、これらのリソースが引き続きワークロードに対応できることを確認します。たとえば、仮想ストレージ ノードの RAM と CPU の割り当てを 2 倍にすると、StorageGRID アプライアンス ノードに提供されるものと同様のリソースが提供されます。さらに、ノードあたりのメタデータの量が 500 GB を超える場合は、ノードあたりの RAM を 48 GB 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、メタデータ予約領域設定の増加、CPU とメモリ使用量の監視については、"[管理する](#)"、"[監視](#)"、そして"[アップグレード](#)"StorageGRID。

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になっている場合は、ノードごとに 8 個の仮想コア (4 個の物理コア) を提供できます。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になっていない場合は、ノードごとに 8 つの物理コアを提供する必要があります。

仮想マシンをホストとして使用し、VM のサイズと数を制御できる場合は、StorageGRID ノードごとに 1 つの VM を使用し、それに応じて VM のサイズを調整する必要があります。

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。

参照"[ストレージとパフォーマンスの要件](#)"。

## ストレージとパフォーマンスの要件

初期構成と将来のストレージ拡張をサポートするために十分なスペースを提供できるように、StorageGRID ノードのストレージ要件を理解する必要があります。

StorageGRID ノードには、次の 3 つの論理カテゴリのストレージが必要です。

- コンテナ プール - ノード コンテナのパフォーマンス層 (10K SAS または SSD) ストレージ。これは、StorageGRID ノードをサポートするホストに Docker をインストールして構成するときに、Docker ストレージ ドライバーに割り当てられます。
- システム データ - StorageGRID ホスト サービスが消費し、個々のノードにマップするシステム データとトランザクション ログのノードごとの永続ストレージ用のパフォーマンス層 (10K SAS または SSD) ストレージ。
- 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用のパフォーマンス層 (10K SAS または SSD) ストレージと容量層 (NL-SAS/SATA) バルク ストレージ。

すべてのストレージ カテゴリで RAID 対応ブロック デバイスを使用する必要があります。非冗長ディスク、SSD、または JBOD はサポートされていません。どのストレージ カテゴリでも共有ストレージまたはローカル RAID ストレージを使用できますが、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保存する必要があります。詳細については、以下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、システム データ、およびオブジェクト メタデータに使用される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な影響を与えます。レイテンシ、1 秒あたりの入出力操作 (IOPS)、およびスループットの点で適切なディスク パフォーマンスを確保するには、これらのボリュームにパフォーマンス層 (10K SAS または SSD) のストレージを使用する必要があります。オブジェクト データの

永続ストレージには、容量層 (NL-SAS/SATA) 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 データに使用されるボリュームでは、ライトバック キャッシュが有効になっている必要があります。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上に存在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ます。StorageGRIDノードで使用するボリュームでFabricPoolによる階層化を無効にすることで、トラブルシューティングとストレージの処理がシンプルになります。



FabricPoolを使用してStorageGRIDに関連するデータをStorageGRID自体に階層化することは絶対にしないでください。StorageGRIDにStorageGRIDデータを階層化すると、トラブルシューティングと運用がより複雑になります。

## 必要なホストの数

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



実稼働環境では、単一の物理ホスト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。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 ノードなどの他のタイプのノードは、同じホストにデプロイすることも、必要に応じて専用のホストにデプロイすることもできます。

## 各ホストのストレージボリュームの数

次の表は、各ホストに必要なストレージ ボリューム (LUN) の数と、そのホストに展開されるノードに基づいて、各 LUN に必要な最小サイズを示しています。

テストされた最大 LUN サイズは 39 TB です。



これらの数値はグリッド全体の数値ではなく、ホストごとの数値です。

LUNの用途	ストレージカテゴリ	LUNの数	最小サイズ/LUN
コンテナエンジンストレージプール	コンテナプール	1	ノード総数 × 100 GB
`/var/local` 音量	システムデータ	このホスト上の各ノードに1つ	90 GB

LUNの用途	ストレージカテゴリ	LUNの数	最小サイズ/LUN
ストレージ ノード	オブジェクトデータ	このホスト上のストレージノードごとに3つ  注: 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。少なくとも 3 個のストレージ ボリュームを設定することをお勧めします。	12 TB (4 TB/LUN) 参照 <a href="#">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 参照 <a href="#">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: メタデータのみ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 監査ログ	システムデータ	このホスト上の管理ノードごとに1つ	200 GB
管理ノード テーブル	システムデータ	このホスト上の管理ノードごとに1つ	200 GB



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 では、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できることとなります。

