ホスト（**Linux**）を準備する

StorageGRID software

NetApp
July 23, 2026

目次

ホスト（Linux）を準備する

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 はホスト全体の `sysctl`設定にいくつかの変更を加えます。

> 「Linux」とは、RHEL、Ubuntu、またはDebianのいずれかの環境を指します。サポートされているバージョンの一覧については、 "NetApp Interoperability Matrix Tool（IMT）"を参照してください。

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 グリッドホストに StorageGRID をインストールします

StorageGRID をすべての Linux グリッドホストに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 Interoperability Matrix Tool 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 のカーネル最小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 `uname -r`を使用して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 ベンダーにお問い合わせください。

「Linux」とは、RHEL、Ubuntu、またはDebianのいずれかの環境を指します。サポートされているバージョンの一覧については、 "NetApp Interoperability Matrix Tool（IMT）"を参照してください。

RHEL

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

Ubuntu

注**:** Ubuntu バージョン 18.04 および 20.04 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。

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-generic	linux-image-5.15.0-47-generic/jammy-updates,jammy-security,now 5.15.0-47.51
24.04	6.8.0-31-generic	linux-image-6.8.0-31-generic/noble,now 6.8.0-31.31

Debian

注： 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（非推奨）	5.10.0-18-amd64	linux-image-5.10.0-18-amd64/stable,now 5.10.150-1
12	6.1.0-9-amd64	linux-image-6.1.0-9-amd64/stable,now 6.1.27-1

手順

1. 販売代理店の指示または標準手順に従って、すべての物理または仮想グリッドホストに Linux をインストールしてください。

グラフィカルなデスクトップ環境はインストールしないでください。

- RHELをインストールする際に標準のLinuxインストーラーを使用する場合は、利用可能であれば「compute node」ソフトウェア構成を選択するか、「minimal install」ベース環境を選択してください。
- Ubuntuをインストールする際は、必ず **standard system utilities** を選択してください。UbuntuホストへのSSHアクセスを有効にするには、**OpenSSH server** 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クリアのままで構いません。

2. すべてのホストがパッケージリポジトリ（RHELの場合はExtrasチャネルを含む）にアクセスできることを確認してください。
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. すべてのスワップエントリを `/etc/fstab`から削除して、設定を永続化します。

> スワップを完全に無効にしないと、パフォーマンスが著しく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

自己展開した Ubuntu 環境で操作していて、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 プロファイルは、StorageGRID ととも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、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用にインストールされます。これらのパッケージを StorageGRID システムコンテナから実行すると、AppArmorプロファイルはブロックされます。DHCP、MySQL、NTP、および 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 プロファイルの処理方法には、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ているパッケージのうち、StorageGRID システムコンテナ内のパッケージと重複するものについては、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、StorageGRID ログファイルに AppArmor が有効になっていることを示すエントリが表示されます。

