



Red Hat Gluster Storage 3.5

Deployment Guide for Public Cloud

Red Hat Gluster Storage on Public Cloudの展開

Red Hat Gluster Storage 3.5 Deployment Guide for Public Cloud

Red Hat Gluster Storage on Public Cloudの展開

Enter your first name here. Enter your surname here.

Enter your organisation's name here. Enter your organisational division here.

Enter your email address here.

法律上の通知

Copyright © 2023 | You need to change the HOLDER entity in the en-US/Deployment_Guide_for_Public_Cloud.ent file |.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

XFS[®] is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

概要

本ガイドでは、パブリッククラウドに Red Hat Gluster Storage をデプロイする方法について説明します。多様性を受け入れるオープンソースの強化 Red Hat では、コード、ドキュメント、Web プロパティにおける配慮に欠ける用語の置き換えに取り組んでいます。まずは、マスター (master)、スレーブ (slave)、ブラックリスト (blacklist)、ホワイトリスト (whitelist) の 4 つの用語の置き換えから始めます。この取り組みは膨大な作業を要するため、今後の複数のリリースで段階的に用語の置き換えを実施して参ります。詳細は、弊社の CTO、Chris Wright のメッセージを参照してください。

目次

第1章 パブリック来るアドにおける RED HAT ストレージ	4
1.1. GLUSTERFS について	4
1.2. 前提条件	4
第2章 AMAZON WEB SERVICES を使用した RED HAT GLUSTER STORAGE へのアクセス	5
2.1. AMAZON MACHINE IMAGE として RED HAT GLUSTER STORAGE へのアクセス	5
2.2. 前提条件	6
2.3. RED HAT GLUSTER STORAGE インスタンスの起動	6
2.4. RED HAT GLUSTER STORAGE インスタンスが実行中であることを確認します。	12
2.5. ストレージのプロビジョニング	15
2.6. RED HAT GLUSTER STORAGE インスタンスの停止および再起動	18
第3章 MICROSOFT AZURE を使用した RED HAT GLUSTER STORAGE へのアクセス	19
3.1. イメージのプロファイルとサイジング	20
3.2. 前提条件	21
3.3. プランニングのガイドライン	21
3.4. ARM モードでの MICROSOFT AZURE での RED HAT GLUSTER STORAGE の設定	22
3.4.1. Red Hat Gluster Storage for Microsoft Azure の取得	22
3.4.2. Microsoft Azure での新規リソースの設定	23
3.4.3. ディスクイメージの Microsoft Azure へのアップロード	26
3.4.4. Gluster Storage インスタンスのデプロイ	27
3.5. GLUSTER STORAGE クラスターの設定	31
3.6. 付録: ASM モードでの MICROSOFT AZURE での RED HAT GLUSTER STORAGE の設定	32
3.6.1. Red Hat Gluster Storage for Microsoft Azure の取得	32
3.6.2. ネットワークポロジの定義	32
3.6.3. ディスクイメージを Microsoft Azure にアップロードする	33
3.6.4. Gluster Storage インスタンスのデプロイ	34
3.6.5. Gluster Storage クラスターの設定	36
3.7. 付録: ISO からのカスタムディスクイメージの作成	36
3.8. 付録: PERFORMANCE CATEGORIZATION	39
3.8.1. ストレージタイプ	39
3.8.2. Bandwidth	39
3.8.3. ディスクライセンス	39
3.8.4. GlusterFS	40
第4章 GOOGLE CLOUD PLATFORM での RED HAT GLUSTER STORAGE の使用	43
4.1. デプロイメントのプランニング	44
4.1.1. 環境	45
4.1.2. 前提条件	45
4.1.3. プライマリストレージプールの設定	46
4.1.4. セカンダリストレージプールの設定	47
4.1.5. クライアント設定	48
4.1.6. 信頼されたプールトポロジ	48
4.1.7. Google Cloud Platform 向けの Red Hat Gluster Storage の取得	49
4.2. GOOGLE COMPUTE ENGINE の設定	49
4.2.1. SSH キー	49
4.2.2. クォータの設定	51
4.3. QCOW2 から .RAW 形式への変換	51
4.4. GOOGLE COMPUTE ENGINE のイメージのパッケージ化	51
4.5. GOOGLE CLOUD STORAGE へのイメージのアップロード	51
4.6. GOOGLE COMPUTE ENGINE へのイメージのインポート	52
4.7. RED HAT GLUSTER STORAGE インスタンスのディスクを設定するための仮想マシンインスタンスの作成	

	52
4.8. 初期データディスクの作成	53
4.9. データディスクのタッチおよび設定	54
4.10. イメージ作成プロセスのディスクの割り当て解除	56
4.11. イメージを使用した複数の RED HAT GLUSTER STORAGE インスタンスの作成	57
4.12. GOOGLE CLOUD DEPLOYMENT MANAGER を使用した複数インスタンスのデプロイ	59
4.13. RED HAT GLUSTER STORAGE の設定	59
4.13.1. ノードのピア	59
4.13.2. 分散ボリュームの作成	60
4.13.3. プライマリーからセカンダリーリージョンへの Geo レプリケーションの設定	64
4.14. データにアクセスするためのクライアントの設定	66
4.14.1. ネイティブクライアントのインストール	66
4.14.2. Red Hat Gluster Storage ボリュームのマウント	66
4.14.3. マウントしたボリュームのテスト	66
4.15. 付録: スクラットから RED HAT GLUSTER STORAGE COMPUTE ENGINE イメージのビルド	67
4.15.1. ISO から RAW ディスクイメージファイルへの Red Hat Gluster Storage のインストール	67
4.15.2. ネットワークインターフェースの有効化と起動	67
4.15.3. Red Hat Gluster Storage Server チャンネルのサブスクリプション	68
4.15.4. システムの更新	68
4.15.5. チューニングとその他の設定	68
4.15.6. Google Compute Engine の仮想マシンのカスタマイズ	68
4.16. 付録: RED HAT GLUSTER STORAGE デプロイメントの設定ファイル	69
付録A 改訂履歴	75

第1章 パブリック来るアドにおける RED HAT ストレージ

Red Hat Gluster Storage for Public Cloud は、パブリッククラウドにスケーラブルな NAS をデプロイするための glusterFS パッケージです。この強力なストレージサーバーは、高可用性、スケーラブル、仮想化、およびオンプレミスの NAS ストレージの一元的な管理プール内で、オンプレミス型のデプロイメントをすべて提供します。

さらに、Red Hat Gluster Storage は、Amazon Web Services (AWS) クラウド内など、Red Hat Gluster Storage をパブリッククラウドに使用するパブリッククラウドにデプロイできます。クラウドで非常にスケーラブルで高可用性 NAS を提供することで、プライベートクラウドまたはデータセンターで使用可能なすべての機能と機能をパブリッククラウドに提供します。

POSIX と互換性のある glusterFS サーバー (XFS ファイルシステム形式を使用してディスクにデータを保存する) は、Network File System (NFS) や Server Message Block (SMB) (CIFS としても知られる) などの業界標準のアクセスプロトコルを使用してアクセスできます。

1.1. GLUSTERFS について

glusterFS は、ネットワークで各種のストレージサーバーを集計し、それらを相互接続して1つの大きな並立ネットワークファイルシステムにします。スタック可能なユーザー空間設計に基づいて、さまざまなワークロードに対して例外的なパフォーマンスを提供し、Red Hat Gluster Storage の主要なビルディングブロックです。

1.2. 前提条件

Red Hat は、特定のサーバーの Red Hat Gluster Storage の互換性に、汎用でユースケースに特化した基準を設けています。

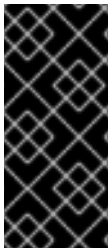
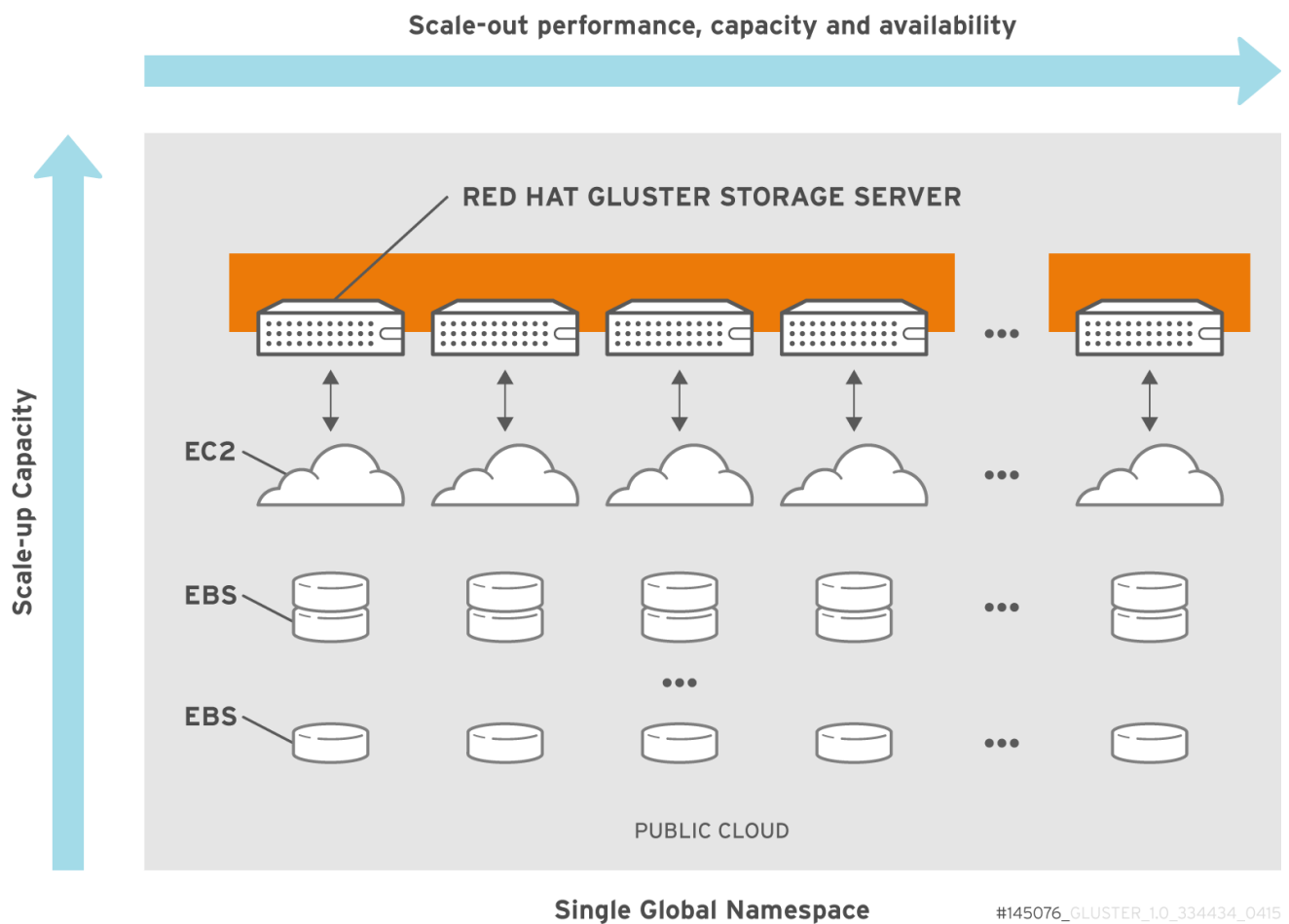
実稼働デプロイメントのサポート可能性を確保するための前提条件として、<https://access.redhat.com/articles/66206#CE> で指定した条件に基づいて Cloud デプロイメント向けに Red Hat Gluster Storage を実行するためのインスタンスタイプを指定します。

第2章 AMAZON WEB SERVICES を使用した RED HAT GLUSTER STORAGE へのアクセス

Red Hat Gluster Storage for Public Cloud は、Amazon Web Services (AWS) パブリッククラウドでスケーラブルなネットワーク接続ストレージ (NAS) をデプロイするための Amazon Machine Image (AMI) として glusterFS をパッケージ化します。このストレージサーバーは、Amazon ユーザー向けに、可用性が高く、スケーラブルで仮想化され、集中化された管理プールを提供します。Red Hat Gluster Storage for Public Cloud は、AWS 内で可用性の高いストレージを提供します。AWS アベイラビリティゾーン間での同期の N 方向レプリケーションは、AWS リージョン内で高可用性を提供します。非同期 Geo レプリケーションは、AWS リージョン全体で高可用性を確保するために、継続的にデータレプリケーションを提供します。glusterFS グローバル名前空間機能は、ディスクおよびメモリーリソースを、物理ハードウェアから抽象化された統合ストレージボリュームに集約します。

以下の図は、AWS の Red Hat Gluster Storage との統合を示しています。

図2.1 Amazon Web Services 統合アーキテクチャー



重要

以下の機能は Amazon Web Services ではサポートされません。

- NFS および CIFS の高可用性
- Red Hat Gluster Storage Web 管理

2.1. AMAZON MACHINE IMAGE として RED HAT GLUSTER STORAGE へのアクセス

Amazon Machine Image (AMI) として Red Hat Gluster Storage へのアクセスを取得するには、以下の資料を参照してください。

- <https://access.redhat.com/knowledge/articles/145693>
- <https://access.redhat.com/solutions/314993>

2.2. 前提条件

- ノードの最小数は 3 です。
- 互換性のある物理サーバー、仮想サーバー、クライアントの OS プラットフォームについては、<https://access.redhat.com/articles/66206> を参照してください。

2.3. RED HAT GLUSTER STORAGE インスタンスの起動

このセクションでは、Amazon Web Services で Red Hat Gluster Storage インスタンスを起動する方法を説明します。

3 方向のレプリケーションでサポートされる設定は、ほぼ同等のサイズの 24 Amazon Elastic Block Store (EBS) ボリュームです。

表2.1 Amazon Web Services でサポートされる設定

EBS ボリュームタイプ	インスタンスごとのボリューム数	インスタンスごとのボリュームの最大数	EBS ボリュームの容量範囲
Magnetic	1	24	1 GiB - 1 TiB
汎用 SSD	1	24	1 GiB - 16 TiB
PIOPS SSD	1	24	4 GiB - 16 TiB
最適化した HDD (ST1)	1	24	500 GiB - 16 TiB
コールド HDD (SC1)	1	24	500 GiB - 16 TiB

- Red Hat Gluster Storage ボリュームスナップショットの作成は、magnetic、一般的な SSD および PIOPS EBS ボリュームでサポートされます。USS を使用してスナップショットコンテンツを参照することもできます。Red Hat Gluster Storage ボリュームスナップショットの管理に関する情報は、『『Red Hat Gluster Storage 管理ガイド』』の「『スナップショットの管理』」を参照してください。

-



警告

階層化は、Red Hat Gluster Storage 3.5 では非推奨です。Red Hat ではこの使用を推奨しておらず、Red Hat Gluster Storage 3.5.3 にアップグレードする新規デプロイメントおよび既存デプロイメントでは階層化をサポートしません。



警告

Gluster-NFS は、Red Hat Gluster Storage 3.5 では非推奨です。Red Hat では Gluster-NFS の使用を推奨しておらず、Red Hat Gluster Storage 3.5.3 にアップグレードする新規デプロイメントおよび既存デプロイメントでのその使用をサポートしません。



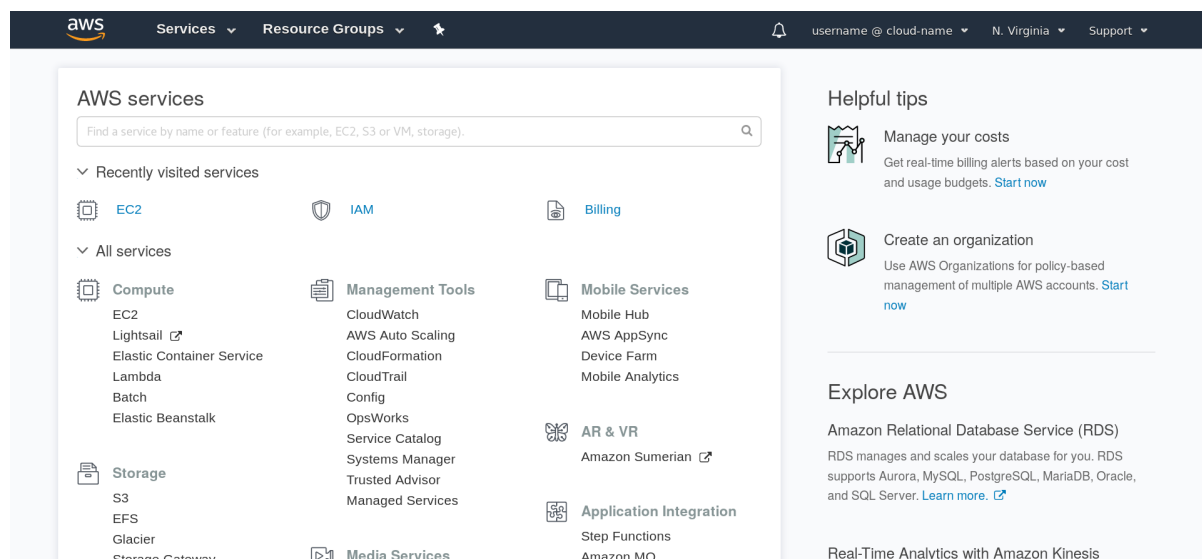
警告

RDMA をトランスポートプロトコルとして使用することは、Red Hat Gluster Storage 3.5 で非推奨となっています。Red Hat ではこの使用を推奨しておらず、Red Hat Gluster Storage 3.5.3 にアップグレードする新規デプロイメントおよび既存デプロイメントでサポートしません。

Amazon Web Service 環境は、Red Hat Gluster Storage 階層機能をサポートします。magnetic EBS ボリュームで作成された既存の Red Hat Gluster Storage ボリュームまたは新しい Red Hat Gluster Storage ボリュームに、PIOPS から作成したブリックをアタッチすることができます。階層化されたボリュームの作成に関する情報は、『『RedHat Gluster ストレージ管理ガイド』』の「『階層の管理』」を参照してください。

Red Hat Gluster Storage インスタンスを起動するには、以下を実行します。

1. <http://aws.amazon.com> で Amazon Web Services ホームページに移動します。
2. Amazon Web Services にログインします。AWS マネジメントコンソール 画面が表示されません。



3. EC2 オプションをクリックします。EC2 管理コンソールが表示されます。

4. **Launch Instance** をクリックします。ステップ 1: Amazon Machine Image (AMI)画面が表示されます。

5. **My AMIs** をクリックし、**Shared with me** チェックボックスを選択します。

6. 必要な AMI を検索し、AMI に対応する **Select** をクリックします。ステップ 2: インスタンスタイプの画面が表示されます。

Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. [Learn more](#) about instance types and how they can meet your computing needs.