## ホストの最小ストレージ容量

次の表は、各タイプのノードに必要な最小ストレージ容量を示しています。この表を使用すると、ホストにどのノードが展開されるかに基づいて、各ストレージ カテゴリでホストに提供する必要があるストレージの最小量を決定できます。



ディスク スナップショットを使用してグリッド ノードを復元することはできません。代わりに、["グリッドノードの回復"](#)各タイプのノードの手順。

ノードの種類	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	100 GB	90 GB	4,000 GB
管理ノード	100 GB	490 GB (3 LUN)	適用できない

ノードの種類	コンテナプール	システムデータ	オブジェクトデータ
ゲートウェイ ノード	100 GB	90 GB	適用できない

## 例: ホストのストレージ要件の計算

同じホストに 3 つのノード (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) 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 670 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。

ノードの種類	LUNの用途	LUNの数	LUNサイズ
ストレージ ノード	Docker ストレージプール	1	300 GB (100 GB/ノード)
ストレージ ノード	`/var/local` 音量	1	90 GB
ストレージ ノード	オブジェクトデータ	3	12 TB (4 TB/LUN)
管理ノード	`/var/local` 音量	1	90 GB
管理ノード	管理ノード 監査ログ	1	200 GB
管理ノード	管理ノード テーブル	1	200 GB
ゲートウェイ ノード	`/var/local` 音量	1	90 GB
合計		<b>9</b>	コンテナプール: 300 GB システムデータ: 670 GB オブジェクトデータ: 12,000 GB

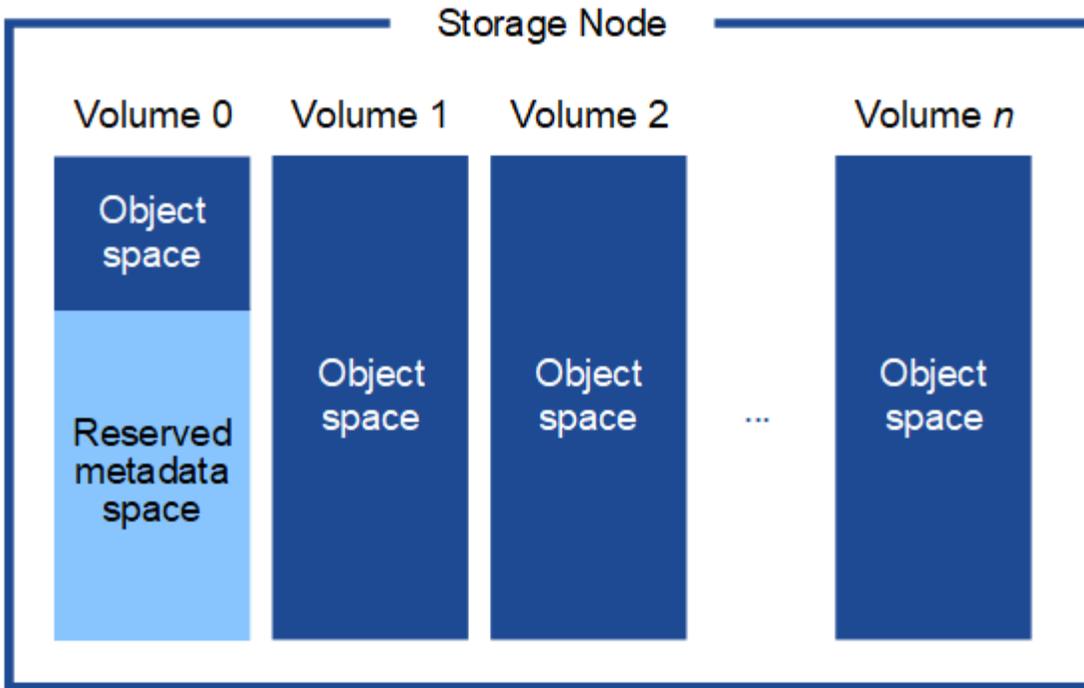
## ストレージノードのストレージ要件

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が、3 個以上のストレージ 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予約します。ストレージ ボリューム 0 およびストレージ ノード内のその他のストレージ ボリュームの残りのスペースは、オブジェクト データ専用で使用されます。



冗長性を提供し、オブジェクト メタデータの損失を防ぐために、StorageGRID はシステム内のすべてのオブジェクトのメタデータのコピーを各サイトに3つ保存します。オブジェクト メタデータの3つのコピーは、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、グリッドにはオブジェクト ストレージ用の最小数のノードも含まれている必要があります。見る["ストレージノードの種類"](#)メタデータのみストレージ ノードの詳細については、こちらをご覧ください。