  以下のコマンドを使用してください：

```
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手順に従ってください： "AppArmorを無効にする"。AppArmorを完全に無効にすることは、新しいバージョンのUbuntuでは不可能な場合があります。

  AppArmor を無効にすると、AppArmor が有効であることを示すエントリは StorageGRID ログファイルに表示されなくなります。

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 インストールが完了したら、後でデプロイする StorageGRID ノードへのマッピング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で準備するために、追加の設定が必要になる場合があります。

「Linux」とは、RHEL、Ubuntu、またはDebianのいずれかの環境を指します。サポートされているバージョンの一覧については、 "NetApp Interoperability Matrix Tool（IMT）"を参照してください。

開始する前に

- "StorageGRID ネットワーク構築ガイドライン"をレビューしました。
- "ノードコンテナ移行要件"に関する情報を確認しました。
- 仮想ホストを使用している場合は、ホストネットワークを設定する前にMACアドレスのクローン作成に関する考慮事項と推奨事項をお読みください。

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ディストリビューションにデプロイされたStorageGRIDコンテナで接続の問題を引き起こしています。

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、オプションで管理者ネットワークおよび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。このアクセス権限は、ホストの物理インターフェースを各グリッドノードの仮想インターフェースに関連付けるマッピングを作成することによって付与されます。ホストインターフェースを作成する際は、すべてのホストへの展開を容易にし、移行を可能にするために、分かりやすい名前を使用してください。

ホストと1つまたは複数のノード間で、同じインターフェースを共有できます。例えば、ホストへのアクセスとノードの管理ネットワークへのアクセスに同じインターフェースを使用することで、ホストとノードのメンテナンスを容易にすることができます。ホストと個々のノード間で同じインターフェースを共有することは可能ですが、すべてのノードは異なる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することはできません。

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、その中間的な方法をとることもできます。ただし、通常は、単一のノードに対して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として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、ある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として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することはありません。

このタスクは様々な方法で完了できます。例えば、ホストが仮想マシンで、各ホストに1つまたは2つのStorageGRID ノードをデプロイする場合は、ハイパーバイザーで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。本番環境で複数のノードをベアメタルホスト上にデプロイする場合は、Linux ネットワークスタックの VLAN および LACP のサポートを活用して、耐障害性と帯域幅共有を実現できます。以下のセクションでは、これら2つの例について詳細なアプローチを説明します。これらの例のいずれかを必ず使用する必要はありません。ご自身のニーズに合った方法であれば、どのような方法でも使用できます。

> ボンドデバイスやブリッジデバイスをコンテナのネットワークインターフェースとして直接使用しないでください。コンテナ名前空間内でボンドデバイスおよびブリッジデバイスとともにMACVLAN を使用する際のカーネルの問題により、ノードが起動できなくなる可能性があります。代わりに、VLAN や仮想イーサネット（veth）ペアなどの非ボンドデバイスを使用してください。ノード設定ファイルで、このデバイスをネットワークインターフェースとして指定してください。

MACアドレスのクローン作成に関する考慮事項と推奨事項

MACアドレスのクローン作成を行うと、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アドレスを使用するようになります。プロミスキャスモードのネットワーク設定を避けるためには、MACアドレスのクローニング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