Filter by: **All Instance types** **Current generation** [Show/Hide Columns](#)

Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
<input type="radio"/>	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
<input checked="" type="radio"/>	General purpose	t2.micro <small>Free tier eligible</small>	1	1	EBS only	-	Low to Moderate	Yes
<input type="radio"/>	General purpose	t2.small	1	2	EBS only	-	Low to Moderate	Yes

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Configure Instance Details](#)

Feedback English (US) © 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

7. インスタンスの必要なタイプを探し、インスタンスタイプに対応するラジオボタンを使用して選択します。
8. **Next:Configure Instance Details** をクリックします。ステップ 3: **Configure Instance Details** 画面が表示されます。

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances [Launch into Auto Scaling Group](#)

Purchasing option Request Spot instances

Network [Create new VPC](#)

Subnet [Create new subnet](#)
206 IP Addresses available

Auto-assign Public IP

IAM role [Create new IAM role](#)

Shutdown behavior

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Add Storage](#)

Feedback English (US) © 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

9. インスタンスの設定を指定するか、デフォルト設定を続行するには、**Next: Add Storage** をクリックします。ステップ 4: **Add Storage** 画面が表示されます。

Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more](#) about storage options in Amazon EC2.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encrypted
Root	/dev/sda1	snap-ba40cac8	10	General Purpose SSD (GP2)	100 / 3000	N/A	<input checked="" type="checkbox"/>	Not Encrypted

[Add New Volume](#)

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. [Learn more](#) about free usage tier eligibility and usage restrictions.

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Add Tags](#)

10. ステップ 4: Add Storage 画面でストレージの詳細を指定し、Next: Add Tags をクリックします。ステップ 5: Add Tags 画面が表示されます。

Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. A copy of a tag can be applied to volumes, instances or both. Tags will be applied to all instances and volumes. [Learn more](#) about tagging your Amazon EC2 resources.

Key	Value	Instances	Volumes
This resource currently has no tags.			

Choose the [Add tag](#) button or [click to add a Name tag](#).
Make sure your [IAM policy](#) includes permissions to create tags.

[Add Tag](#) (Up to 50 tags maximum)

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Configure Security Group](#)

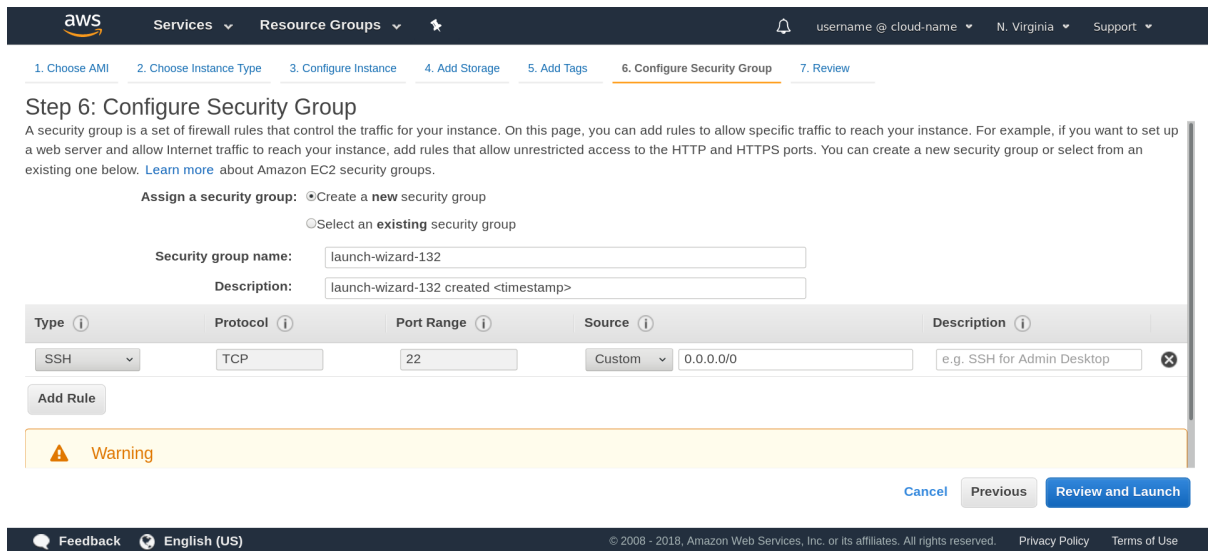
11. Add をクリックして、各タグの Value フィールドに必要な情報を入力します。



重要

Name タグの追加が必要です。Name タグを追加するには、クリックして Name タグを追加をクリックします。この名前を後で使用して、インスタンスが正常に動作していることを確認できます。

12. Next: Configure Security Group をクリックします。ステップ 6: 「セキュリティグループの設定」画面が表示されます。



Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

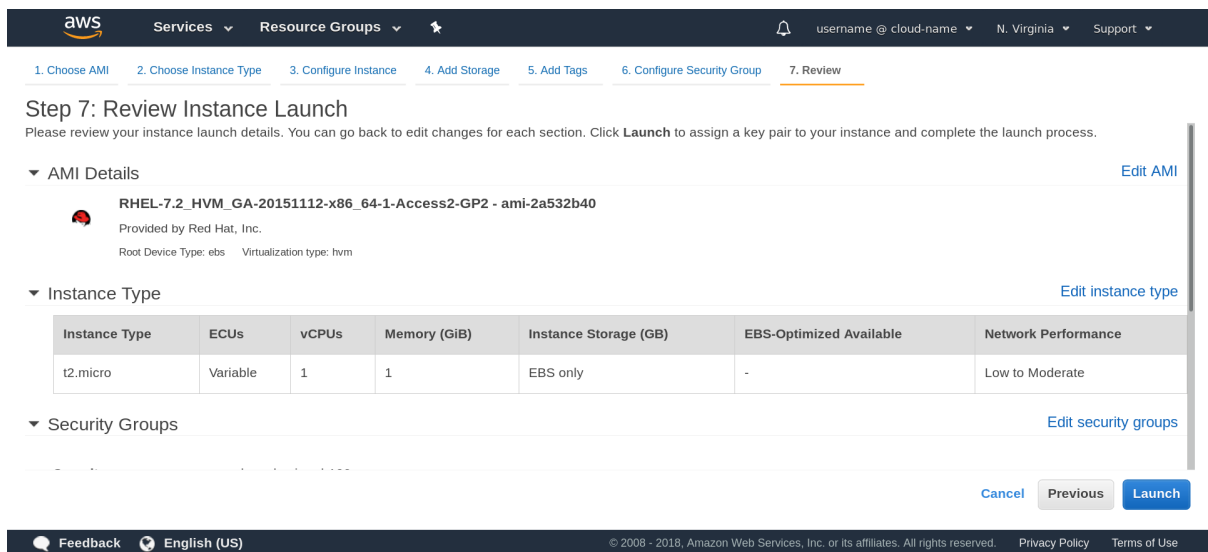
Assign a security group: Create a new security group
 Select an existing security group

Security group name:
 Description:

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop

Warning


13. 新しいセキュリティーグループを作成するか、既存のセキュリティーグループを選択します。
14. 新しいまたは選択したセキュリティーグループで、以下の TCP ポート番号を開くようにしてください。
 - 22: 作成したインスタンスへの ssh アクセスを許可
15. **Review and Launch** をクリックします。ステップ7: インスタンス起動の確認画面が表示されます。



Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

▼ AMI Details [Edit AMI](#)

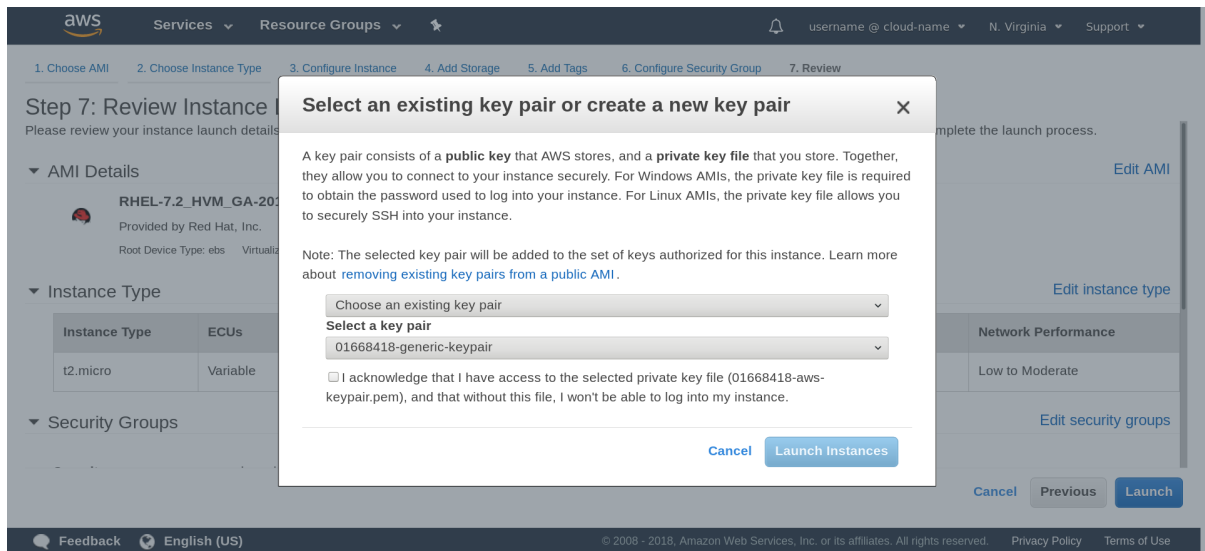
 **RHEL-7.2_HVM_GA-20151112-x86_64-1-Access2-GP2 - ami-2a532b40**
 Provided by Red Hat, Inc.
 Root Device Type: ebs Virtualization type: hvm

▼ Instance Type [Edit instance type](#)

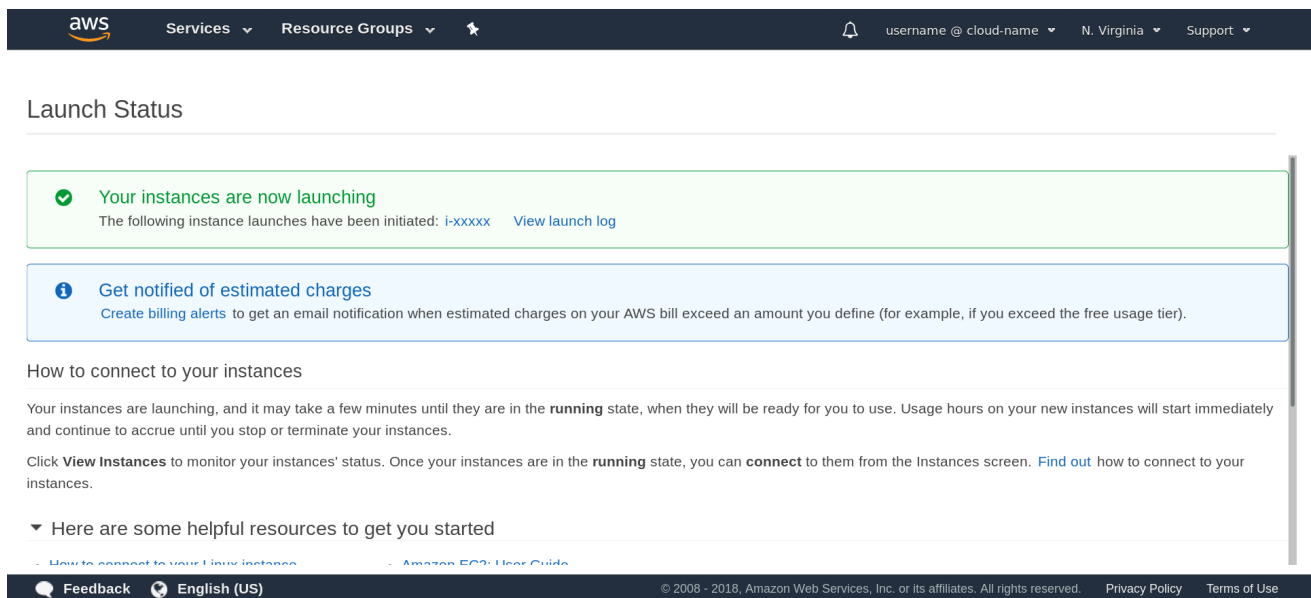
Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only	-	Low to Moderate

▼ Security Groups [Edit security groups](#)

16. 必要な設定を確認して編集し、**起動** をクリックします。
17. 既存のキーペアを選択するか、新規キーペアを作成して **インスタンスの起動** をクリックします。



Launch Status 画面には、インスタンスの起動が表示されます。

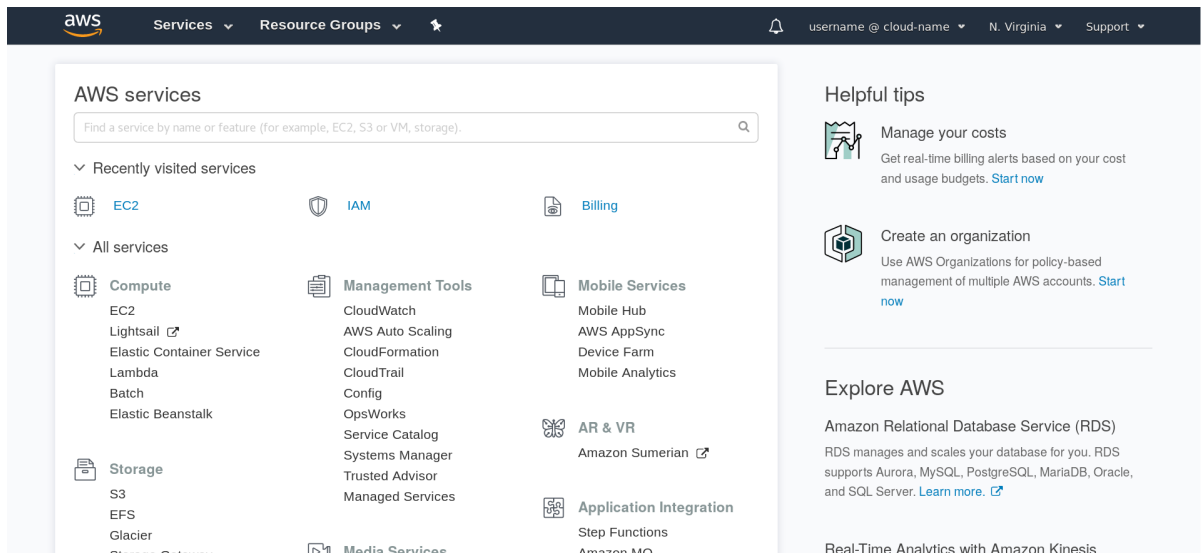


2.4. RED HAT GLUSTER STORAGE インスタンスが実行中であることを確認します。

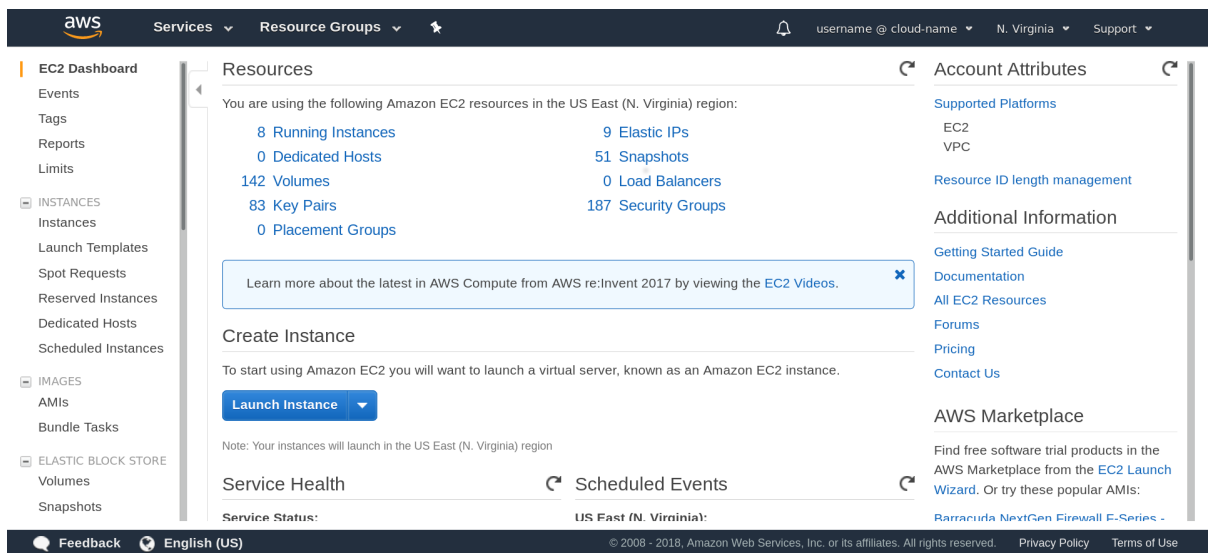
Red Hat Gluster Storage インスタンスにリモートログインを実行し、コマンドを実行し、Red Hat Gluster Storage インスタンスが実行されていることを確認できます。