- 単一サイト グリッドの場合、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 ノードが構成されます。
- マルチサイト グリッドの場合、オブジェクトとメタデータ用にサイトごとに少なくとも1つのストレージ ノードが構成されます。

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ときは、すべてのオブジェクト メタデータのそのノードの部分に十分なスペースがあることを確認する必要があります。

- 少なくとも、ボリューム0には4TBを割り当てる必要があります。



ストレージ ノードに1つのストレージ ボリュームのみを使用し、そのボリュームに4TB以下を割り当てた場合、ストレージ ノードは起動時にストレージ読み取り専用状態になり、オブジェクトのメタデータのみを保存する可能性があります。



ボリューム0 (非本番環境のみ) に500GB未満を割り当てると、ストレージ ボリュームの容量の10%がメタデータ用に予約されます。

- 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 リソースと一致する必要があります。例えば：
  - 既存のStorageGRIDサイトがSG6000またはSG6100アプライアンスを使用している場合、ソフトウェアベースのメタデータのみノードは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
  - Cassandra データベース用の 8 TB SSD または同等のストレージ (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サイトを追加する場合、新しいサイトの合計メタデータ容量は少なくとも既存のStorageGRIDサイトと一致し、新しいサイトのリソースは既存のStorageGRIDサイトのストレージノードと一致する必要があります。

- 新しいシステム (StorageGRID 11.6 以降) をインストールしていて、各ストレージ ノードに 128 GB 以上の RAM がある場合は、ボリューム 0 に 8 TB 以上を割り当てます。ボリューム 0 に大きな値を使用すると、各ストレージ ノード上のメタデータに許可されるスペースを増やすことができます。
- サイトに異なるストレージ ノードを構成する場合は、可能であればボリューム 0 に同じ設定を使用します。サイトに異なるサイズのストレージ ノードが含まれている場合、最も小さいボリューム 0 を持つストレージ ノードによってそのサイトのメタデータ容量が決まります。

詳細については、"[オブジェクトメタデータストレージの管理](#)"。

## ノードコンテナの移行要件

ノード移行機能を使用すると、ノードをあるホストから別のホストに手動で移動できます。通常、両方のホストは同じ物理データセンター内にあります。

ノードの移行により、グリッド操作を中断することなく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 1 つずつ別のホストに移動します。ノードの移行には各ノードの短いダウンタイムのみが必要であり、グリッド サービスの操作や可用性には影響しません。

StorageGRIDノード移行機能を使用する場合は、展開が追加の要件を満たしている必要があります。

- 単一の物理データセンター内のホスト間で一貫したネットワーク インターフェース名
- 単一の物理データセンター内のすべてのホストからアクセスできる、StorageGRIDメタデータおよびオブジェクト リポジトリ ボリュームの共有ストレージ。たとえば、NetApp E シリーズ ストレージ アレイを使用できます。

仮想ホストを使用しており、基盤となるハイパーバイザー層が VM の移行をサポートしている場合は、StorageGRIDのノード移行機能の代わりにこの機能を使用することをお勧めします。この場合、これらの追加要件は無視できます。

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ます。説明書をご覧ください"[グリッドノードのシャットダウン](#)"。

## VMware 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 上でベアメタル インストールを実行すると、OpenStack Live Migration および VMware live vMotion によって仮想マシンのクロック時間がジャンプし、どのタイプのグリッド ノードでもサポートされま

せん。まれではありますが、クロック時刻が正しくないと、データが失われたり、構成が更新されたりする可能性があります。

コールド移行がサポートされています。コールド移行では、ホスト間で移行する前にStorageGRIDノードをシャットダウンします。説明書をご覧ください"[グリッドノードのシャットダウン](#)"。

## 一貫したネットワークインターフェース名

ノードをあるホストから別のホストに移動するには、StorageGRIDホスト サービスが、ノードの現在の場所にある外部ネットワーク接続が新しい場所で複製できることをある程度確信している必要があります。この信頼性は、ホスト内で一貫したネットワーク インターフェース名を使用することで得られます。

たとえば、Host1 上で実行されているStorageGRID NodeA が次のインターフェース マッピングで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ース (それぞれ、グリッド、管理、クライアント ネットワーク インターフェース) に対応します。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 インターフェースに対応しており、同じ物理インターフェース ボンドに從属する 3 つの VLAN インターフェースです。