特定の環境においては、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。これは、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それぞれ専用の仮想NICを使用できるためです。コンテナにホスト上の専用NICのMACアドレスを使用させることで、プロミスキャスモードのネットワーク構成を使用する必要がなくなります。

> MACアドレスのクローン作成は、仮想サーバー環境での使用を想定しており、すべての物理アプライアンス構成で正常に動作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ースがビジー状態であるためにノードが起動しない場合は、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かもしれません。さらに、仮想環境によっては、リンクが確立されている間、ネットワークインターフェース上でのMACクローニングが防止される可能性もあります。インターフェースがビジー状態のためノードがMACアドレスを設定できず起動できない場合、ノードを起動する前にリンクを「ダウン」に設定することで問題が解決する可能性があります。

MACアドレスのクローン作成はデフォルトでは無効になっており、ノード構成キーで設定する必要があります。StorageGRID のインストール時に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アドレスを使用するようになります。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`ノード構成キーを使用して設定した場合は、そのアドレスが代わりに使用されます。ホストとコンテナは常に異なるMACアドレスを持ちます。

> 仮想ホストで MAC クローニングを有効にし、同時にハイパーバイザーでプロミスキャスモードを有効にしない場合、ホストのインターフェースを使用した Linux ホストのネットワーク接続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つのユースケースがあります。

- MAC クローニングが有効になっていません：ノード構成ファイルの `_CLONE_MAC`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 MAC を使用し、コンテナは StorageGRID によって生成された MAC を持ちます。ただし、 `_NETWORK_MAC`キーに MAC が指定されている場合を除きます。 `_NETWORK_MAC`キーにアドレスが設定されている場合、コンテナには `_NETWORK_MAC`キーで指定されたアドレスが設定されます。このキー構成では、プロミスキャスモードの使用が必要です。
- MAC クローニングが有効な場合：ノード構成ファイルの `_CLONE_MAC`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 を使用し、 `_NETWORK_MAC`キーに MAC が指定されていない限り、ホストは StorageGRID が生成した MAC を使用します。 `_NETWORK_MAC`キーにアドレスが設定されている場合、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、プロミスキャスモードを使用しないでください。

> MAC アドレスの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 アドレスに対してすべてのインターフェースがデータを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、プロミスキャスモード、MAC アドレスの変更、および偽造送信に対して **Accept** に設定されていることを確認してください。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を作成するための手順"を参照してください。

MACクローン作成例

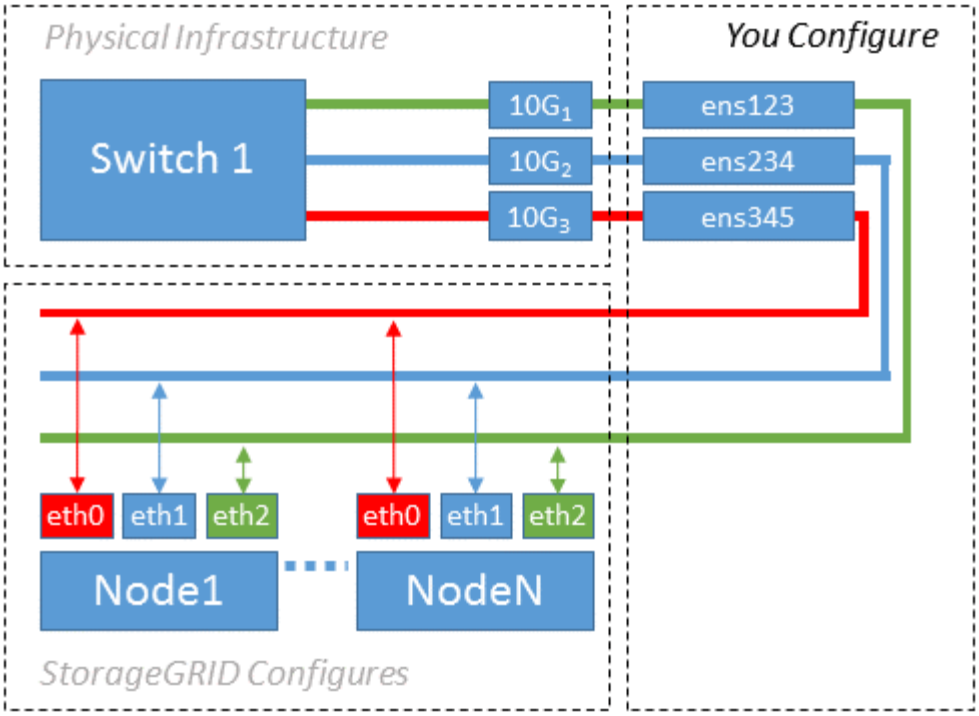
インターフェース ens256 の MAC アドレスが 11:22:33:44:55:66 であるホストと、ノード設定ファイルに以下のキーがある場合の MAC クローニングの有効化例：

- `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**マッピング

例 1 では、ホスト側の設定がほとんど、あるいはまったく必要ない、シンプルな物理インターフェー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、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際に、 `ensXYZ`インターフェースを自動的に作成します。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すること以外に、特別な設定は必要ありません。どの ensXYZ がどの StorageGRID ネットワーク（Grid、Admin、または Client）に対応するかを確認しておく必要があります。これにより、設定プロセスの後半で正しいマッピングを指定できます。