Red Hat Gluster Storage インスタンスが実行中であることを確認するには、以下を実行します。

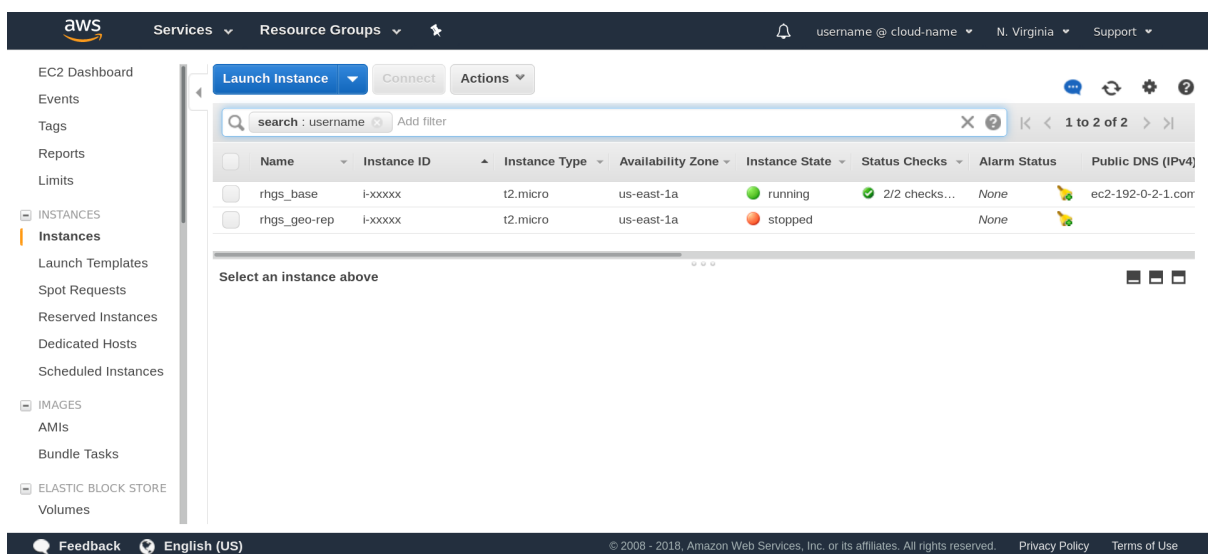
1. <http://aws.amazon.com> に移動します。
2. Amazon Web Services にログインします。AWS マネジメントコンソール 画面が表示されます。



3. EC2 オプションをクリックします。EC2 管理コンソールが表示されます。



4. Instances セクションで、Instances リンクをクリックします。画面には、現在のインスタンスが表示されます。



5. Instance State 列をチェックして、インスタンスが実行していることを確認します。黄色の円は、インスタンスが実行されていることを示す緑色の円で、保留中のステータスを示します。

インスタンスを選択し、**Description** タブに表示される詳細を確認します。

6. 次のコマンドを使用して、パブリック DNS フィールドのドメイン名を使用して Red Hat Amazon Machine Image インスタンスにリモートでログインします。

```
# ssh -i Key_Pair_File Public_DNS
# sudo su
```



注記

インスタンスの起動時に、選択したキーペアを使用する必要があります。

たとえば、キーペアのファイル名は `rhs-aws.pem` で、パブリック DNS は `ec2-user@ec2-23-20-52-123.compute-1.amazonaws.com` です。

```
# ssh -i rhs-aws.pem ec2-user@ec2-23-20-52-123.compute-1.amazonaws.com
# sudo su
```

7. 以下のコマンドを使用して、**glusterd** デーモンがインスタンスで実行されていることを確認します。

```
# service glusterd status
```



注記

Samba および NFS-Ganesha チャネルはデフォルトで無効になります。スタンドアロンの Samba および NFS-Ganesha を使用するには、以下を実行します。

- Red Hat Gluster Storage Samba リポジトリを有効にしてインストールする場合は、『『Red Hat Gluster Storage 3.5 インストールガイド』』の「『Red Hat Gluster Storage への Samba のデプロイ』」を参照してください。
- Red Hat Gluster Storage NFS-Ganesha リポジトリを有効にしてインストールする場合は、『『Red Hat Gluster Storage 3.5 インストールガイド』』の「『Red Hat Gluster Storage への NFS-Ganesha のデプロイ』」を参照してください。

 重要

yum update を使用して Amazon EC2 Red Hat Gluster Storage AMI を更新する前に、<https://access.redhat.com/solutions/1556793> のナレッジベースの記事に記載の手順に従ってください。

2.5. ストレージのプロビジョニング

Amazon Elastic Block Storage (EBS) は、Amazon EC2 インスタンスで使用するために特別に設計されています。Amazon EBS は、raw、未フォーマットの外部ブロックデバイスのように機能するストレージを提供します。本セクションでは、論理ボリュームの形式で、ブリックを作成するためのストレージをプロビジョニングする手順をステップごとに説明します。これらのブリックを使用して、Gluster アナラートされた複製ボリュームなどの異なるタイプのボリュームを作成し、Gluster の 3 方向の複製ボリュームを作成できます。

 重要

- Red Hat Gluster Storage 3.4 以降、Red Hat は、Red Hat Gluster Storage インスタンスが使用する場合に複数の EBS ボリュームの集約を推奨しません。本章の手順に従って、AWS 環境で Red Hat Gluster Storage を設定します。
- Red Hat Gluster Storage EC2 インスタンスでサポートされる設定は、最大 24 個の Amazon EBS ボリュームです。
- 3 方向の複製ボリュームまたは Arbitrated ボリューム用にストレージをプロビジョニングする場合、3 つの異なるゾーンにボリュームの各レプリカセットを作成する必要があります。この設定では、2 つのアベイラビリティゾーンが利用できない場合でも、データの可用性に影響を与えることはありません。クライアント側のクォーラムはデフォルトで有効になっています。そのため、2 つのゾーンが利用できないと、アクセスが読み取り専用になります。
- 仮想マシンのスナップショット、または Red Hat Gluster Storage Server がゲストオペレーティングシステムとしてインストールされているインスタンス、ファイバーチャネルまたは iSCSI SAN スナップショットなどの外部スナップショットはサポートされていません。
- 調整された複製ボリュームまたは 3 方向の複製ボリュームを作成する場合には、汎用 SSD または PIOPS SSD で構成される **EBS** ボリュームタイプを使用する必要があります。

1. 以下のコマンドを使用して物理ボリューム (PV) を作成します。

```
# pvcreate device
```

以下は例になります。

```
# pvcreate /dev/xvdd
Physical volume "/dev/xvdd" successfully created.
```



注記

- `/dev/babel` は、ストレージデバイスです。複数のボリュームがある場合は、このコマンドをすべてのディスクで実行する必要があります。以下は例になります。

```
# pvcreate /dev/xvdd /dev/xvde /dev/xvdf...
```

- デバイス名と調整値は、使用中のデバイスによって異なります。

2. 以下のコマンドを使用して PV からボリュームグループ (VG) を作成します。

```
# vgcreate volume_group device
```

以下は例になります。

```
# vgcreate rhs_vg /dev/xvdd
Volume group "rhs_vg" successfully created
```



注記

`/dev/babel` は、ストレージデバイスです。複数のデバイスがある場合は、複数の VG を作成します。

3. 使用可能な最大サイズまたは EBS ボリュームサイズの 0.5% のプールサイズのシンプールを作成します。

```
# lvcreate --thinpool VOLGROUP/POOLNAME -l 100%FREE --chunksize chunk_size --
poolmetadatasize metadata_pool_size --zero n
```

メタデータ LV に可能な最大サイズは 16 GiB です。Red Hat Gluster Storage は、サポートされている最大サイズのメタデータデバイスを作成することを推奨します。領域が懸念される場合は、最大数未満を割り当てることができますが、この場合はプールサイズの最小 0.5% を割り当てる必要があります。

例 1

デバイスの最大サイズの可能な最大メタデータデバイスサイズでシンプールを作成するには、次のコマンドを実行します。

```
# lvcreate --thinpool rhs_vg/rhs_pool -l 100%FREE --chunksize 256K --poolmetadatasize 16g
--zero n
Using default stripesize 64.00 KiB.
Logical volume "rhs_pool" created.
```

例 2

EBS ボリュームのサイズが 20 GiB の 0.5% プールメタデータサイズを作成するには、以下を実行します。

```
# lvcreate --thinpool rhs_vg1/rhs_pool1 -l 100%FREE --chunksize 256K --poolmetadatasize
0.1g --zero n
```

```
Using default stripesize 64.00 KiB.
Rounding up size to full physical extent 104.00 MiB
```

4. **lvs** コマンドを実行して、EBS ボリュームで利用可能な最大デバイス領域を取得します。論理ボリュームの作成時に、このコマンドに表示されている **LSize** サイズを使用できます。これにより、EBS ボリューム全体が使用され、容量がなくなるようになります。

```
# lvs rhs_vg
LV   VG   Attr   LSize Pool Origin Data% Meta% Move Log Cpy%Sync Convert
rhs_pool rhs_vg twi-a-t-- 1.97t   0.00 0.02
```

5. 直前のコマンドで取得した **LSize** を使用して論理ボリュームを作成し、シンプル全体を1つのLVに消費するか、1つのシンプルから複数のLVsを作成することができます。

```
# lvcreate -V lv_size -T VOLGROUP/POOLNAME -n lv_name
```

例 1

直前のコマンドで取得した **LSize** を使用して単一のLVを作成し、シンプル全体を1つの論理ボリュームに消費します。

```
# lvcreate -V 1.96T -T rhs_vg/rhs_pool -n rhs_lv
Using default stripesize 64.00 KiB.
Rounding up size to full physical extent 1.96 TiB
Logical volume "rhs_lv" created.
```

シンプルの自動拡張アクティベーション/`thin_pool_autoextend_threshold` は 100 未満である必要があります。

例 2

1つのシンプルから複数のLVを作成します。この例では、シンプル `rhs_pool1` からサイズ 2G のLVを作成します。

```
# lvcreate -V 2G -T rhs_vg1/rhs_pool1 -n rhs_lv
Using default stripesize 64.00 KiB.
Logical volume "rhs_lv" created.

# lvcreate -V 2G -T rhs_vg1/rhs_pool1 -n rhs_lv1
Using default stripesize 64.00 KiB.
Logical volume "rhs_lv1" created.
```

6. 以下のコマンドを使用して、論理ボリュームをフォーマットします。

```
# mkfs.xfs -i size=512 -n size=8192 /VOLGROUP/POOLNAME/
```

たとえば、`/dev/rhgs_vg/rhgs_lv` をフォーマットするには、次のコマンドを実行します。

```
# mkfs.xfs -i size=512 -n size=8192 /dev/rhgs_vg/rhgs_lv
```

7. 以下のコマンドを使用して、マウントポイントを作成し、論理デバイスをマウントします。

```
# mkdir File Mount Path
# mount -t xfs -o inode64,noatime Logical Volume Path File Mount Path
```

以下は例になります。

```
# mkdir /export/mountlv
# mount -t xfs -o inode64,noatime /dev/rhgs_vg/rhgs_lv /export/mountlv
```

8. **/etc/fstab** に以下の行を追加して、ファイルシステムを自動的にマウントします。

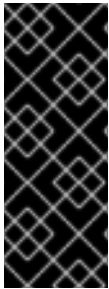
```
# Logical Volume Path File Mount Path xfs rw,inode64,noatime,nouuid 0 0
```

以下は例になります。

```
# /dev/rhgs_vg/rhgs_lv /export/mountlv xfs rw,inode64,noatime,nouuid 0 0
```

EBS ボリュームを追加したら、マウントポイントをブリックとして使用し、既存のボリュームまたは新規ボリュームの一部として使用できます。Arbitrated replicated volumes および 3 方向の複製ボリュームを作成する方法は、『『Red Hat Gluster Storage Administration Guide』』の「『Red Hat Gluster Storage ボリューム』」の章を参照してください。

2.6. RED HAT GLUSTER STORAGE インスタンスの停止および再起動



重要

Red Hat Gluster Storage インスタンスを停止して再起動すると、Amazon Web Services は新しい IP アドレスとホスト名を割り当てます。これにより、インスタンスが仮想ハードウェアとの関連付けが失われるため、信頼できるストレージプールへの中断が生じます。エラーを防ぐには、再起動した Red Hat Gluster Storage インスタンスを信頼できるストレージプールに追加します。『『Red Hat Gluster Storage 管理ガイド』』の「『信頼できるストレージプールへのサーバーの追加』」を参照してください。

Red Hat Gluster Storage インスタンスを再起動すると、IP アドレスとホスト名が維持され、仮想ハードウェアとの関連付けは失われません。これにより、信頼できるストレージプールへの接続が中断されることはありません。

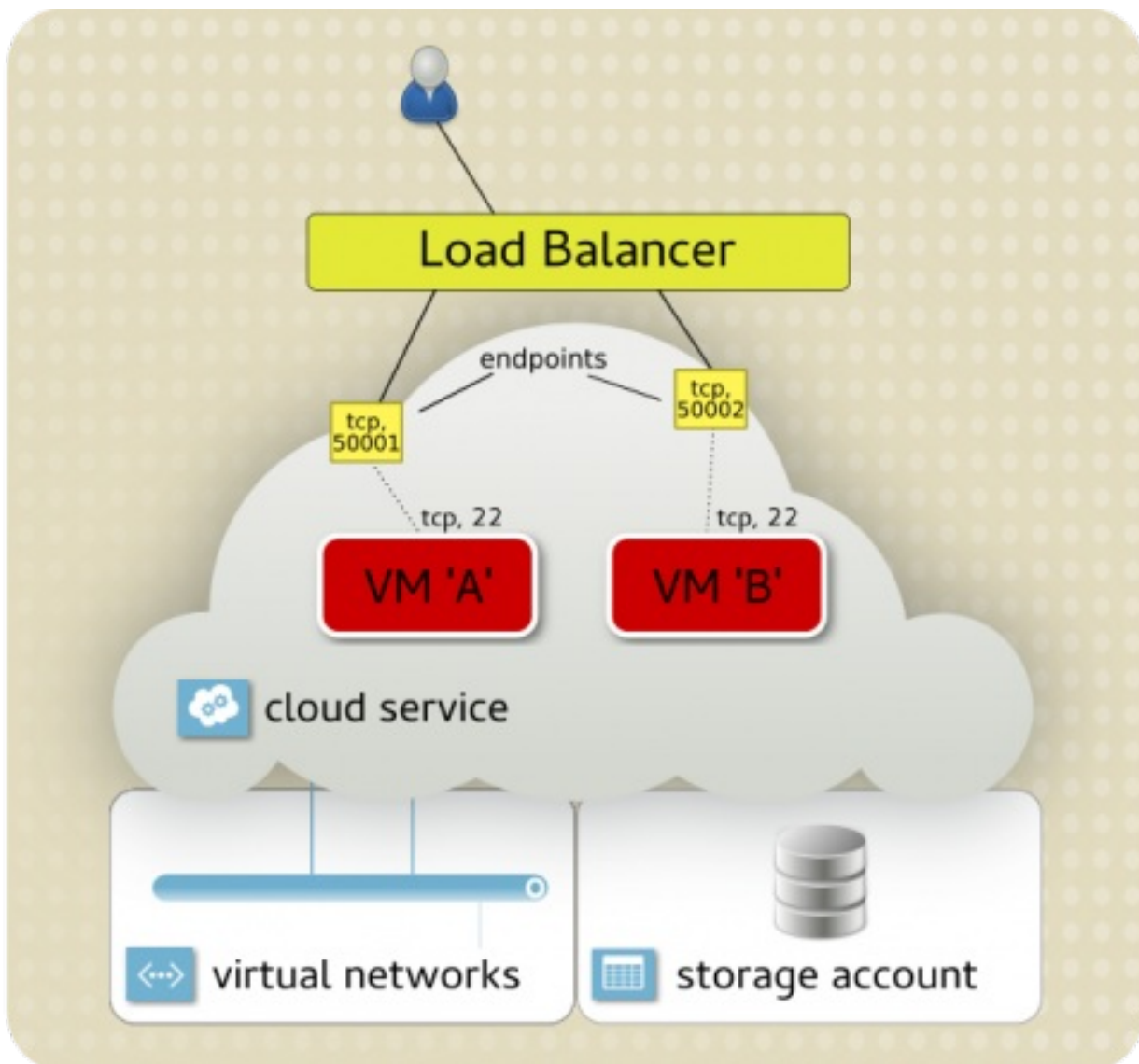
第3章 MICROSOFT AZURE を使用した RED HAT GLUSTER STORAGE へのアクセス