ここで、NodeA を Host2 に移行するとします。Host2 にも bond0.1001、bond0.1002、bond0.1003 という名前のインターフェースがある場合、システムは、同様の名前のインターフェースが Host2 でも Host1 と同じ接続性を提供するものと想定して、移動を許可します。Host2 に同じ名前のインターフェースがない場合、移動は許可されません。

複数のホスト間で一貫したネットワークインターフェース命名を実現する方法は多数あります。["ホストネットワークを構成する"](#)いくつかの例を示します。

## 共有ストレージ

高速でオーバーヘッドの少ないノード移行を実現するために、StorageGRIDノード移行機能ではノード データを物理的に移動しません。代わりに、ノードの移行は、次のようにエクスポート操作とインポート操作のペアとして実行されます。

### 手順

1. 「ノード エクスポート」操作中に、HostA で実行されているノード コンテナから少量の永続状態データが抽出され、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。次に、HostA 上のノード コンテナのインスタンスが解除されます。
2. 「ノードのインポート」操作中に、HostA で有効だったものと同じネットワーク インターフェースとブロックストレージ マッピングを使用する HostB 上のノード 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。

この動作モードでは、移行が許可され、機能するためには、ノードのすべてのシステム データとオブジェクト

トストレージボリュームが HostA と HostB の両方からアクセスできる必要があります。さらに、HostA と HostB 上の同じ LUN を参照することが保証される名前を使用して、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、DMマルチパスがホスト上で使用されており、エイリアスフィールドが使用されている StorageGRID ストレージノード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。`/etc/multipath.conf` すべてのホストで使用できる一貫性のあるわかりやすいブロックデバイス名を提供します。

```
/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（UbuntuまたはDebian）を準備する

インストール中にホスト全体の設定がどのように変化するか

ベアメタルシステムでは、StorageGRIDはホスト全体にいくつかの変更を加えます。``sysctl`` 設定。

以下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をインストールする

すべての Ubuntu または Debian 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

オペレーティング システムが、以下に示す StorageGRID の最小カーネル バージョン要件を満たしていることを確認してください。コマンドを使用する `uname -r` オペレーティング システムのカーネル バージョンを取得するには、または OS ベンダーに問い合わせてください。

注: Ubuntu バージョン 18.04 および 20.04 のサポートは廃止されており、将来のリリースでは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
18.04.6 (非推奨)	5.4.0-150-ジェネリック	linux-image-5.4.0-150-generic/bionic-updates、bionic-security、現在 5.4.0-150.167~18.04.1
20.04.5 (非推奨)	5.4.0-131-ジェネリック	linux-image-5.4.0-131-generic/focal-updates、現在は5.4.0-131.147
22.04.1	5.15.0-47-ジェネリック	linux-image-5.15.0-47-generic/jammy-updates、jammy-security、現在 5.15.0-47.51
24.04	6.8.0-31-ジェネリック	linux-image-6.8.0-31-generic/noble、現在は6.8.0-31.31

注: Debian バージョン 11 のサポートは非推奨となっており、将来のリリースでは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 (非推奨)	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ーの指示または標準の手順に従って、すべての物理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。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. すべてのホストが Ubuntu または Debian 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. スワップが有効な場合:
  - a. 次のコマンドを実行します。\$ sudo swapoff --all