図には複数の StorageGRID ノードが示されていますが、通常はこの構成をシングルノードの仮想マシンに使用します。

スイッチ1が物理スイッチである場合は、インターフェース10G1～10G3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ボンディングによるVLANの伝送

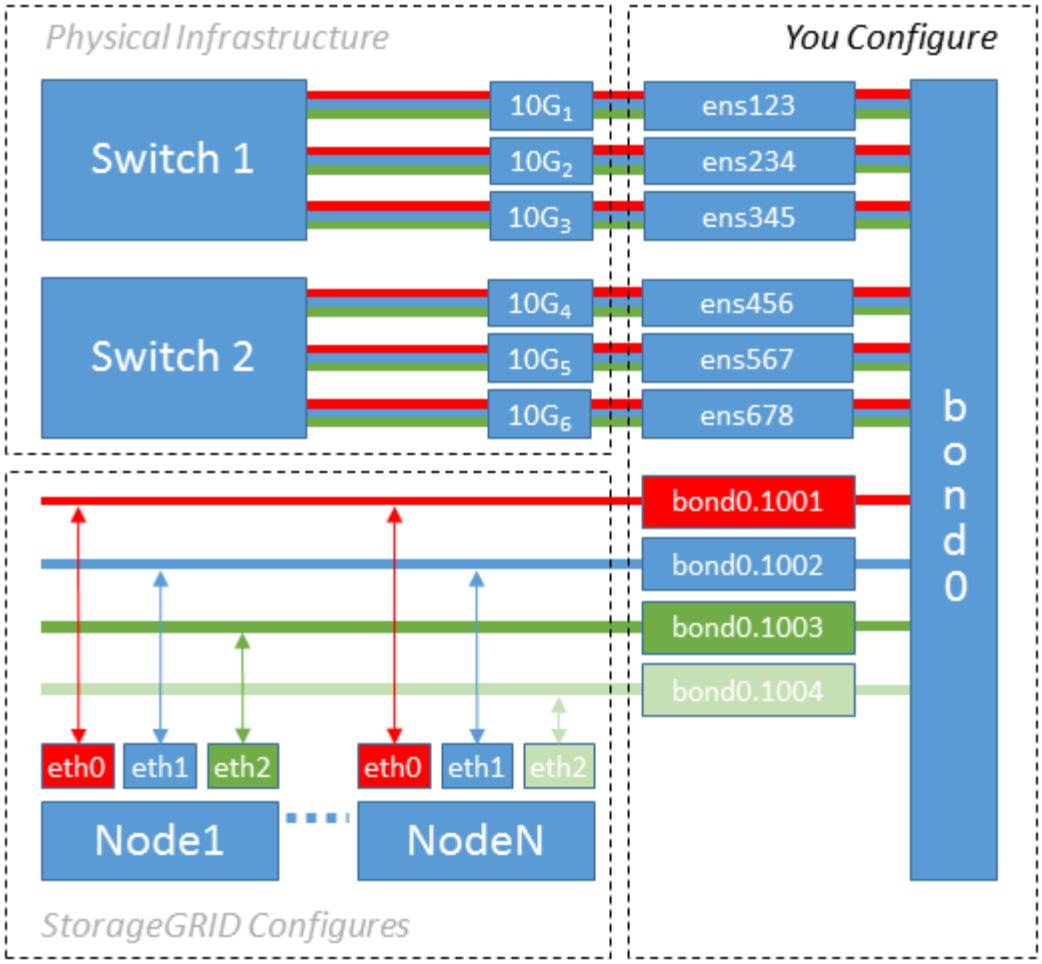
例2では、ネットワークインターフェースのボンディングと、使用しているLinuxディストリビューションでのVLANインターフェースの作成について理解していることを前提としています。

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を共有することを可能にする、汎用的で柔軟なVLANベースの方式について説明します。この例は、特にベアメタルホストに当てはまります。

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の3つの独立したサブネットがあると仮定します。サブネットはそれぞれ別の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）を介してホストに提示されます。ボンド上に、bond0.1001、bond0.1002、bond0.1003という3つのVLANインターフェースを設定します。

同一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、ボンディング上にVLANインターフェイスを追加し、それらをホストにマッピングできます（図ではbond0.1004として示されています）。



手順

1. StorageGRID ネットワーク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 ボンドに集約します。

   すべてのホストで同じ名前をボンドに使用します。たとえば、 `bond0`。

2. このボンドを関連付けられた「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 `physdev-name.VLAN ID`を使用して作成します。

   手順1と手順2を実行するには、ネットワークリンクのもう一方の端を接続するエッジ スイッチに適切な設定が必要となることに注意してください。エッジ スイッチポートもLACPポートチャネルに集約し、トランクとして設定して、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方式に対応した、インターフェース構成ファイルのサンプルが提供されています。

関連情報

- "UbuntuおよびDebianにおける/etc/network/interfacesの例"
- "RHEL 用の /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

このタスクを完了するために必要な情報を提供する以下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"
- "ノードコンテナ移行の要件"

「Linux」とは、RHEL、Ubuntu、またはDebianのいずれかの環境を指します。サポートされているバージョンの一覧については、 "NetApp Interoperability Matrix Tool（IMT）"を参照してください。

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際は、「ストレージ要件」の表を使用して、以下の項目を決定します。

- 各ホストに必要なボリューム数（そのホストにデプロイされるノードの数と種類に基づく）
- 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）
- 各ボリュームのサイズ

この情報と、Linux が各物理ボリュームに割り当てた永続名は、ホスト上に StorageGRID ノードをデプロイする際に使用します。

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十分です。

メタデータ専用のストレージノードでは、オブジェクトデータ用のLUNは1つだけで済みます。

ボリューム名のリストを作成する際に、「生」の特殊デバイスファイル（(`/dev/sdb`など）の使用は避けてください。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、システムの正常な動作に影響を与えることがあります。iSCSI LUN とデバイスマッパーマルチパスを使用している場合は、特に SAN トポロジー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、 `/dev/mapper`ディレクトリでマルチパスエイリアスの使用を検討してください。または、永続的なデバイス名として `/dev/disk/by-path/`配下のシステム作成ソフトリンクを使用することもできます。

次に例を示します。