Red Hat Gluster Storage は、ユーザーおよびアプリケーションに柔軟なファイルサービスレイヤーを提供することで、ワークロードに合わせて簡単にスケーリングできる方法で設計されています。柔軟性は、Red Hat Gluster Storage の主要な強度です。Gluster は、Microsoft Azure を含む、オンプレミス環境、プライベートクラウド、パブリッククラウドなど、仮想または物理サーバーにデプロイすることができます。

統合アーキテクチャー

Microsoft Azure のアーキテクチャーは、ソリューションの作成方法を形化します。Microsoft Azure は、platform-as-a-service (PaaS) 環境または infrastructure-as-a-service (IaaS) 環境として機能できるクラウドサービスを提供します。Gluster Storage の場合、クラウドサービスは、仮想インスタンスをデプロイする論理コンテナを提供する IaaS レイヤーでなければなりません。IaaS コンテナでは、Microsoft Azure は DNS や DHCP などのネットワークサービスを提供します。これにより、物理デプロイメントを管理するのと同様の仮想インスタンスを管理します。

図3.1統合アーキテクチャー



クラウドサービスは、**cloudapp.net** ドメインに適用される接頭辞である名前で定義されます。クラウドサービス内のインスタンスへのアクセスは、クラウドサービス名および TCP ポート (エンドポイント) を指定して行われます。通常、これは SSH アクセスです。たとえば、クラウドサービス内で 30 個

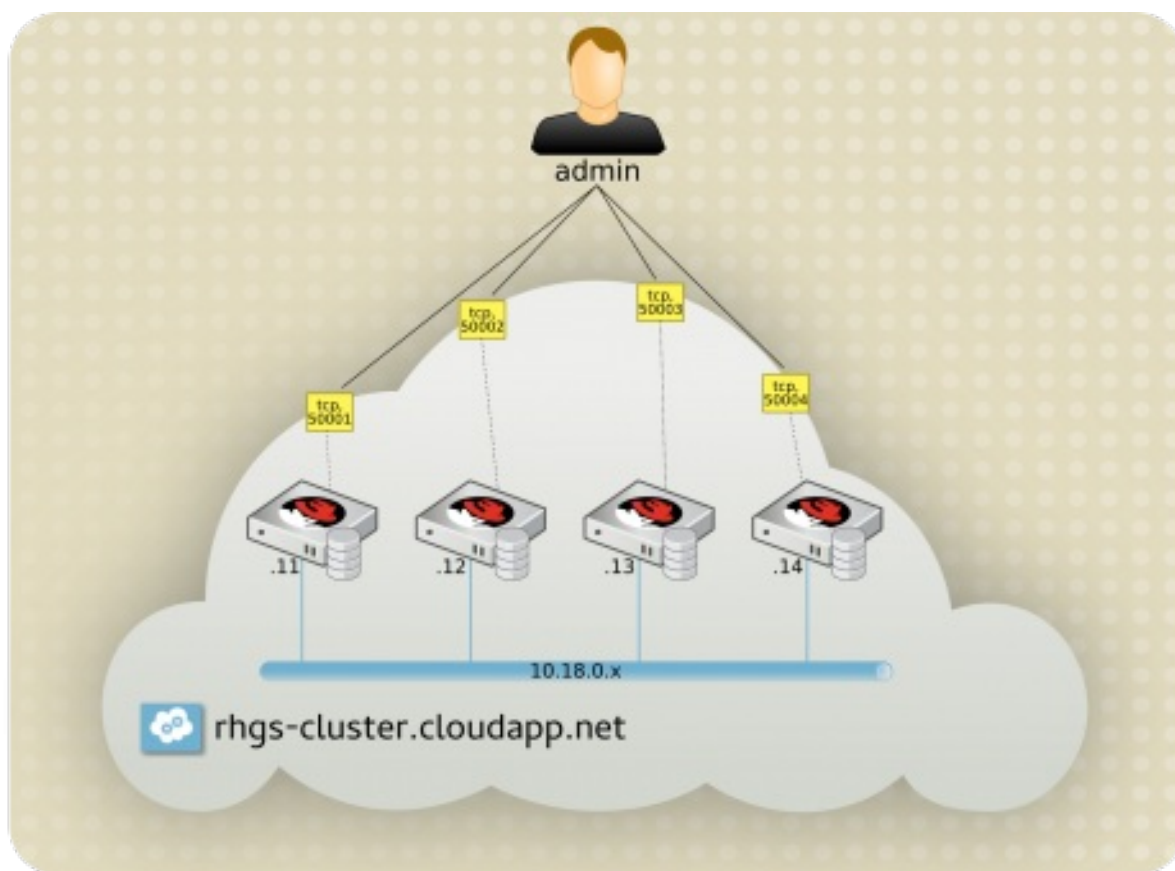
の仮想インスタンスが実行されている場合があります。そのため、インスタンスごとに異なるエンドポイントを公開すると、個別にアクセスできます。**50,001** はインスタンス A のポート 22 にリンク、**50,002** はインスタンス B のポート 22 にリンク。

仮想ネットワークにより、クラウドサービス内のインスタンスに対する制御および接続性が向上します。仮想ネットワークは、Microsoft Azure インフラストラクチャー内で純粋に機能するように設定するか、サイト間 VPN 接続を介してオンプレミスネットワークをクラウドサービスに接続するために使用できます。

最後のキーアーキテクチャー要素はストレージアカウントです。ストレージアカウントは、Microsoft Azure 内のストレージサービスへのアクセスを提供します。このアカウントはデータに固有の namespace を提供し、Blob、テーブル、キュー、およびファイルなどの多くのアクセスプロトコルをサポートします。データは SSD (premium) または HDD (標準) のいずれかに保存できます。

本章のワークフローでは、Red Hat Gluster Storage クラスタを作成します。

図3.2 Microsoft Azure および Red Hat Gluster Storage のワークフロー



重要

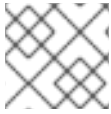
以下の Red Hat Gluster Storage Server 機能は、Microsoft Azure ではサポートされていません。

- Red Hat Gluster Storage Console および Nagios Monitoring
- NFS-Ganesha および CIFS High Availability

3.1. イメージのプロファイルとサイジング

Microsoft Azure は、プロジェクトのワークロードに基づいて、選択する各種仮想マシン設定を提供します。設定例では、Standard Tier/A2 インスタンスを想定しています。

本章で実施されるタスクは、Standard Tier/A2 インスタンスのサイズに基づいています。



注記

Red Hat Gluster Storage の各インスタンスには、最小で2つのコアが必要です。

Microsoft Azure は、オペレーティングシステムディスクの他に、リソースディスクにすべてのインスタンスも割り当てます。これは、インスタンスが実行されている物理ホストのローカルストレージからインスタンスに対してランタイム時に指定される非永続 (一時) ディスクです。リソースディスクは `/mnt/resource` に表示され、アプリケーションのスワップ領域と一時ストレージを提供するために Windows Azure Linux エージェントで設定されます。

各インスタンスタイプについて、Microsoft Azure ポータルには、CPU コア数および RAM を明確に示すが、各インスタンスがサポートする設定可能なディスクの数は表示されません。1 から 32 までの設定可能なデータディスク範囲の数は、インスタンスタイプによって異なります。

Red Hat Gluster Storage はストレージプラットフォームであるため、インスタンスのサイジング時に新たな計画の考慮事項があります。

- 仮想ディスクの最大サイズは 1023 GB です。複数の 1023 GB のディスクを集約することで、大きなディスクサイズに対応することができます。
- ディスクが定義されると、そのサイズを簡単に変更することはできません。Microsoft Azure Standard Storage の容量コストは使用をベースとしており、割り当てられた容量ではなく、Red Hat Gluster Storage ノードに割り当てられるすべてのディスクが 1023 GB になることが推奨されます。
- CPU、RAM、およびディスク数などの属性は、インスタンスの作成後に簡単に変更できますが、ネットワークの特性は使用できません。設定を計画する際には、インスタンスの作成前にネットワークトポロジーと接続に必要な接続を考慮してください。Microsoft Azure インスタンスは、複数のネットワークカードと複数の仮想ネットワークをサポートしますが、これらのタイプの高度なネットワーク機能は Windows Powershell を使用してのみ設定可能です。

3.2. 前提条件

- <https://access.redhat.com/articles/uploading-rhel-image-to-azure#install-the-azure-cross-platform-cli-on-your-azure-administration-server-6> 記載の指示に従って、Microsoft Azure CLI をインストールします。
- <https://access.redhat.com/articles/migrating-to-red-hat-cloud-access> 記載の指示に従って、システムを Red Hat から Microsoft Azure に移行します。
- ノードの最小数は 3 です。
- 互換性のある物理サーバー、仮想サーバー、クライアントの OS プラットフォームについては、<https://access.redhat.com/articles/66206> を参照してください。

Gluster Storage は、<https://azure.microsoft.com/en-in/documentation/articles/powershell-install-configure> の説明に基づいて Windows Powershell 環境を使用して管理することもできます。ただし、本章には記載されていません。この手順はすべて、Microsoft Azure CLI を使用します。

3.3. プランニングのガイドライン

以下は、Microsoft Azure に Red Hat Gluster Storage を設定するためのガイドラインです。

- Microsoft Azure サービスを操作し、制御するための管理サーバーを指定します。単純な Gluster Storage デプロイメント (単一サイト、単一 NIC) の場合、管理プラットフォームは Linux サーバー/ワークステーションになります。より複雑なデプロイメントでは、Powershell を使用する Windows デスクトップが推奨されます。
- `initramfs` ファイルに含まれる Hyper-V ドライバーを使用して、Red Hat Enterprise Linux 7 をベースにカスタムイメージをビルドします。これらのドライバーが存在しない場合は、インスタンスの起動に失敗します。
- Red Hat Gluster Storage ノードに仮想ネットワークを使用します。
- Geo レプリケーションの場合は、共通の `/etc/hosts` ファイルをすべてのノードにデプロイするか、共有 DNS サーバーを使用します。
- 標準ストレージに対する課金は、使用される容量に基づいています。したがって、データディスクの最大サイズ (1023 GB) を使用し、将来の管理オーバーヘッドを最小限にするために、インストール時にインスタンスをサポートする数だけ割り当てると便利です。
- NFS が Gluster Storage ノードに接続するのに推奨される方法としては、クロックの速度が高くなる CPU を持つ **D** シリーズインスタンスを使用することを検討してください。
- 可用性を強化するために、可用性セットを使用して Gluster Storage ノードを複製セット内にグループ化します。
- **mdadm** を使用して、ディスクを組み合わせせてより大きなディスクを形成します。
- より少ない仮想マシンを使用して、最高の容量を提供します。
- 高可用性データアクセスの場合は、ネイティブ **glusterfs** クライアントで複製された GlusterFS ボリュームを使用します。
- 各 Gluster Storage ノードで実行している SSH サービスへの公開アクセスには、デフォルト以外の SSH ポートを使用します (つまり、`vm create` で `--ssh` を使用します)。

3.4. ARM モードでの MICROSOFT AZURE での RED HAT GLUSTER STORAGE の設定

このセクションでは、Azure Resource Manager モードで Microsoft Azure で Red Hat Gluster Storage を設定する手順を説明します。



注記

Microsoft Azure には、Azure Service Management (ASM) モード (Azure Classic と呼ばれる) モードと Azure Resource Manager (ARM) モードには 2 つのデプロイメントモードがあります。本セクションの手順では、ARM モードおよび ARM クロスプラットフォームのコマンドラインインターフェース (Xplat-CLI) コマンドを使用して、Red Hat Gluster Storage を Microsoft Azure にデプロイします。Xplat-CLI コマンドは、これらの 2 つのデプロイメントモードによって若干異なります。Azure Service Management (ASM) モードを使用して Microsoft Azure で Red Hat Gluster Storage を設定する場合は、「[付録: ASM モードでの Microsoft Azure での Red Hat Gluster Storage の設定](#)」を参照してください。

3.4.1. Red Hat Gluster Storage for Microsoft Azure の取得

Red Hat サブスクリプションまたは Red Hat 評価サブスクリプションを使用して Red Hat Gluster Storage Server ファイルをダウンロードするには、以下を実行します。

1. Red Hat カスタマーポータル (<https://access.redhat.com/login>) にアクセスし、ユーザー名およびパスワードを入力してログインします。
2. **Downloads** をクリックし、**Software & Download Center** に移動します。
3. Red Hat Gluster Storage Server エリアで、**Download Software** をクリックして **VHD** イメージの最新バージョンをダウンロードします。
4. ファイルがダウンロードされたディレクトリーに移動し、ファイルで **sha256sum** コマンドを実行します。

以下に例を示します。

```
$ sha256sum rhgs-azure-3.5-rhel-7-x86_64.tar.gz
2d083222d6a3c531fa2fbbd21c9ea5b2c965d3b8f06eb8 ff3b2b0efce173325d rhgs-azure-
3.5-rhel-7-x86_64.tar.gz
```

sha256sum ユーティリティーで生成された値は、ファイルについて Red Hat カスタマーポータルに表示される値と一致している必要があります。これらが同じでない場合は、ダウンロードが不完全または破損しているため、ファイルを再度ダウンロードする必要があります。ダウンロードの試行後にチェックサムが正常に検証されていない場合は、Red Hat サポートにお問い合わせください。

5. ダウンロードしたファイル **rhgs-azure-[version].zip** を展開してアーカイブコンテンツを展開します。

以下に例を示します。

```
# tar -xvzf rhgs-azure-3.5-rhel-7-x86_64.tar.gz
```

3.4.2. Microsoft Azure での新規リソースの設定

Microsoft Azure で新規リソースをセットアップする前に、Azure cross-platform コマンドラインインターフェース (Xplat-CLI) をインストールする必要があります。Azure Xplat-CLI をインストールし、Microsoft Azure で新規リソースを設定する手順は、<https://access.redhat.com/articles/2706961#install-the-azure-cross-platform-cli-2> にあります。

Microsoft Azure VHD をアップロードする前に、Microsoft Azure で作成した仮想マシンのリソースが必要です。以下の手順はすべて、Microsoft Azure Administration Server から完了します。

1. リソースグループを作成します。

```
# azure group create -l [azure-region] -n [resource-group]
```

以下に例を示します。

```
# azure group create -l eastus -n rhgsarm
info: Executing command group create
+ Getting resource group rhgsarm
+ Creating resource group rhgsarm
info: Created resource group rhgsarm
data: Id: /subscriptions/2586c64b-38b4-4527-a140-012d49dfc02c/resourceGroups/rhgsarm
```

```

data: Name: rhgsarm
data: Location: eastus
data: Provisioning State: Succeeded
data: Tags: null
data:
info: group create command OK

```

2. ストレージアカウントを作成します。以下のコマンドは、使用するレプリケーションオプションを特定できるいくつかの詳細を設定します。詳細は、<https://azure.microsoft.com/en-us/documentation/articles/storage-redundancy/> の Microsoft の『Account Replication Options』の記事を参照してください。

```

# azure storage account create --sku-name [LRS, ZRS, GRS, RA-GRS] --kind [Storage, Blob Storage] -l [azure-region] -g [resource-group] [account-name]

```

以下に例を示します。

```

# azure storage account create --sku-name LRS --kind Storage -l eastus -g rhgsarm
rhgsstorage
info: Executing command storage account create
+ Checking availability of the storage account name
+ Creating storage account
info: storage account create command OK

```

3. 次の手順で使用するストレージアカウントキーを取得します。

```

# azure storage account keys list -g [resource-group] [account-name]