  - b. すべてのスワップエントリを削除します `etc/fstab` 設定を保持します。



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

## AppArmor プロファイルのインストールを理解する

自己展開の Ubuntu 環境で操作し、AppArmor 強制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 で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、基本オペレーティング システムにインストールするパッケージに対して AppArmor プロファイルが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 の基本パッケージは AppArmor と競合し、他の基本パッケージも競合する可能性があります。

AppArmor プロファイルを処理するには、次の 2 つの選択肢があります。

- StorageGRID システム コンテナ内のパッケージと重複する、ベース システムにインストールされたパッケージの個別のプロファイルを無効にします。個々のプロファイルを無効にすると、AppArmor が有効になっていることを示すエントリが StorageGRID ログ ファイルに表示されます。

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 を完全に無効にします。Ubuntu 9.10 以降の場合は、Ubuntu オンライン コミュニティの指示に従ってください。"[AppArmorを無効にする](#)"。新しいバージョンの Ubuntu では、AppArmor を完全に無効にすることはできない可能性があります。

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が表示されなくなります。

## ホストネットワークを構成する (UbuntuまたはDebian)

ホスト上で Linux のインストールが完了したら、後で展開するStorageGRIDノードにマッピングするのに適したネットワーク インターフェイスのセットを各ホスト上に準備するために、追加の構成を実行する必要がある場合があります。

開始する前に

- あなたは、"[StorageGRIDネットワークガイドライン](#)"。
- 以下の情報を確認しました"[ノードコンテナの移行要件](#)"。
- 仮想ホストを使用している場合は、[MACアドレスの複製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している場合は、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 アダプタにより、特定の Linux ディストリビューションに展開されたStorageGRIDコンテナで接続の問題が発生しました。

### タスク概要

グリッド ノードは、グリッド ネットワークにアクセスできる必要があります。オプションで管理ネットワークとクライアント ネットワークにもアクセスできる必要があります。このアクセスを提供するには、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は、すべてのホスト間での展開を容易にし、移行を可能にするために、フレンドリ名を使用します。

同じインターフェイスをホストと 1 つ以上のノード間で共有できます。たとえば、ホストとノードのメンテナンスを容易にするために、ホスト アクセスとノード管理ネットワーク アクセスに同じインターフェイスを使用できます。ホストと個々のノード間で同じインターフェイスを共有できますが、すべて異なる IP アドレスを持つ必要があります。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。

同じホスト ネットワーク インターフェイスを使用して、ホスト上のすべてのStorageGRIDノードにグリッド ネットワーク インターフェイスを提供することも、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することも、その中間を行うこともできます。ただし、通常は、単一のノードのグリッド ネットワ

ーク インターフェイスと管理ネットワーク インターフェイスの両方に同じホスト ネットワーク インターフェイスを提供したり、1つのノードのグリッド ネットワーク インターフェイスと別のノードのクライアント ネットワーク 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提供したりすることはありません。

このタスクはさまざまな方法で完了できます。たとえば、ホストが仮想マシンであり、ホストごとに1つまたは2つのStorageGRIDノードを展開している場合は、ハイパーバイザーに適切な数のネットワーク 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。実稼働環境で使用するためベアメタル ホストに複数のノードを展開する場合は、フォールト トレランスと帯域幅の共有のために、Linux ネットワーク スタックのVLAN および LACP サポートを活用できます。次のセクションでは、これら2つの例の詳細なアプローチについて説明します。これらの例のいずれかを使用する必要はありません。ニーズを満たす任意のアプローチを使用できます。