```
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それぞれにわかりやすい名前を割り当てることで、StorageGRID の初期インストールおよび今後のメンテナンス手順を簡素化できます。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マッパーマルチパスドライバーを使用している場合は、 `alias`ファイルの `/etc/multipath.conf`フィールドを使用できます。

次に例を示します。

```
multipaths {
     multipath {
          wwid 3600a09800059d6df00005df2573c2c30
          alias docker-storage-volume-hostA
     }
     multipath {
          wwid 3600a09800059d6df00005df3573c2c30
          alias sgws-adm1-var-local
     }
     multipath {
          wwid 3600a09800059d6df00005df4573c2c30
          alias sgws-adm1-audit-logs
     }
     multipath {
          wwid 3600a09800059d6df00005df5573c2c30
          alias sgws-adm1-tables
     }
     multipath {
          wwid 3600a09800059d6df00005df6573c2c30
          alias sgws-gw1-var-local
     }
     multipath {
          wwid 3600a09800059d6df00005df7573c2c30
          alias sgws-sn1-var-local
     }
     multipath {
          wwid 3600a09800059d6df00005df7573c2c30
          alias sgws-sn1-rangedb-0
     }
     …
```

このようにエイリアスフィールドを使用すると、エイリアスがホスト上の `/dev/mapper`ディレクトリにブロックデバイスとして表示されるため、構成やメンテナンス操作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場合に、わかりやすく検証しやすい名前を指定できます。

共有ストレージをセットアップして StorageGRID ノードの移行をサポートし、Device Mapper Multipathing を使用する場合は、同じ場所に配置されているすべてのホストに共通の `/etc/multipath.conf`を作成してインストールできます。各ホストで異なるコンテナエンジンの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、各コンテナエンジンの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ことで、覚えやすくなるため、この方法を推奨します。

ソフトウェアのみのデプロイメントにおけるコンテナエンジンとしての Docker のサポートは非推奨となりました。今後のリリースでは、Docker は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する"
- "ストレージとパフォーマンスの要件"
- "ノードコンテナ移行の要件"

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デプロイメントにおけるコンテナエンジンとしての Docker のサポートは非推奨となりました。今後のリリースでは、Docker は別のコンテナエンジンに置き換えられる予定です。