```

以下に例を示します。

```

# azure storage account keys list -g rhgsarm rhgsstorage
info: Executing command storage account keys list
+ Getting storage account keys
data: Name Key Permissions
data: -----
data: key1
ba8zt8LSUznaGX92DV1zWhj3ikrpIQEsZXxmNQcd194JaD/NgSwueVVmPfAjOMptpu3fDR/7ol
T2smo/9ZON+w== Full
data: key2
5r+wVkfzb8S6oBpZCo9vQSuVhyNLuaOcfpOR048zWnRr8kAsTs7Vp72C/ldWGGQE9liKrFXx
PaDb5GosaA3Zeg== Full
info: storage account keys list command OK

```

Microsoft Azure は、ストレージコンテナへのアクセスを許可する鍵の 2 つを生成します。Microsoft Azure は、キーの再生成目的で 2 つのキーを生成します。再生成時に、セキュアなアクセスに 1 つのキーを使用します。一方、Microsoft Azure はキーを再生成して新規キーを作成します。新しいキーがある場合は、そのキーに切り替えて、他の鍵を次の時間に再生成します。これは、Microsoft Azure ストレージコンテナのパスワードを変更するのに適していません。これは定期的に行うべきですが、2 つのキーのみが作成されているため、これは実行しません。キーおよびキーの再生成に関する情報は、<https://azure.microsoft.com/en-us/documentation/articles/storage-configure-connection-string/> の Microsoft の Storage Connection Strings を参照してください。

4. キーをエクスポートします。前の手順で作成した key1 文字列をコピーし、AccountKey= ファイルに貼り付けます。

```
# export
AZURE_STORAGE_CONNECTION_STRING="DefaultEndpointsProtocol=https;AccountName=[account-name];AccountKey=[storage-account-key]"
```

以下に例を示します。

```
# export
AZURE_STORAGE_CONNECTION_STRING="DefaultEndpointsProtocol=https;AccountName=rhgsstorage;AccountKey=ba8zt8LSUznaGX92DV1zWhj3ikrplQEsZXmNQcd194JaD/NgSwueVVmPfAjOMptpu3fDR/7olT2smo/9ZON+w=="
```

5. ストレージコンテナを作成します。

```
# azure storage container create [container-name]
```

以下に例を示します。

```
# azure storage container create rhgscontainer
info: Executing command storage container create
+ Creating storage container rhgscontainer
+ Getting storage container information
data: {
  data: name: 'rhgscontainer',
  data: metadata: {},
  data: etag: '"0x8D44F1EAD8604B4"',
  data: lastModified: 'Tue, 07 Feb 2017 06:00:54 GMT',
  data: lease: { status: 'unlocked', state: 'available' },
  data: requestId: '124a5521-0001-007a-3507-81b138000000',
  data: publicAccessLevel: 'Off'
  data: }
info: storage container create command OK
```

6. Microsoft Azure 仮想ネットワークを作成します。

```
# azure network vnet create -g [resource-group] -l [azure-region] -a [CIDR-address-prefix-vnet] [vnet-name]
```

以下に例を示します。

```
# azure network vnet create -g rhgsarm -l eastus -a 10.0.0.0/8 rhgsvnet1
info: Executing command network vnet create
+ Looking up the virtual network "rhgsvnet1"
+ Creating virtual network "rhgsvnet1"
data: Id :
/subscriptions/2586c64b-38b4-4527-a140-012d49dfc02c/resourceGroups/rhgsarm/providers/Microsoft.Network/virtualNetworks/rhgsvnet1

data: Name : rhgsvnet1
data: Type : Microsoft.Network/virtualNetworks
data: Location : eastus
data: Provisioning state : Succeeded
```

```
data: Address prefixes:
data: 10.0.0.0/8
info: network vnet create command OK
```

7. Microsoft Azure 仮想ネットワークのサブネットを作成します。

```
# azure network vnet subnet create -g [resource-group] -e [vnet-name] -a [CIDR-address-
prefix-subnet] [subnet-name]
```

以下に例を示します。

```
# azure network vnet subnet create -g rhgsarm -e rhgsvnet1 -a 10.0.1.0/24 rhgssubnet1
info: Executing command network vnet subnet create
+ Looking up the virtual network "rhgsvnet1"
+ Looking up the subnet "rhgssubnet1"
+ Creating subnet "rhgssubnet1"
data: Id : /subscriptions/2586c64b-38b4-4527-a140-
012d49dfc02c/resourceGroups/rhgsarm/providers/Microsoft.Network/virtualNetworks/rhgsvnet1
/subnets/rhgssubnet1
data: Name : rhgssubnet1
data: Provisioning state : Succeeded
data: Address prefix : 10.0.1.0/24
info: network vnet subnet create command OK
```



重要

マシンをシャットダウンしない場合には、ストレージ接続キーは以下の手順により維持されます。この手順を行う前にシャットダウンした場合は、以下のコマンドでキー (-k "[storage-account-key]) を渡す必要があります。キーを設定ファイルに追加して、マシンのブート時に毎回読み込むことができます。

3.4.3. ディスクイメージの Microsoft Azure へのアップロード

ディスクイメージはアップロードされ、Gluster Storage ノードを作成するテンプレートとして使用できます。イメージを Microsoft Azure にアップロードするには、VHD イメージが保管されているディレクトリに移動し、以下のコマンドを実行します。

```
# azure storage blob upload -t page -a [account-name] --container [container-name] -b [azure-image-
name].vhd [path to image-name].vhd
```



注記

アップロード速度の速度によっては、これには時間がかかる場合があります。

以下に例を示します。

```
# azure storage blob upload -t page -a rhgsstorage --container rhgscontainer -b rhgsimage72.vhd
rhgs-azure-cluster.vhd
info: Executing command storage blob upload
+ Checking blob rhgsimage72.vhd in container rhgscontainer
+ Uploading rhgs-azure-cluster.vhd to blob rhgsimage72.vhd in container rhgscontainer
+ Getting storage blob information
```

```

data: Property  Value
data: -----
data: container  rhgscontainer
data: name       rhgsimage72.vhd
data: blobType   PageBlob
data: contentLength 21475885568
data: contentType application/octet-stream
data: contentMD5  G965mhnwQtBZVI6oRW3tpw==
info: storage blob upload command OK

```

3.4.4. Gluster Storage インスタンスのデプロイ

Microsoft Azure の個別の Gluster Storage インスタンスをクラスターに設定できます。まず、準備したイメージからインスタンスを作成してから、データディスクを割り当てる必要があります。

インスタンスの作成時に、以下の2つの方法を使用してインスタンスにアクセスできます。

- SSH 鍵を使用した認証
 - パスワードを使用した認証
1. Red Hat Gluster Storage 仮想マシンを作成します。



注記

このコマンドの実行時に `vnic-name`、`public-ip-name`、および `public-ip-domain-name` が作成されます。使用する名前を入力し、コマンドの実行時に作成されます。`-Q` オプションの直前の手順でイメージの URL をコピーアンドペーストします。`-G` オプションは ssh キーを生成します。表示される詳細で `key-pem` ファイルへのパスを書き留めておきます。Azure 仮想マシンへの ssh へのパスが必要です。



重要

Microsoft Azure 仮想マシン名には、文字、数字、ピリオド、およびハイフンを含めることができます。名前はハイフンで起動したり終了したり、数字を完全に構成したり、512 文字以下にする必要があります。詳細は、Microsoft の「『Recommended naming conventions for Azure resources』」を参照してください。<https://docs.microsoft.com/en-us/azure/guidance/guidance-naming-conventions>

SSH 鍵を使用した認証

- a. 以下のコマンドを入力して、Red Hat Gluster Storage インスタンスを作成し、SSH 鍵を使用して認証します。

```

azure vm create -g [resource-group] -l [azure-region] -f [vnic-name] -y [os-type] -Q
[image-url] -G -z [vm-size] -i [public-ip-name] -o [storage-account-name] -R [storage-
container-name] -m [public-ip-allocation-method] -w [public-ip-domain-name] -t [public-ip-
idletimeout] -F [vnet-name] -j [vnet-subnet-name] -u [admin-name] -n [virtual-machine-
name]

```

以下に例を示します。

```
# azure vm create -g rhgsarm -l eastus -f rhgsimage72vnic -y Linux -Q
https://rhgsstorage.blob.core.windows.net:443/rhgscontainer/rhgsimage72.vhd -G -z
Standard_A2 -i rhgs72_pub -o rhgsstorage -R rhgscontainer -m dynamic -w rhgs72 -t 10
-F rhgsvnet1 -j rhgssubnet1 -u clouduser -n rhgs72
info: Executing command vm create
+ Looking up the VM "rhgs72"
info: You can use /root/.azure/ssh/rhgs72-key.pem private key for SSH authentication.
info: Verifying the public key SSH file: /root/.azure/ssh/rhgs72-cert.pem
info: Using the VM Size "Standard_A2"
info: The [OS, Data] Disk or image configuration requires storage account
+ Looking up the storage account rhgsstorage info: Using
"https://rhgsstorage.blob.core.windows.net:443/rhgscontainer/rhgsimage72.vhd" as the
user image.
+ Looking up the NIC "rhgsimage72vnic" info: Found an existing
NIC "rhgsimage"
info: Found an IP configuration with virtual network subnet id "/subscriptions/2586c64b-
38b4-4527-a140-
012d49dfc02c/resourceGroups/rhgsarm/providers/Microsoft.Network/virtualNetworks/rhgsv
net1/subnets/rhgssubnet1" in the NIC "rhgsimage72vnic"
info: This NIC IP configuration is already configured with the provided public ip
"rhgs72_pub"
info: The storage URI 'https://rhgsstorage.blob.core.windows.net/' will be used for boot
diagnostics settings, and it can be overwritten by the parameter input of '--boot-
diagnostics-storage-uri'.
+ Creating VM "rhgs72"
info: vm create command OK
```

- b. SSH セッションを開始し、管理者名と公開鍵ファイルを使用して、実行中の仮想マシンに接続します。

```
# ssh -i [path-to-key-pem] [admin-name@public-ip-address]
```

以下に例を示します。

```
# ssh -i /root/.azure/ssh/rhgs72-key.pem clouduser@rhgs72.east.cloudapp.azure.com
```

パスワードを使用した認証

- a. 以下のコマンドを入力して、Red Hat Gluster Storage インスタンスを作成し、SSH パスワードを使用して認証します。プロンプトが表示されたら、パスワードを入力する必要があります。

```
# azure vm create -g [resource-group] -l [azure-region] -f [vnic-name] -y [os-type] -Q
[image-url] -z [vm-size] -i [public-ip-name] -o [storage-account-name] -R [storage-
container-name] -m [public-ip-allocation-method] -w [public-ip-domain-name] -t [public-ip-
idletimeout] -F [vnet-name] -j [vnet-
subnet-name] -u [admin-name] -n [virtual-machine-name]
```

以下に例を示します。

```
# azure vm create -g rhgsarm732 -l eastus -f rhgs732vnic -y Linux -Q
https://rhgsstorage732.blob.core.windows.net/rhgscontainer732/rhgsimage732.vhd -z
Standard_A2 -i rhgs732_pub -o rhgsstorage732 -R rhgscontainer732 -m dynamic -w
rhgs7322 -t 10 -F rhgsvnet732 -j rhgssubnet732 -u clouduser -n rhgs732-2
```



```

info: Executing command vm create
+ Looking up the VM "rhgs732-2"
Enter password for clouduser: *****
Confirm password: *****
info: Using the VM Size "Standard_A2"
info: The [OS, Data] Disk or image configuration requires storage account
+ Looking up the storage account rhgsstorage732
info: Using
"https://rhgsstorage732.blob.core.windows.net/rhgscontainer732/rhgsimage732.vhd" as
the user image.
+ Looking up the NIC "rhgs732vnic"
info: Found an existing NIC "rhgs732vnic"
info: Found an IP configuration with virtual network subnet id "/subscriptions/2586c64b-
38b4-4527-a140-
012d49dfc02c/resourceGroups/rhgsarm732/providers/Microsoft.Network/virtualNetworks/r
hgsvnet732/subnets/rhgssubnet732" in the NIC "rhgs732vnic"
info: This NIC IP configuration is already configured with the provided public ip
"rhgs732_pub"
info: The storage URI 'https://rhgsstorage732.blob.core.windows.net/' will be used for
boot diagnostics settings, and it can be overwritten by the parameter input of '--boot-
diagnostics-storage-uri'.
+i reating VM "rhgs732-2"
info: vm create command OK

```

- b. SSH セッションを開始し、管理者名とインスタンスの作成時に使用されるパスワードを使用して、実行中の仮想マシンに接続します。

```
# ssh [admin-name@public-ip-address]
```

以下に例を示します。

```
# ssh clouduser@rhgs72.east.cloudapp.azure.com
```

2. 新規作成された仮想マシンにデータディスクを追加します。

```
# azure vm disk attach-new resource-group vm-name size-in-gb
```

以下に例を示します。

```

# azure vm disk attach-new rhgsarm rhgs72 1023
info: Executing command vm disk attach-new
+ Looking up the VM "rhgs72"
info: New data disk location:
https://rhgsstorage.blob.core.windows.net/rhgscontainer/rhgs72-20170220-121452858.vhd
+ Updating VM "rhgs72"
info: vm disk attach-new command OK

```

3. インスタンスを作成し、すべてのインスタンスにディスクを割り当てる上記のステップを実行します。
4. インスタンスが適切に作成されたことを確認します。

```
# azure vm list -g [resource group]
```

-

For example,

```
# azure vm list rhgsarm
info: Executing command vm list
+ Getting virtual machines
data: ResourceGroupName Name ProvisioningState PowerState Location Size
data: -----
data: rhgsarm rhgs72 Succeeded VM deallocated eastus Standard_A2
data: rhgsarm rhgs72-1 Succeeded VM deallocated eastus Standard_A2
info: vm list command OK

# azure vm show -g rhgsarm rhgs72
info: Executing command vm show
+ Looking up the VM "rhgs72"
+ Looking up the NIC "rhgsimage72vnic"
+ Looking up the public ip "rhgs72_pub"
Id :/subscriptions/2586c64b-38b4-4527-a140-012d49dfc02c/resourceGroups/rhgsarm/providers/Microsoft.Compute/virtualMachines/rhgs72d
ata: ProvisioningState :Succeeded
data: Name :rhgs72
data: Location :eastus
data: Type :Microsoft.Compute/virtualMachines data:
data: Hardware Profile:
data: Size :Standard_A2
data:
data: Storage Profile:
data:
data: OS Disk:
data: OSType :Linux
data: Name :clib438d5640249ac7c-os-1486533879443
data: Caching :ReadWrite
data: CreateOption :FromImage
data: Vhd:
data: Uri
:https://rhgsstorage.blob.core.windows.net/rhgscontainer/clib438d5640249ac7c-os-1486533879443.vhd
data:
data: OS Profile:
data: Computer Name :rhgs72
data: User Name :clouduser
data: Secrets :[]
data: Linux Configuration:
data: Disable Password Auth :true
data:
data: Network Profile:
data: Network Interfaces:
data: Network Interface #1:
data: Primary :true
data: MAC Address :00-0D-3A-13-36-A0
data: Provisioning State :Succeeded
data: Name :rhgsimage72vnic
data: Location :eastus
data: Public IP address :40.121.209.95
data: FQDN :rhgs72.eastus.cloudapp.azure.com
data:
```

```

data: Diagnostics Profile:
data: BootDiagnostics Enabled :true
data: BootDiagnostics StorageUri:https://rhgsstorage.blob.core.windows.net/
data:
data: Diagnostics Instance View:
info: vm show command OK

```

- Microsoft Azure の可用性セットは、保持するインスタンスに対するフォールトトレランスのレベルを提供し、システム障害や予定されている停止から保護します。これは、同じ可用性セット内のインスタンスが異なる障害や、Microsoft Azure データセンター内のアップグレードドメインをデプロイすることで実現されます。
- Gluster Storage がブリック間でデータを複製する場合は、レプリカセットを特定の可用性セットに関連付けます。レプリケーション設計で可用性セットを使用すると、Microsoft Azure インフラストラクチャー内のインシデントはレプリカセットのすべてのメンバーを同時に影響を受けることができません。

3.5. GLUSTER STORAGE クラスターの設定

信頼できるストレージプール(クラスター)を形成するようにこれらのインスタンスを設定します。



注記

Red Hat Enterprise Linux 7 マシンを使用している場合は、Azure ポータルにログインして仮想マシンのパスワードをリセットし、仮想マシンも再起動します。Red Hat Enterprise Linux 6 マシンでは、パスワードのリセットは必要ありません。

1. キーまたはパスワードを使用して各ノードにログインします。

```

# ssh -i [path-to-key-pem] [admin-name@public-ip-address]

or

# ssh [admin-name@public-ip-address]

```

以下に例を示します。

```

# ssh -i /root/.azure/ssh/rhgs72-key.pem clouduser@rhgs72.east.cloudapp.azure.com

or

# ssh clouduser@rhgs72.east.cloudapp.azure.com