ボンドまたは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。これを行うと、コンテナ名前空間内のボンドおよびブリッジデバイスでMACVLANを使用する際のカーネルの問題によって発生するノードの起動が妨げられる可能性があります。代わりに、VLAN または仮想イーサネット (veth) ペアなどの非結合デバイスを使用します。ノード構成ファイルでこのデバイスをネットワーク インターフェイスとして指定します。

## MACアドレスの複製に関する考慮事項と推奨事項

MAC アドレスの複製により、コンテナはホストの MAC アドレスを使用し、ホスト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 アドレスを使用ようになります。無差別モードのネットワーク構成の使用を避けるには、MAC アドレスの複製を使用する必要があります。

### MACクローニングを有効にする

特定の環境では、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 アドレスの複製によってセキュリティを強化できます。コンテナがホスト上の専用 NIC の MAC アドレスを使用するようにすると、プロミスクラス モードのネットワーク構成の使用を回避できます。



MAC アドレスの複製は仮想サーバーのインストールで使用することを目的としており、すべての物理アプライアンス構成で正常に機能するとは限りません。



MAC クローニング対象のインターフェイスがビジー状態であるためにノードの起動に失敗した場合は、ノードを起動する前にリンクを「ダウン」に設定する必要がある場合があります。さらに、仮想環境により、リンクが稼働している間、ネットワーク インターフェイス上の MAC の複製が防止される可能性もあります。インターフェイスがビジー状態であるためにノードが MAC アドレスの設定と起動に失敗した場合は、ノードを起動する前にリンクを「ダウン」に設定すると問題が解決する可能性があります。

MAC アドレスの複製はデフォルトで無効になっており、ノード構成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これを有効にする必要があります。

各ネットワークには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、コンテナはホストのNICのMACアドレスを使用します。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、``_NETWORK_MAC``ノード構成キーの場合は、代わりにそのアドレスが使用されます。ホストとコンテナは常に異なるMACアドレスを持ちます。



ハイパーバイザーでプロミスキャスモードを有効にせずに仮想ホストで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するLinuxホストネットワークが機能しなくなる可能性があります。

## MACクローニングのユースケース

MACクローニングには、考慮すべき2つの使用例があります。

- MACクローンが有効になっていない場合：``_CLONE_MAC``ノード構成ファイル内のキーが設定されていないか、「false」に設定されている場合、ホストはホストNICのMACを使用し、コンテナは、MACがノード構成ファイルで指定されていない限り、StorageGRIDによって生成されたMACを持ちます。``_NETWORK_MAC``鍵。アドレスが``_NETWORK_MAC``キーを指定すると、コンテナのアドレスは``_NETWORK_MAC``鍵。このキー構成では、プロミスキャ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の場合：``_CLONE_MAC``ノード構成ファイルのキーが「true」に設定されている場合、コンテナはホストNICのMACアドレスを使用し、ホストはStorageGRIDが生成したMACアドレスを使用します（MACアドレスがノード構成ファイルで指定されていない場合）。``_NETWORK_MAC``鍵。アドレスが``_NETWORK_MAC``キーの場合、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。このキー構成では、無差別モードを使用しないでください。



MACアドレスの複製を使用せず、ハイパーバイザーによって割り当てられたもの以外のMACアドレスのデータをすべてのインターフェイスで受信および送信できるようにする場合は、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、無差別モード、MACアドレスの変更、および偽造送信に対して承認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される可能性があるため、両方の場所で設定が同じであることを確認してください。

MACクローニングを有効にするには、"[ノード構成ファイルの作成手順](#)"。

## MACクローニングの例

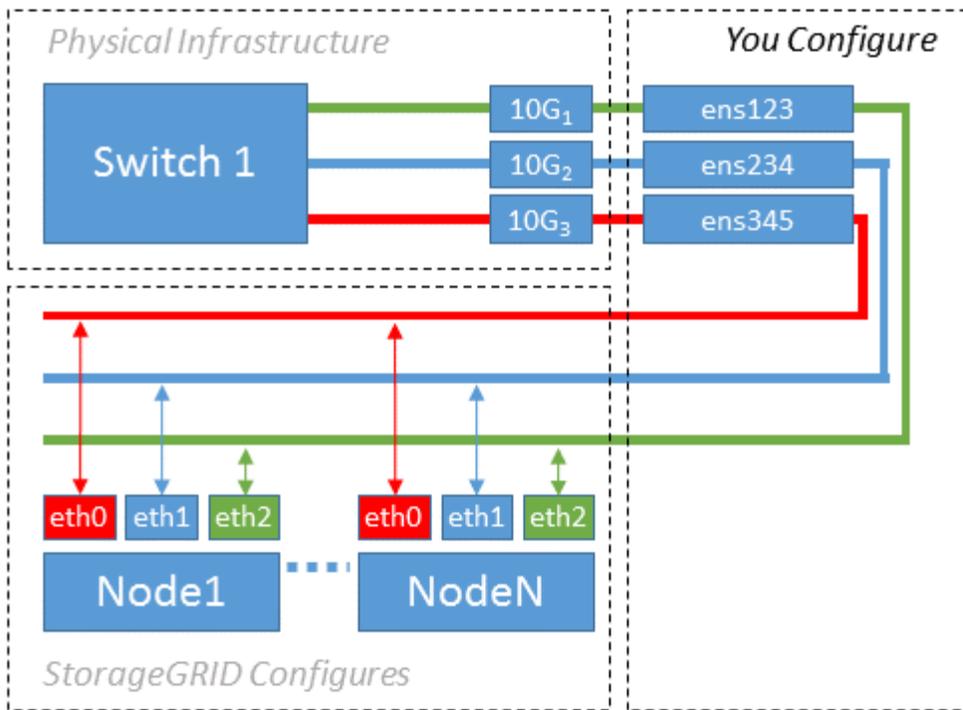
インターフェイス `ens256` のMACアドレス `11:22:33:44:55:66` を持つホストと、ノード構成ファイル内の次のキーを使用してMACクローニングを有効にする例:

- `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、ホスト側の構成をほとんどまたはまったく必要としない単純な物理インターフェイス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、インストール時または起動時、あるいはインターフェイスがホットアドされたときに、ensXYZ インターフェイスを自動的に作成します。起動後にインターフェイ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。

図には複数の StorageGRID ノードが表示されていますが、通常はこの構成は単一ノードの VM に使用されません。

スイッチ 1 が物理スイッチの場合、インターフェイス 10G<sub>1</sub> から 10G<sub>3</sub> に接続されている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結合によるVLANの伝送

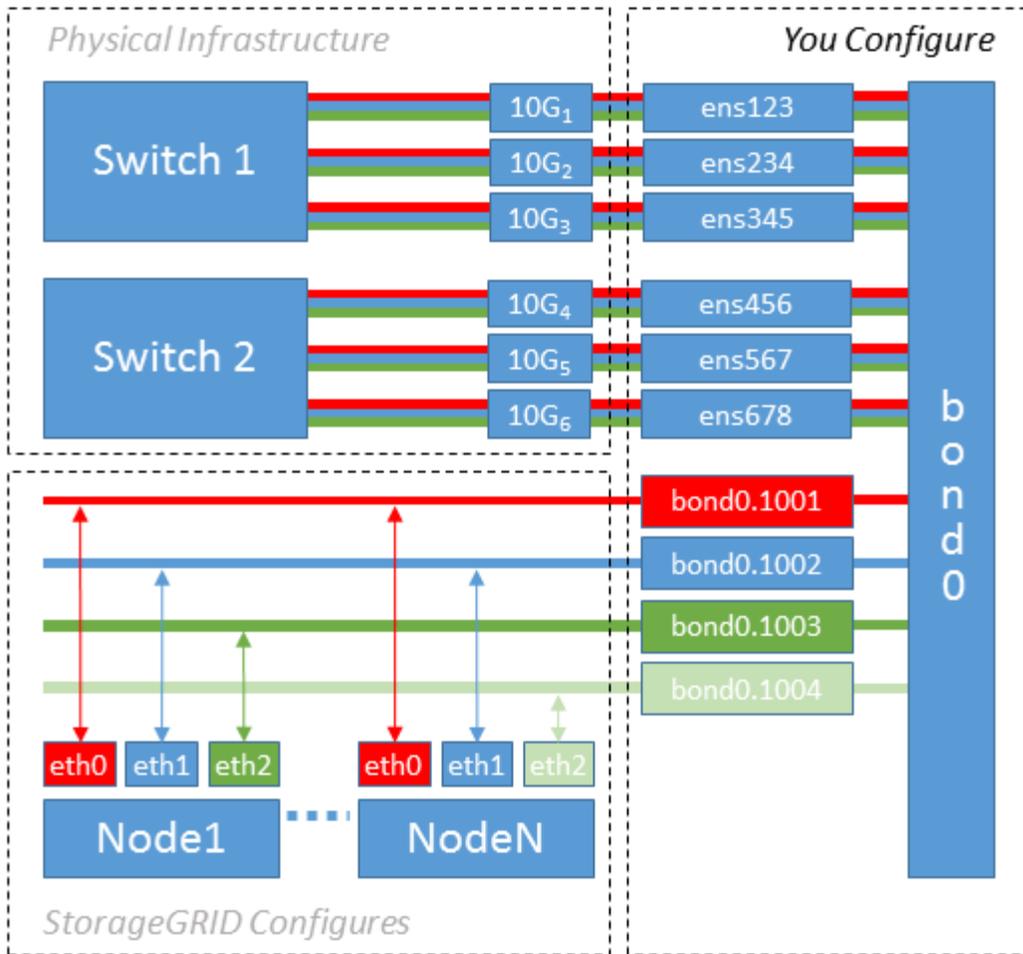
例 2 では、ネットワーク インターフェイスのボンディングと、使用している Linux ディストリビューションでの VLAN インターフェイスの作成に精通していることを前提としています。

### タスク概要

例 2 では、単一のホスト上のすべてのノード間で利用可能なすべてのネットワーク帯域幅の共有を容易にする、汎用的で柔軟な VLAN ベースのスキームについて説明します。この例は、ベアメタル ホストに特に当てはまります。

この例を理解するために、各データセンターにグリッド、管理、クライアント ネットワークの 3 つの個別のサブネットがあると仮定します。サブネットは個別の VLAN (1001、1002、および 1003) 上にあり、LACP 結合トランク ポート (bond0) 上のホストに提示されます。ボンドには、bond0.1001、bond0.1002、bond0.1003 の 3 つの VLAN インターフェイスを設定します。

同じホスト上のノード ネットワークに個別の VLAN とサブネットが必要な場合は、ボンドに VLAN インターフェイスを追加し、それらをホストにマップできます (図では bond0.1004 として表示)。



## 手順

1. StorageGRIDネットワーク接続に使用され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 ボンドに集約します。

すべてのホスト上の結合に同じ名前を使用します (例: bond0)。

2. 標準のVLANインターフェイス命名規則を使用して、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作成します。 `physdev-name.VLAN ID`。

手順 1 と 2 では、ネットワーク リンクのもう一方の端を終端するエッジ スイッチで適切な構成が必要であることを注意してください。エッジ スイッチ ポートも LACP ポート チャネルに集約され、トランクとして設定され、必要なすべての VLAN を通過できるように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のサンプル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

["/etc/network/interfacesの例"](#)

## ホストストレージを構成する

各ホストにブロック 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

このタスクを実行するために必要な情報を提供する次のトピックを確認しました。

- ["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"](#)

タスク概要

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、次の事項を決定します。

- 各ホストに必要なボリュームの数（そのホストに展開されるノードの数とタイプに基づく）
- 各ボリュームのストレージ カテゴリ (システム データまたはオブジェクト データ)
- 各巻のサイズ

ホストにStorageGRIDノードを展開するとき、この情報と、Linux によって各物理ボリュームに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ン分割したり、フォーマットしたり、マウントしたりする必要はありません。ホストから見えるようにするだけです。



メタデータのみストレージ ノードには、オブジェクト データ LUN が 1 つだけ必要です。

「raw」特殊デバイスファイルの使用を避ける(`/dev/sdb`ボリューム名のリストを作成するときに、たとえば、これらのファイルはホストの再起動によって変更される可能性があり、システムの適切な動作に影響を与えます。iSCSI LUNとデバスマッパーマルチパスを使用している場合は、`/dev/mapper`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、ディレクトリにこの情報を格納する必要があります。あるいは、システムによって作成されたソフトリンクを使用することもできます。`/dev/disk/by-path/`永続的なデバイス名用。

例えば：