「Linux」とは、RHEL、Ubuntu、またはDebianのいずれかの環境を指します。サポートされているバージョンの一覧については、 "NetApp Interoperability Matrix Tool（IMT）"を参照してください。

タスク概要

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、次の内容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。

- Podman： `/var/lib/containers`
- Docker： `/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上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向け：

```
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向け：

```
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ます。

Dockerをインストールする

StorageGRID システムは、コンテナの集合体として Linux 上で動作させることができます。

- Ubuntu または Debian に 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、以下の手順に従ってDockerをインストールしてください。それ以外の場合は、Podmanをインストールする。

> ソフトウェアのみのデプロイメントにおけるコンテナエンジンとしての Docker のサポートは非推奨となりました。今後のリリースでは、Docker は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. お使いのLinuxディストリビューションの手順に従って、Dockerをインストールしてください。

   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のウェブサイトからダウンロードできます。

2. 以下の2つのコマンドを実行して、Dockerが有効になり起動していることを確認してください。

   ```
   sudo systemctl enable docker
   ```

   ```
   sudo systemctl start docker
   ```

3. 以下のコマンドを入力して、想定されるバージョンの Docker がインストールされていることを確認してください。

   ```
   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。

Podmanをインストールする

StorageGRIDシステムはコンテナの集合体として動作します。Podmanコンテナエンジンを使用する場合は、以下の手順に従ってPodmanをインストールしてください。それ以外の場合は、Dockerをインストールする。

手順

1. お使いのLinuxディストリビューションの手順に従って、PodmanとPodman-Dockerをインストールしてください。

   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. 以下のコマンドを入力して、PodmanとPodman-Dockerの想定されるバージョンがインストールされていることを確認してください。

   ```
   sudo docker version
   ```

   Podman-Dockerパッケージを使用すると、Dockerコマンドを使用できます。

   クライアントとサーバーのバージョンは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構成する"

LinuxへのStorageGRIDホストサービスのインストール

お使いのオペレーティングシステムタイプに対応した StorageGRID パッケージを使用して、StorageGRID ホストサービスをインストールします。

「Linux」とは、RHEL、Ubuntu、またはDebianのいずれかの環境を指します。サポートされているバージョンの一覧については、 "NetApp Interoperability Matrix Tool（IMT）"を参照してください。

RHEL

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

タスク概要

これらの手順では、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。代替手段として、インストールアーカイブに含まれているDNFリポジトリの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。お使いのLinuxオペレーティングシステムに対応するDNFリポジトリの手順を参照してください。

手順

1. StorageGRID RPM パッケージを各ホストにコピーするか、共有ストレージ上で利用できるようにしてください。

   例えば、 `/tmp`ディレクトリに配置すると、次のステップでサンプルコマンドを使用できます。

2. 各ホストにrootユーザーとして、またはsudo権限を持つアカウントでログインし、以下のコマンドを記載されている順序で実行してください。

```
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 `/tmp`以外のディレクトリに配置した場合は、使用したパスに合わせてコマンドを修正してください。

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。代替手段として、インストールアーカイブに含まれているAPTリポジトリの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。お使いのLinuxオペレーティングシステムに対応するAPTリポジトリの手順を参照してください。

手順

1. StorageGRID の DEB パッケージを各ホストにコピーするか、共有ストレージ上で利用できるようにしてください。

   例えば、 `/tmp`ディレクトリに配置すると、次のステップでサンプルコマンドを使用できます。

2. 各ホストにrootユーザーとして、またはsudo権限を持つアカウントでログインし、以下のコマンドを実行してください。

   最初に `images`パッケージをインストールし、次に `service`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 `/tmp`以外のディレクトリに配置した場合は、使用したパスに合わせてコマンドを修正してください。

   ```
   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。 `sudo dpkg --install /tmp/storagegrid-webscale-images-version-SHA.deb`コマンドは、Python 3 をインストールするまで失敗します。

著作権に関する情報

商標に関する情報