```

2. **subscription-manager** コマンドを使用して各ノードを Red Hat Network に登録し、関連する Red Hat Storage サブスクリプションをアタッチします。

Red Hat Gluster Storage 3.5 チャンネルをサブスクライブする方法については、『[Red Hat Gluster Storage 3.5 Installation Guide](#)』の『Installing Red Hat Gluster Storage』を参照してください。

3. 各ノードを更新して、最新の拡張機能およびパッチが適用されていることを確認します。

```

# yum update

```

- 『『Red Hat Gluster Storage Administration Guide』』の「『信頼できるストレージプールへのサーバーの追加』」の章に記載の手順に従って、信頼できるストレージプールを作成します。

3.6. 付録: ASM モードでの MICROSOFT AZURE での RED HAT GLUSTER STORAGE の設定

このセクションでは、Microsoft Azure で Red Hat Gluster Storage を設定する手順を説明します。

3.6.1. Red Hat Gluster Storage for Microsoft Azure の取得

Red Hat サブスクリプションまたは Red Hat 評価サブスクリプションを使用して Red Hat Gluster Storage Server ファイルをダウンロードするには、以下を実行します。

- Red Hat カスタマーポータル (<https://access.redhat.com/login>) にアクセスし、ユーザー名およびパスワードを入力してログインします。
- Downloads** をクリックし、**Software & Download Center** に移動します。
- Red Hat Gluster Storage Server エリアで、**Download Software** をクリックして **VHD** イメージの最新バージョンをダウンロードします。
- ファイルがダウンロードされたディレクトリーに移動し、ファイルで **sha256sum** コマンドを実行します。

以下に例を示します。

```
$ sha256sum rhgs-azure-3.5-rhel-7-x86_64.tar.gz
2d083222d6a3c531fa2fbbd21c9ea5b2c965d3b8f06eb8ff3b2b0efce173325d rhgs-azure-3.5-rhel-7-x86_64.tar.gz
```

sha256sum ユーティリティーで生成された値は、ファイルについて Red Hat カスタマーポータルに表示される値と一致している必要があります。これらが同じでない場合は、ダウンロードが不完全または破損しているため、ファイルを再度ダウンロードする必要があります。ダウンロードの試行後にチェックサムが正常に検証されていない場合は、Red Hat サポートにお問い合わせください。

- ダウンロードしたファイル **rhgs-azure-[version].zip** を展開してアーカイブコンテンツを展開します。

以下に例を示します。

```
# tar -xvzf rhgs-azure-3.5-rhel-7-x86_64.tar.gz
```

3.6.2. ネットワークポロジの定義

デフォルトでは、クラウドサービスにインスタンスをデプロイすると、動的に割り当てられた内部 IP アドレスが選択されます。このアドレスは変更でき、サイトからサイトにより異なります。設定によっては、インスタンスが接続できるように、アカウント内に1つまたは複数の仮想ネットワークを定義することを検討してください。これにより、オンプレミス環境と同様のネットワーク設定を確立します。

シンプルなネットワークを作成するには、以下を実行します。

- Gluster Storage ノードのクラウドサービスを作成します。

```
# azure service create --serviceName service_name --location location
```

以下に例を示します。

```
# azure service create --serviceName rhgs313-cluster --location "East US"
info: Executing command service create
+ Creating cloud service
data: Cloud service name rhgs313-cluster
info: service create command OK
```

cloudapp.net がサービス名に追加され、完全なサービス名がインターネットに直接公開されま
す。この場合は、**rhgs313-cluster.cloudapp.net** になります。

2. Gluster Storage ノードが接続する仮想ネットワークを作成します。この例では、ネットワーク
が East US の場所内に作成されます。

```
# azure network vnet create --vnet "rhgs313-vnet" --location "East US" --address-space
10.18.0.0 --cidr 16
info: Executing command network vnet create
info: Using default subnet start IP: 10.18.0.0
info: Using default subnet cidr: 19
+ Looking up network configuration
+ Looking up locations
+ Setting network configuration
info: network vnet create command OK
```

これは単一のリージョン内のネットワークを定義します。

Gluster Storage 内の Geo レプリケーションなどの機能には vnet-to-vnet の設定が必要です。vnet-to-
vnet 設定は、VPN ゲートウェイを介して仮想ネットワークに接続します。各仮想ネットワークは、障
害回復シナリオに対処するために、同じリージョン内またはリージョン全体で指定することが可能で
す。VPN を結合するには共有キーが必要であり、Microsoft Azure CLI で共有鍵を渡すことはできませ
ん。vnet-to-vnet 設定を定義するには、Windows Powershell を使用するか、Microsoft Azure REST API
を使用します。

3.6.3. ディスクイメージを Microsoft Azure にアップロードする

ディスクイメージはアップロードされ、Gluster Storage ノードを作成するテンプレートとして使用で
きます。



注記

Microsoft Azure コマンドは、xplat-cli を使用するよう設定されたローカルアカウント
から発行する必要があります。

イメージを Microsoft Azure にアップロードするには、VHD イメージが保管されているディレクトリー
に移動し、以下のコマンドを実行します。

```
# azure vm image create image_name --location location --os linux VHD_image_name
```

以下に例を示します。

```
# azure vm image create rhgs-3.1.3 --location "East US" --os linux rhgs313.vhd
```

```

info: Executing command vm image create
+ Retrieving storage accounts
info: VHD size : 20 GB
info: Uploading 20973568.5 KB
Requested:100.0% Completed:100.0% Running: 0 Time: 7m50s Speed: 3876 KB/s
info: https://bauderhel7.blob.core.windows.net/vm-images/rhgs313.vhd was uploaded successfully
info: vm image create command OK

```

完了したら、イメージが利用可能であることを確認します。

```
# azure vm image list | awk '$3 == "User" {print $2;}'
```



注記

インスタンスイメージ一覧の出力は、パブリックイメージとアカウントに固有のイメージ (User) を表示するため、**awk** は Microsoft Azure アカウント下に追加されたイメージのみを表示するために使用されます。

3.6.4. Gluster Storage インスタンスのデプロイ

Microsoft Azure の個別の Gluster Storage インスタンスをクラスターに設定できます。まず、準備したイメージからインスタンスを作成してから、データディスクを割り当てる必要があります。

1. 準備済みイメージからインスタンスを作成するには、

```
# azure vm create --vm-name vm_name --availability-set name_of_the_availability_set --vm-size size --virtual-network-name vnet_name --ssh port_number --connect cluster_name username_and_password
```

以下に例を示します。

```
# azure vm create --vm-name rhgs313-1 --availability-set AS1 -S 10.18.0.11 --vm-size Medium --virtual-network-name rhgs313-vnet --ssh 50001 --connect rhgs313-cluster rhgs-3.1.3 rhgsuser 'AzureAdm1n!'
info: Executing command vm create
+ Looking up image rhgs-313
+ Looking up virtual network
+ Looking up cloud service
+ Getting cloud service properties
+ Looking up deployment
+ Creating VM
info: OK
info: vm create command OK
```

2. 各インスタンスに 1023 GB のデータディスクを追加します。

```
# azure vm disk attach-new VM_name 1023
```

たとえば、以下のようになります。

```
# azure vm disk attach-new rhgs313-1 1023
info: Executing command vm disk attach-new
+ Getting virtual machines
```

```
+ Adding Data-Disk
info:  vm disk attach-new command OK
```

3. インスタンスを作成し、すべてのインスタンスにディスクを割り当てる上記のステップを実行します。
4. インスタンスが適切に作成されたことを確認します。

```
# azure vm list
# azure vm show vm-name
```

- Microsoft Azure の可用性セットは、保持するインスタンスに対するフォールトトレランスのレベルを提供し、システム障害や予定されている停止から保護します。これは、同じ可用性セット内のインスタンスが異なる障害や、Microsoft Azure データセンター内のアップグレードドメインをデプロイすることで実現されます。
- Gluster Storage がブリック間でデータを複製する場合は、レプリカセットを特定の可用性セットに関連付けます。レプリケーション設計で可用性セットを使用すると、Microsoft Azure インフラストラクチャー内のインシデントはレプリカセットのすべてのメンバーを同時に影響を受けることができません。
- 各インスタンスには、SSH アクセスを許可するためのクラウドサービスに追加される静的 IP (-S) 内の静的 IP (-S) が割り当てられ、SSH アクセスを許可できます (--ssh ポート)。
- bash の解釈の問題を回避するために、パスワードの周りの引用符 (') があります。

例

以下は、準備したイメージから 4 つのインスタンスを作成する例です。

- これらの名前は **rhgs31-n** です。
- これらの IP アドレスは 10.18.0.11 から 10.18.0.14 です。

インスタンスが作成される (**azurevm create**) になり、同じ可用性セットに追加できます (--**availability-set**)。

```
for i in 1 2 3 4; do as=$((i/3)); azure vm create --vm-name rhgs31-$i --availability-set AS$as -S
10.18.0.1$i --vm-size Medium --virtual-network-name rhgs-vnet --ssh 5000$i --connect rhgs-cluster
rhgs3.1 rhgsuser 'AzureAdm1n!'; done
```

各インスタンスに 4 つの 1023 GB のデータディスクを追加します。

```
for node in 1 2 3 4; do for disk in 1 2 3 4; do azure vm disk attach-new rhgs31-$node 1023; done ;
done
```

インスタンスが適切に作成されたことを確認します。

```
# azure vm list
# azure vm show vm-name
```



注記

この例では静的 IP アドレスを使用しますが、これは必須ではありません。単一 Gluster Storage クラスタを作成し、geo レプリケーションなどの機能が必要ない場合は、Microsoft Azure によって自動的に割り当てられる動的 IP を使用できます。唯一の重要なことは、Gluster Storage クラスタは名前前で定義されます。

3.6.5. Gluster Storage クラスタの設定

信頼できるストレージプール (クラスタ) を形成するようにこれらのインスタンスを設定します。



注記

Red Hat Enterprise Linux 7 マシンを使用している場合は、Microsoft Azure ポータルにログインして仮想マシンのパスワードをリセットし、仮想マシンも再起動します。Red Hat Enterprise Linux 6 マシンでは、パスワードのリセットは必要ありません。

1. 各ノードにログインします。

```
# ssh rhgsuser@rhgs313-cluster.cloudapp.net -p 50001
```

2. **subscription-manager** コマンドを使用して各ノードを Red Hat Network に登録し、関連する Red Hat Storage サブスクリプションをアタッチします。

Red Hat Gluster Storage 3.5 チャンネルをサブスクライブする方法については、『[Red Hat Gluster Storage 3.5 Installation Guide](#)』の『Installing Red Hat Gluster Storage』を参照してください。

3. 各ノードを更新して、最新の拡張機能およびパッチが適用されていることを確認します。

```
# yum update
```

4. 『『Red Hat Gluster Storage Administration Guide』』の「『信頼できるストレージプールへのサーバーの追加』」の章に記載の手順に従って、信頼できるストレージプールを作成します。

3.7. 付録: ISO からのカスタムディスクイメージの作成

Microsoft Azure 内のインスタンスは、ディスクイメージから作成されます。Gluster Storage には、デフォルトの Microsoft Azure が提供するイメージではなく、カスタムイメージが必要です。カスタム仮想マシンイメージの構築は通常 Hyper-V で実行されますが、Microsoft Azure のカスタムイメージは、ネイティブ Linux ツールを使用してビルドすることもできます。

カスタムイメージを設定するための全体的なプロセスには約 30 分かかります。

1. Gluster Storage の最新 ISO を https://access.redhat.com/downloads/content/186/ver=3.1/rhel---7/3.1/x86_64/product-software からダウンロードします。
2. **virt-manager** を使用して、2つのコアである 4 GB RAM、2つのコア 20 GB の virtio HDD と 1つの NIC を持つ qcow2 イメージを作成します。

- ISO イメージからインスタンスを起動し、Gluster Storage のインストールを完了します。Windows Azure エージェントはスワップ領域用に一時ディスクを設定するため、スワップ領域を割り当てないでください。
- インスタンスを再起動してログインします。
- 汎用ホスト名を設定します。

Red Hat Enterprise Linux 7 の場合:

```
# hostnamectl set-hostname localhost.localdomain
```

Red Hat Enterprise Linux 6 の場合:

```
# vim /etc/sysconfig/network  
  
NETWORKING=yes  
HOSTNAME=localhost.localdomain
```

- DHCP が `/etc/sysconfig/network-scripts/ifcfg-eth0` に設定されていることを確認します。

```
DEVICE=eth0  
ONBOOT=yes  
BOOTPROTO=dhcp  
IPV6INIT=no  
TYPE=Ethernet  
USERCTL=no  
PEERDNS=yes
```

- Microsoft Azure および Hyper-V との競合を回避するために、udev ルールを更新します。

```
# rm -f /etc/udev/rules.d/70-persistent-net.rules  
# rm -f /lib/udev/rules.d/75-persistent-net-generator.rules
```

- Red Hat Enterprise Linux 7 では、Gluster Storage にデフォルトのファイアウォールルールを適用します。これらのルールセットは、ノード間の通信、GlusterFS クライアント、および NFS に使用されます。

```
# firewall-cmd --zone=public --add-service=glusterfs --permanent  
# firewall-cmd --zone=public --add-service=nfs --add-service=rpc-bind --permanent
```

- 仮想マシンを登録します。

```
# subscription-manager register --auto-attach  
# subscription-manager repos --disable=*
```

- Extras リポジトリおよび Gluster Storage リポジトリを有効にします。これは rhel-6- または rhel-7- のいずれかになります。

```
# subscription-manager repos --enable rhel-7-server-rpms --enable rhel-7-server-extras-rpms  
--enable rh-gluster-3-for-rhel-7-server-rpms
```

- システムを更新して、Microsoft Azure Linux エージェントをインストールします。

```
# yum update -y
# yum -y install WALinuxAgent
```

12. Gluster Storage のインストール時に定義されたスワップ領域を無効にします。これは Red Hat Enterprise Linux 7 で必要になります。Microsoft Azure は、ランタイム時に一時ストレージを割り当てます。このストレージは swap に使用されるため、スワップ領域を明示的に定義する必要はありません。

```
# swapoff -v /dev/rhgs/swap
# sed -i '/* swap/d' /etc/fstab
```

Red Hat Enterprise Linux 6 では、インストーラーはディスク設定を変更できるため、スワップが定義されていません。ただし、論理ボリュームを作成している場合は、RHEL 7 で設定を削除します。

13. **Red Hat Enterprise Linux 7 のみ** Azure で実行されている Linux 仮想マシンには、initramfs イメージ内に hv_storvsc ドライバーおよび hv_vmbus ドライバーが必要です。Red Hat Enterprise Linux 6 インストーラーにはこれらのドライバーが自動的に含まれていますが、Red Hat Enterprise Linux 7 では、インストーラーは Hyper-V デバイスがインストール時に検出される場合にのみこれらのドライバーを追加します。virt-manager を使用して仮想マシンイメージをビルドする場合は、これらの Hyper-V ドライバーを手動で追加します。
 - a. 以下の内容を **/etc/dracut.conf** に追加します。
 - b. **initramfs** を再生成します。

```
# dracut -f -v
```

14. カーネルブート設定を更新します。

Red Hat Enterprise Linux 7 の場合:

- a. /etc/default/grub に GRUB_CMDLINE_LINUX 変数を設定します。

```
``rd.lvm.lv=rhgs/root console=ttyS0 earlyprintk=ttyS0 rootdelay=300
```

- b. grub2 設定を更新します。

```
# grub2-mkconfig -o /boot/grub2/grub.cfg
```

- c. rhqb、quiet、または crashkernel=auto パラメーターを削除します。

Red Hat Enterprise Linux 6 の場合:

- a. /boot/grub/menu.lst でカーネルブート行を更新します。

```
console=ttyS0 earlyprintk=ttyS0 rootdelay=300 numa=off
```

- b. rhqb、quiet、または crashkernel=auto パラメーターを削除します。

15. 起動時に Windows Azure エージェントを起動できるようにします。

- Red Hat Enterprise Linux 7 の場合:

-

```
# systemctl enable waagent
```

- Red Hat Enterprise Linux 6 の場合:

```
# chkconfig waagent on
```

16. Red Hat Subscription Manager を使用して仮想マシンの登録を解除します。

```
# subscription-manager unregister
```

17. インスタンスのプロビジョニングを解除してローカル設定を削除します。これにより、インスタンスは Microsoft Azure 内のディスクイメージとして使用することができます。

```
# yum clean all
# waagent -force -deprovision
# export HISTSIZE=0
# poweroff
```

18. インスタンスの XML をダンプし、作成した仮想ディスクのファイル名を見つけ、Microsoft Azure と互換性のある VHD ファイルに変換します。この例では、インスタンスは **qcow2** ディスク形式を使用して最初に作成されました。

```
# virsh dumpxml image-name
# qemu-img convert -f qcow2 -O vpc -o subformat=fixed -O vpc rhgs313.qcow2 rhgs313.vhd
```