```
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。

これらの各ブロックストレージボリュームにわかりやすい名前を割り当てると、StorageGRIDの初期インストールと将来のメンテナンス手順が簡素化されます。共有ストレージボリュームへの冗長アクセスにデバイス Mapper マルチパスドライバを使用している場合は、`alias` あなたのフィールド `/etc/multipath.conf` ファイル。

例えば：

```

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
    }
    ...
}

```

このようにエイリアスフィールドを使用すると、エイリアスはブロックデバイスとして表示されます。`/dev/mapper` ホスト上のディレクトリに保存され、構成またはメンテナンス操作でブロック ストレージ ボリュームの指定が必要なときに、わかりやすく簡単に検証できる名前を指定できます。

StorageGRIDノードの移行をサポートするために共有ストレージを設定し、デバイスマッパーマルチパスを使用している場合は、共通の `/etc/multipath.conf` 同じ場所にあるすべてのホストで。各ホストで異なる Docker ストレージ ボリュームを使用するようにしてください。エイリアスを使用し、各 Docker ストレージ ボリューム LUN のエイリアスにターゲット ホスト名を含めると、覚えやすくなるため、推奨されます。



ソフトウェアのみのデプロイメント用のコンテナ エンジンとしての Docker のサポートは非推奨になりました。Docker は将来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

- ["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"

## コンテナエンジンのストレージボリュームを構成する

コンテナ エンジン (Docker または Podman) 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デプロイメント用のコンテナ エンジンとしての Docker のサポートは非推奨になりました。Docker は将来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。

### タスク概要

Dockerストレージボリュームにローカルストレージを使用する予定で、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。 /var/lib。

### 手順

1. Docker 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。

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

2. Docker ストレージ ボリュームをマウントします。

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

3. docker-storage-volume-device のエントリを /etc/fstab に追加します。

この手順により、ホストの再起動後にストレージ ボリュームが自動的に再マウントされるようになります。

## Dockerをインストールする

StorageGRIDシステムは、Docker コンテナのコレクションとして Linux 上で実行されます。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。



ソフトウェアのみのデプロイメント用のコンテナ エンジンとしての Docker のサポートは非推奨になりました。Docker は将来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. 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。



Linux ディストリビューションに Docker が含まれていない場合は、Docker 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 が有効になっていて起動していることを確認します。

```
sudo systemctl enable docker
```

```
sudo systemctl start docker
```

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

```
sudo docker version
```

クライアントとサーバーのバージョンは 1.11.0 以降である必要があります。

#### 関連情報

["ホストストレージを構成する"](#)

## StorageGRIDホストサービスをインストールする

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

#### タスク概要

これらの手順では、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。代わりに、インストール アーカイブに含まれる 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。Linux オペレーティング システムの APT リポジトリの説明を参照してください。

#### 手順

1. StorageGRID DEB パッケージを各ホストにコピーするか、共有ストレージで使用できるようにします。

例えば、`/tmp`ディレクトリに保存しておく、次の手順でサンプル コマンドを使用できるようになります。

2. 各ホストに root として、または sudo 権限を持つアカウントを使用してログインし、次のコマンドを実行します。

インストールする必要があります `images` 最初にパッケージを、そして `service` パッケージ2。パッケージを以下のディレクトリ以外の場所に置いた場合 `tmp` 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 2.7 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。

## 著作権に関する情報

Copyright © 2025 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。