3.8. 付録: PERFORMANCE CATEGORIZATION

Microsoft Azure 内で Red Hat Gluster Storage が提供できる可能性のあるパフォーマンスを判断するインフラストラクチャーおよびアーキテクチャー要素が多数あります。

3.8.1. ストレージタイプ

Microsoft Azure には、標準およびプレミアムの物理ストレージの2つのクラスがあります。標準ストレージはハードディスクドライブでサポートされていますが、プレミアムストレージはソリッドステートドライブで配信されます。これらのストレージクラスは、ディスクにそれぞれ 500 IOPS と 5,000 IOPS の IOPS ターゲットを提供します。

より一般的な考慮点は、データを保護する方法です。デフォルトでは、Microsoft Azure は、別の障害ドメインにデータの3つのコピーを同期的に保存し、データをセカンダリーデータセンター（デフォルトの GRS レプリケーションスキーム）に非同期的に配置します。

3.8.2. Bandwidth

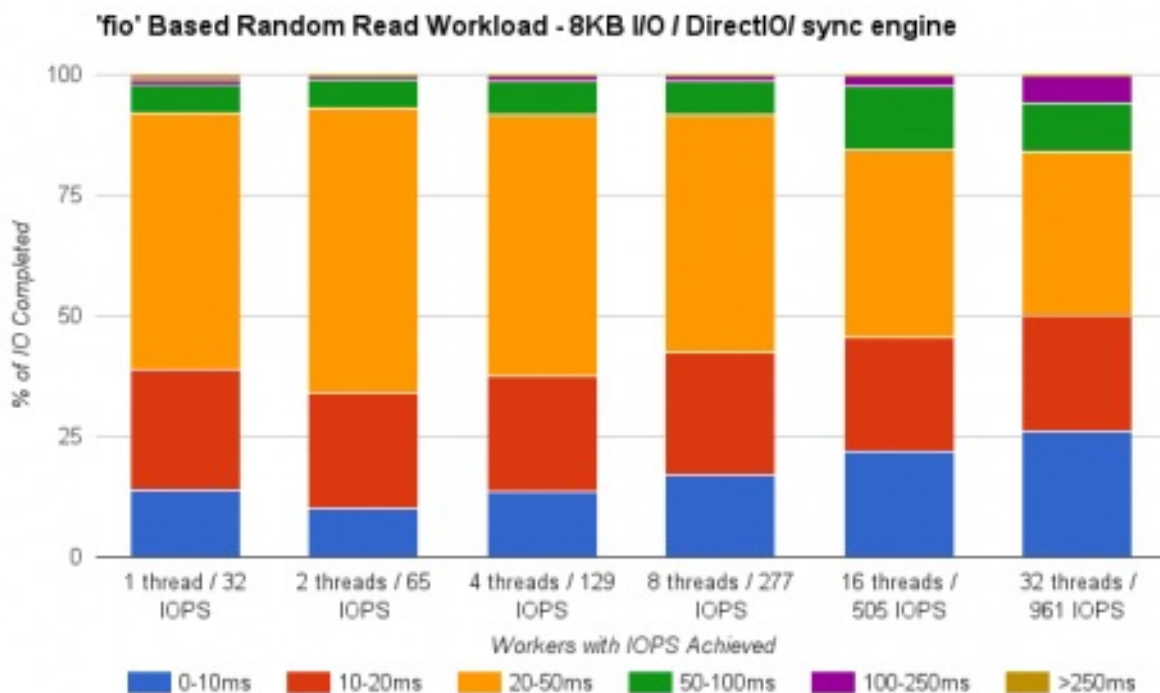
クライアントと Red Hat Gluster Storage ノード間の上限を判断するために、**iperf** を使用して単純なテストが実行されました。このテストでは、単一のネットワークインターフェースが 600 - 700 Mbit の間で配信されることが予想されていることが分かります。

3.8.3. ディスクライセンス

4つのディスクは Standard Tier/A2 インスタンスに割り当てられ、**mdadm** ツールを使用して RAID0 セットを形成するために集約されました。LUN は、LVM、dm-thinp、および XFS ファイルシステムに基づいて、推奨されるベストプラクティスを使用して設定されていました。その後、**fiio** ツールを使用

して、同時実行レベルが増えたときに、基礎となるディスクの無作為な読み取りプロファイルを表示していました。

図3.3 ディスクライセンス



このベンチマークは具体的な結果を生成しませんが、基礎となるストレージの潜在的なI/Oプロファイルを示します。

注意点

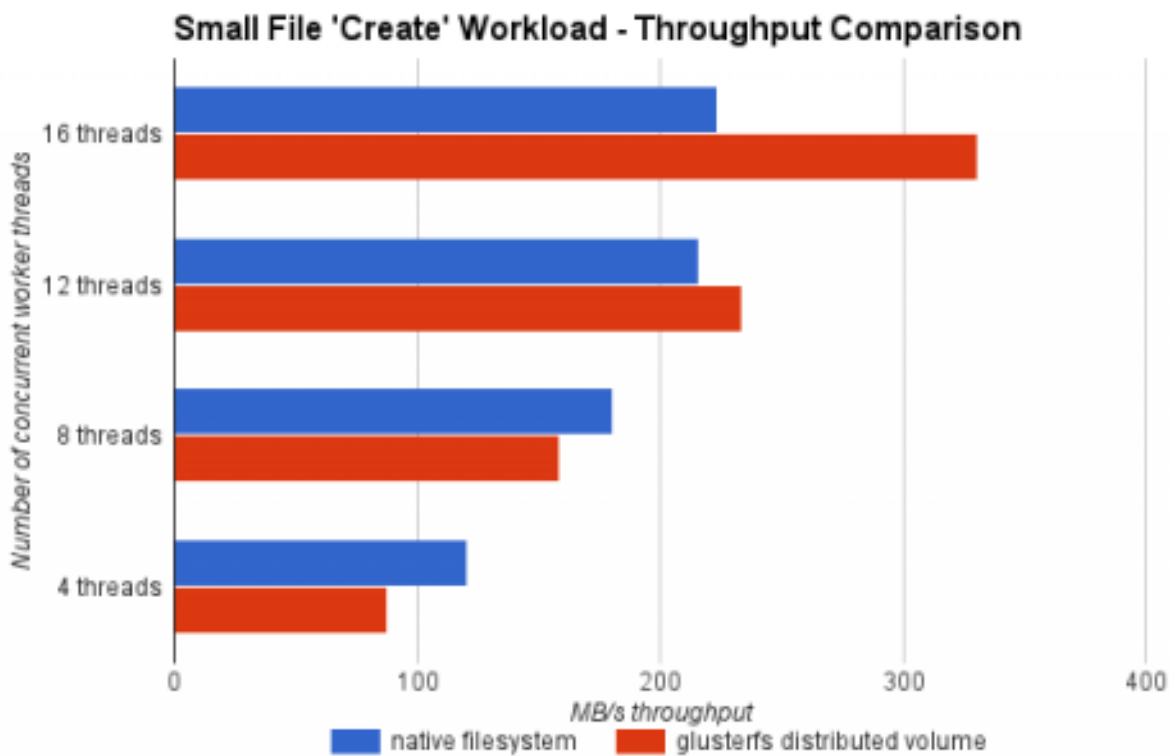
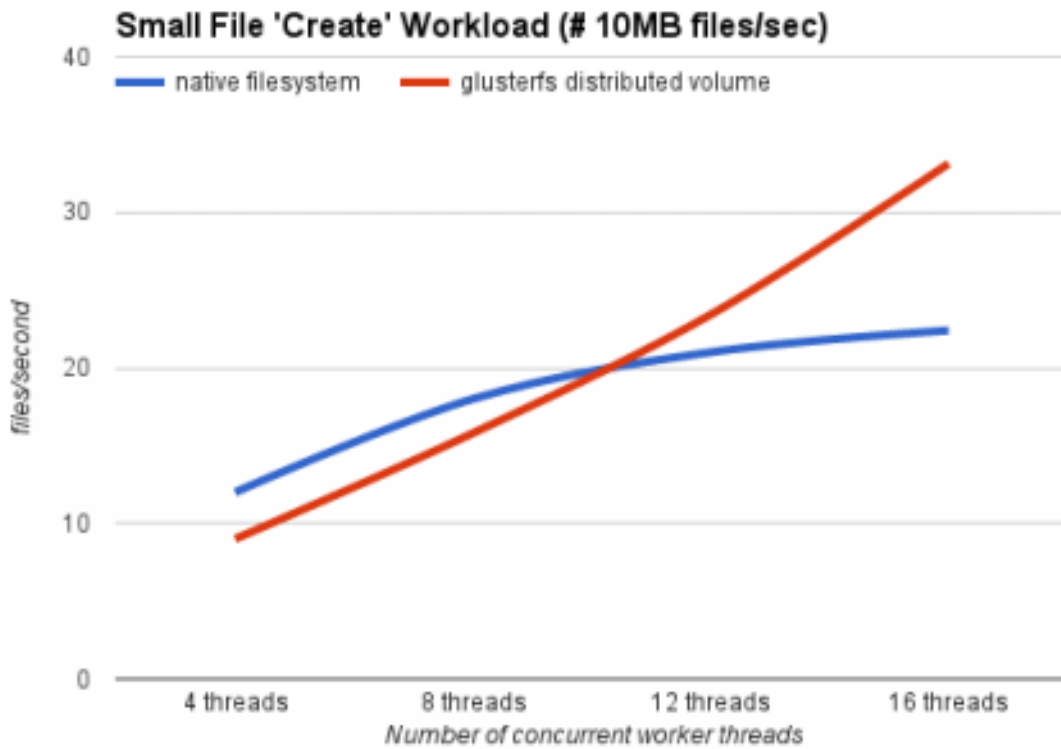
- 一般的なレイテンシーは 20 - 50 ミリ秒の範囲内です。
- より高い IOPS を取得するには、マルチスレッドのワークロードが必要です。つまり、1つの thread=32 IOPS, 32 スレッド = 961 IOPS です。
- 仮想ドライブを **mdadm** と組み合わせると、LUN は1つの仮想ディスクを超えて IOPS を提供できます。

3.8.4. GlusterFS

Gluster Storage のパフォーマンステストは、同様な環境からのパフォーマンスが示唆され、期待されるパフォーマンスを示しています。ベンチマークの目的で、小規模ファイルツールを使用して、ユーザー環境の一般的な複数の同時ファイル作成をシミュレートしています。

作成ワークロードでは、入れ子状のディレクトリー階層内に一連の 10 MB のファイルを作成し、メタデータ操作やファイル作成およびスループットなどです。

図3.4 Gluster パフォーマンス： Small ファイル「Create」ワークロード



注意点：

- ネイティブファイルシステムは適切に起動しますが、ネイティブファイルシステムのフリーズと GlusterFS ボリュームがスケーリングを継続しますが、8-12 スレッド間でパフォーマンスのクロスオーバーが発生します。
- GlusterFS ボリュームのスループットは、クライアントワークロードの増加によって直線的に増加します。

- 同時実行性が高いと、GlusterFS ボリュームはローカルファイルシステムを最大 47% に上げます。
- 同時実行性が高い場合には、ネイティブファイルシステムの負荷が低下します。テストの実行時にディスクサブシステムの統計を調査すると、問題は I/O の待機時間が増加していました (70 - 90%)。

第4章 GOOGLE CLOUD PLATFORM での RED HAT GLUSTER STORAGE の使用

Red Hat Gluster Storage は、Google Cloud Platform (GCP) 上のクラウド/スケールアプリケーションのデータのニーズに対応します。Red Hat Gluster Storage は、GCP で実行されるソフトウェア定義のファイルストレージソリューションを提供します。これにより、お客様のアプリケーションは、スケールアウトした柔軟性とパフォーマンスを備えた従来のファイルインターフェースを使用することができます。

Red Hat Gluster Storage 設計の中核となるのは、ストレージを設計するための全く新しい方法です。その結果、システムは、高いスケーラビリティ、高い回復力、さらなるパフォーマンスを発揮することができます。

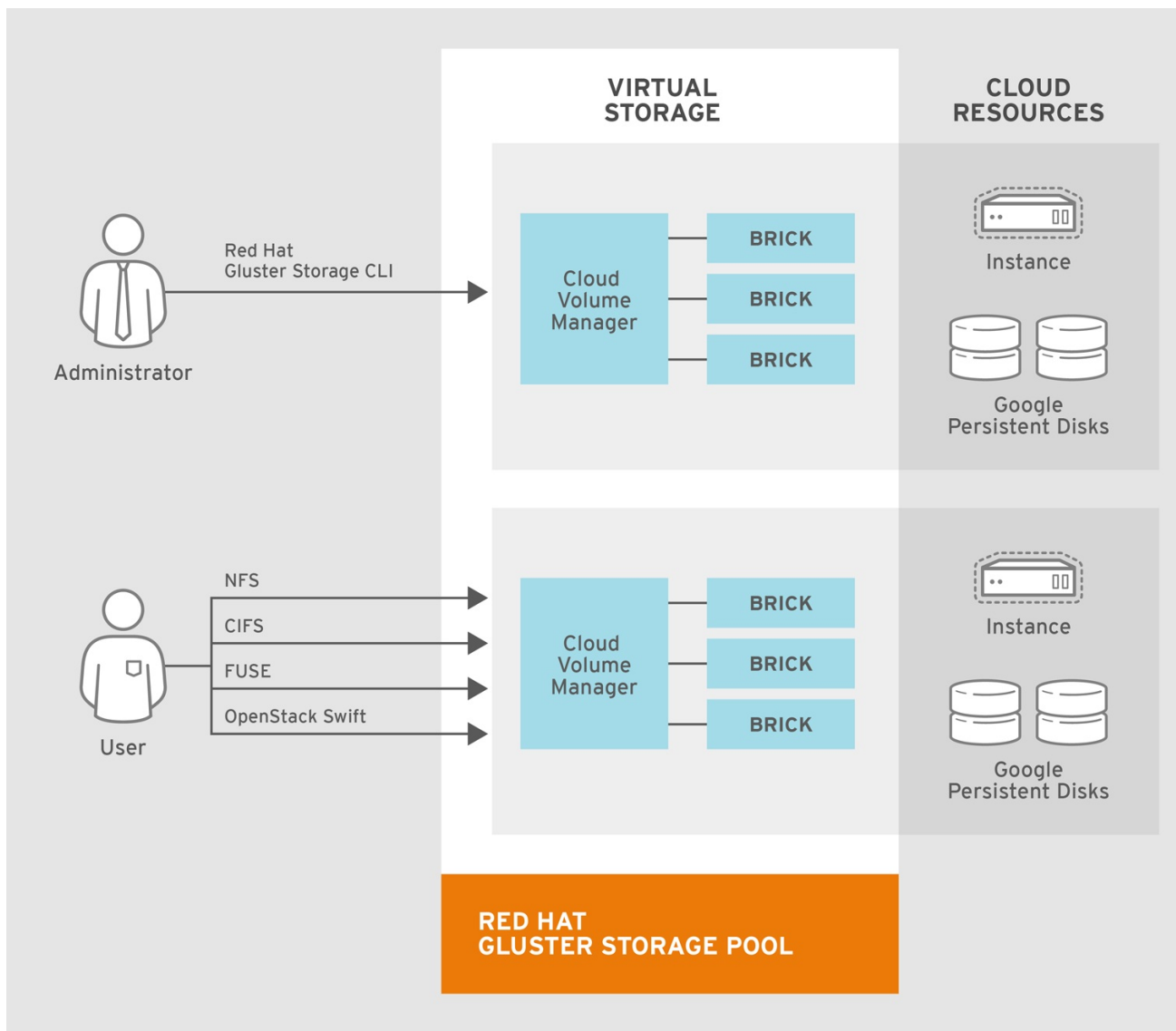
Google Cloud Platformの概要

Google Cloud Platform は Google のパブリッククラウドオフリングで、完全に統合されたクラウド環境を実行するためのサービスを多数提供します。Google Compute Engine は、仮想マシン環境をドライブとして、管理します。この章は、この仮想マシンのインフラストラクチャーに基づいています。この仮想フレームワークは、指定されたワークロードの需要を満たすために、Red Hat Gluster Storage 環境をスケールアウトするネットワーク、ストレージ、および仮想マシンを提供します。

Google Compute Engine の詳細は <https://cloud.google.com>、および <https://cloud.google.com/compute/docs> を参照してください。

以下の図は、Google Cloud Platform の Red Hat Gluster Storage との統合を示しています。

図4.1 統合アーキテクチャー



GLUSTER_389871_0216

Red Hat Gluster Storage のアーキテクチャー、概念、実装に関する詳細は、『『Red Hat Gluster Storage Administration Guide』』を参照してください。

本章では、10 の x 2 Distribute-Replicate ボリュームを使用して Red Hat Gluster Storage 環境を Google Cloud Platform にデプロイするために必要な手順を説明します。

4.1. デプロイメントのプランニング

本章では、100 TB の分散および複製されたファイルシステム領域をモデル化します。Red Hat Gluster Storage クライアントであるアプリケーションサーバーモデルには、ストリーミングビデオキャプチャーおよび取得シミュレーションを実行する 10 台の仮想マシンインスタンスが含まれます。このシミュレーションでは、分散ストレージシステムが最も適している可能性のある、他の一般的なユースケースでは、I/O パターンのワークロードを表す混合のワークロードを表すことができます。

このスケーリングにより、ストレージ容量およびクライアントアクティビティーのハイエンドシミュレーションをモデル化できますが、最高のパフォーマンスは大幅に小さくなる場合があります。モデルが個々の要件にスケールダウンされ、ユースケースを考慮する必要があります。たとえば、ゾーン間の同期レプリケーション、障害ドメインへの同期レプリケーションの分離、およびリモートの地理的サイトへの非同期レプリケーションの分離など、このアーキテクチャーの基本的なアプローチを考慮する必要があります。

永続ディスクの最大サイズ

元のテストビルドは、VM ごとの永続ディスクサイズを最大 10 TB で制限していました。Google では、その制限を 64 TB に拡大しました。Red Hat は、Google の現在の最大サイズ (64 TB) までの仮想マシンごとに永続ディスクをサポートします。(64 TB はディスクごとのものと仮想マシンごとの最大値の両方)、実際のデータディスクの最大値は 64 TB、オペレーティングシステムのディスクサイズを減らすことに注意してください。

その他実際のユースケースでは、本章で示されているものよりも、クライアント接続が大幅に関係します。サーバーおよびストレージスケールにフォーカスするため、ここで実行された特定の部分がクライアントスケールに制限されていましたが、基本的なスループットテストにはストレージシステムの線形のスケール機能が表示されていました。常に、特定のユースケースに合わせて、独自の設計を調整し、パフォーマンスとスケールリングの制限についてテストする必要があります。

4.1.1. 環境

scale ターゲットは使用可能なストレージの約 100 TB です。これは、プライマリープールのゾーン間で 2 方向同期レプリケーションを使用し、さらに障害復旧用の別のリージョンのセカンダリープールに非同期的ジオレプリケーションを実行します。この記事の執筆時点では、Google Compute Engine の永続ディスクの現在の最大サイズは 10 TB であるため、この設計にはプライマリープールに 20 ブリックとセカンダリープールに 10 ブリックが必要です。セカンダリープールには、同期的に複製されない単一のデータコピーがあります。

現在、VM ごとの永続ストレージは 10 TB の永続ディスクであるため、実際のデータディスクは 20 GB の root ボリュームの永続ディスクに対応できるように 10,220 GB で設定されます。

すべてのノードは、Red Hat Enterprise Linux 7.7 イメージ上で Red Hat Gluster Storage 3.5 を使用します。これは、KVM であるローカルの仮想化システムで作成され、設定されます。ローカルリージョン内の Red Hat Gluster Storage レプリカピアは、各リージョンの別々のゾーンに置かれます。これにより、ゾーンが停止した場合に、同期レプリカのコピーを高可用性にすることが可能です。

Red Hat Gluster Storage サーバーノードは、**n1-highmem-4** マシンタイプとして構築されます。このマシンタイプは、Red Hat Gluster Storage の公開済みリソース要件に基づく最小限の設定です。予想されるクラウドユースケースに基づいて、最小のメモリーサイズに対して、いくつかの収益が得られました。**n1-highmem-8** マシンタイプは、アプリケーションや特定のニーズに応じて、より適切な一致になります。

4.1.2. 前提条件

- Google アカウント
- Google Cloud SDKGoogle Cloud SDK には、Google Cloud Platform でリソースを簡単に作成し、管理できるツールおよびライブラリーが含まれています。これは後で使用され、複数の Red Hat Gluster Storage インスタンスの作成が容易になります。Google Cloud SDK を設定し、インストールする方法は、<https://cloud.google.com/sdk> を参照してください。
- Red Hat Gluster Storage ソフトウェアチャンネルにアクセスするためのサブスクリプション Red Hat Gluster Storage 3.5 チャンネルをサブスクライブする方法については、『[Red Hat Gluster Storage 3.5 Installation Guide](#)』の『Installing Red Hat Gluster Storage』を参照してください。
- ノードの最小数は 3 です。
- 互換性のある物理サーバー、仮想サーバー、クライアントの OS プラットフォームについては、<https://access.redhat.com/articles/66206> を参照してください。

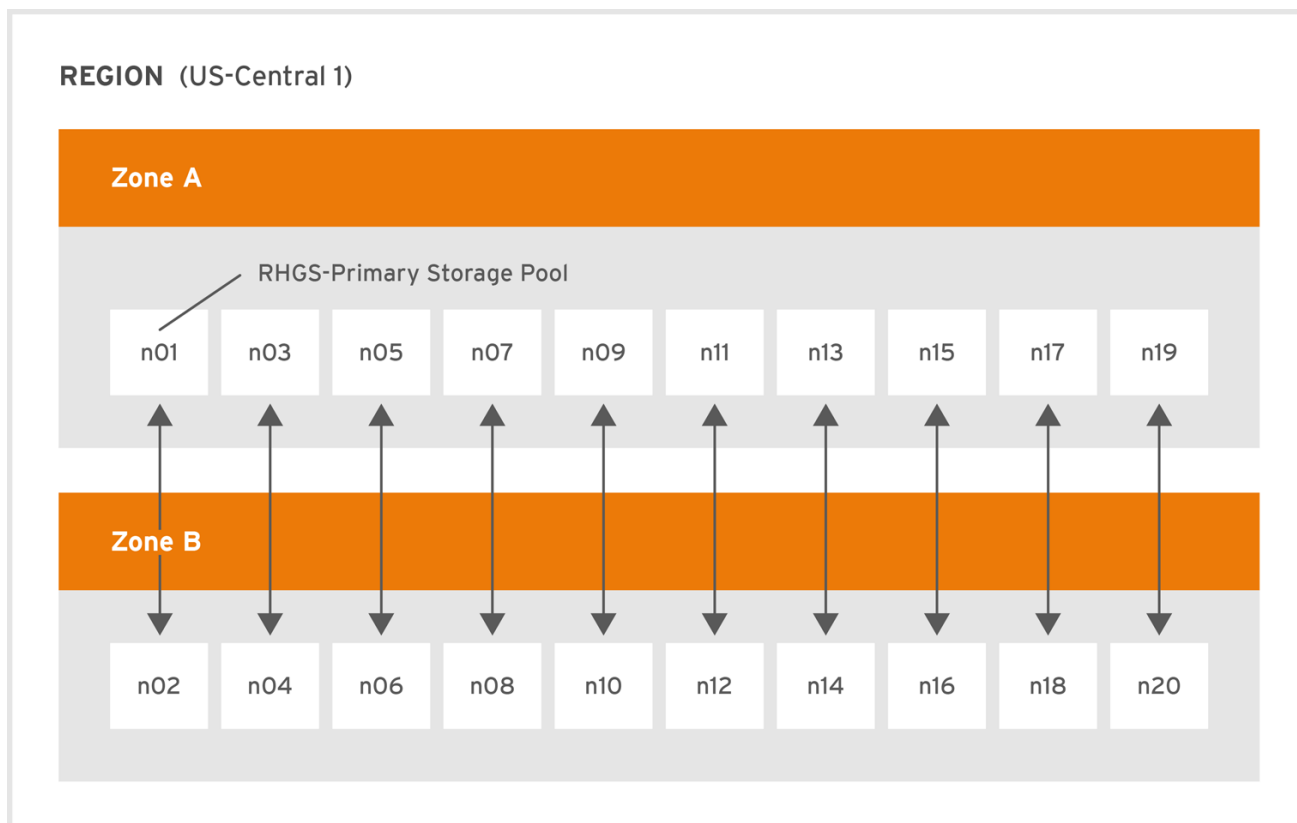
4.1.3. プライマリストレージプールの設定

- 10 x 2 Distribute-Replicate ボリュームに設定された Red Hat Gluster Storage
- 20 x n1-highmem-4 インスタンス:

リソース	仕様
vCPU	4
メモリー	26 GB
ブートディスク	20 GB 標準永続ディスク
データディスク	10,220 GB の標準永続ディスク。1つのインスタンスに対する永続ディスクの割り当ての最大数は 10 TB です。したがって、データディスクの最大サイズは 10 TB、ブートディスクの 20 GB のサイズ (10,220 GB) を引いた合計です。
イメージ	Red Hat Enterprise Linux 7.7 でのカスタム Red Hat Gluster Storage 3.5

- 仮想マシンゾーンの割り当て:

各 Gluster 同期レプリカのペアがゾーンに置かれ、ゾーンの失敗の影響を制限します。単一のゾーン障害により、データアクセスが失われることはありません。同期レプリカのペアの設定は、**gluster volume create** コマンドで定義された順序の機能です。



GLUSTER_389871_0216

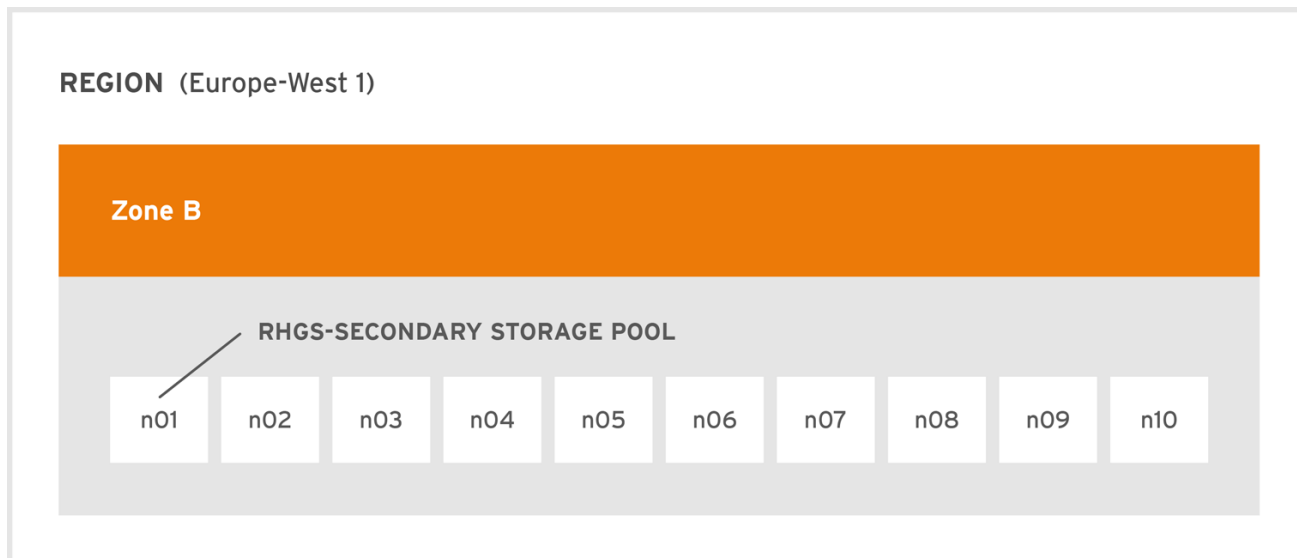
4.1.4. セカンダリーストレージプールの設定

- 10 x 1 分散ボリュームで設定された Gluster
- 10 x n1-highmem-4 インスタンス:

リソース	仕様
vCPU	4
メモリー	24 GB
ブートディスク	20 GB 標準永続ディスク
データディスク	10,220 GB 標準永続ディスク
イメージ	Red Hat Enterprise Linux 7.7 でのカスタム Red Hat Gluster Storage 3.5

- 仮想マシンゾーンの割り当て:

障害復旧のリモートリージョンの Geo レプリケーションを介して、非同期レプリケーションの受信側として設計されたセカンダリーストレージプール。この保護層のコストを制限するため、このストレージプールはローカルリージョン内で同期的に複製されず、分散のみの gluster ボリュームが使用されます。停止の影響を制限するため、このリージョンのすべてのノードが同じゾーンに配置されます。



GLUSTER_389871_0216

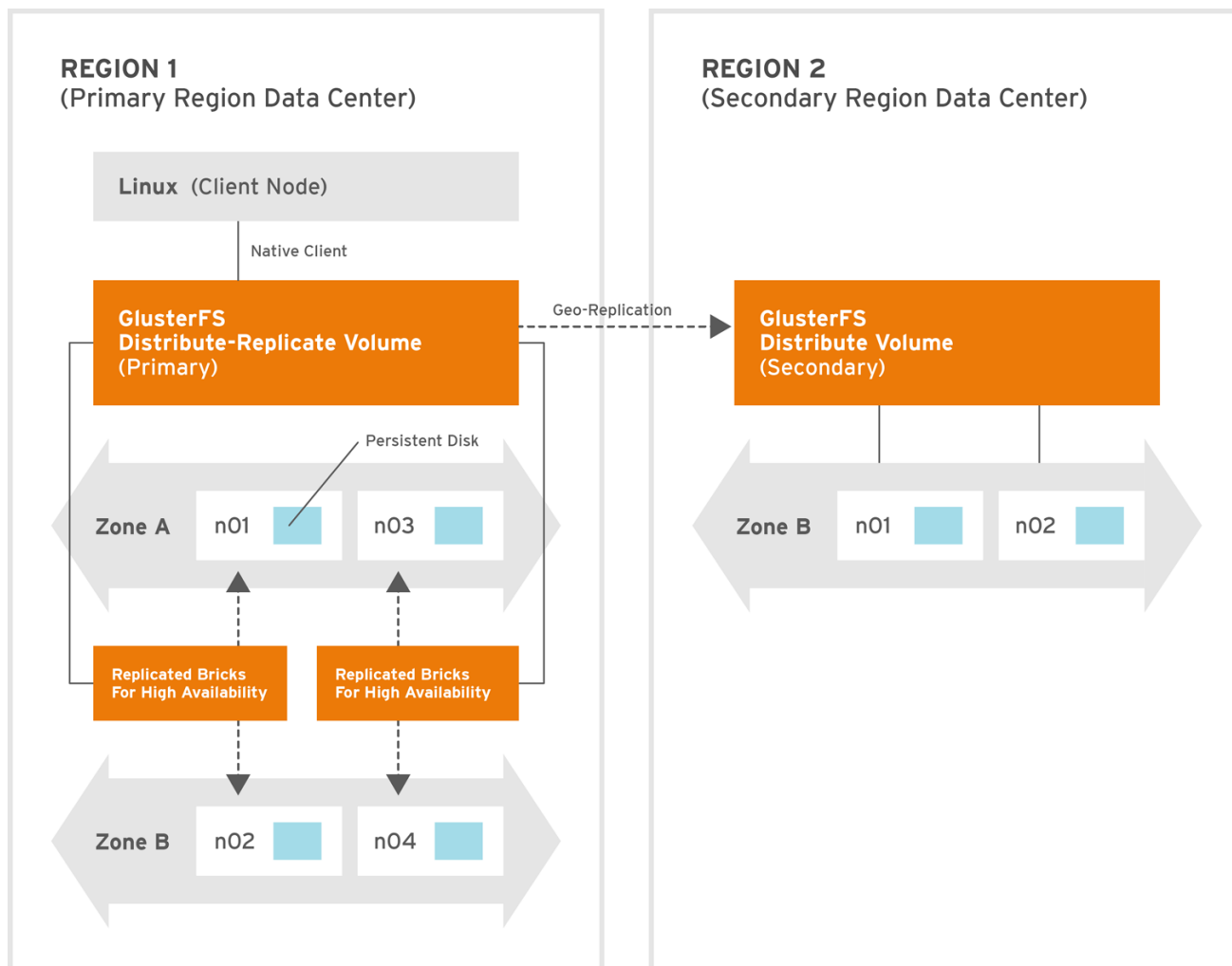
4.1.5. クライアント設定

クライアント仮想マシンは、US-CENTRAL1 リージョン、ゾーン A および B 全体でできるだけ均等に配布されています。

- 10 x n1-standard-2 インスタンス:

リソース	仕様
vCPU	2
メモリー	7.5 GB
ブートディスク	10 GB の標準永続ディスク
イメージ	Red Hat Enterprise Linux 7.7 でのカスタム Red Hat Gluster Storage 3.5

4.1.6. 信頼されたプールのポロジー



GLUSTER_389871_0216

4.1.7. Google Cloud Platform 向けの Red Hat Gluster Storage の取得

Red Hat サブスクリプションまたは Red Hat 評価サブスクリプションを使用して Red Hat Gluster Storage Server ファイルをダウンロードするには、以下を実行します。

1. Red Hat カスタマーポータル (<https://access.redhat.com/login>) にアクセスし、ユーザー名およびパスワードを入力してログインします。
2. **Downloads** をクリックし、**Software & Download Center** に移動します。
3. Red Hat Gluster Storage Server エリアで、**Download Software** をクリックして **qcow2** イメージの最新バージョンをダウンロードします。

4.2. GOOGLE COMPUTE ENGINE の設定

Google Compute Engine を設定するには、以下の手順を実行します。

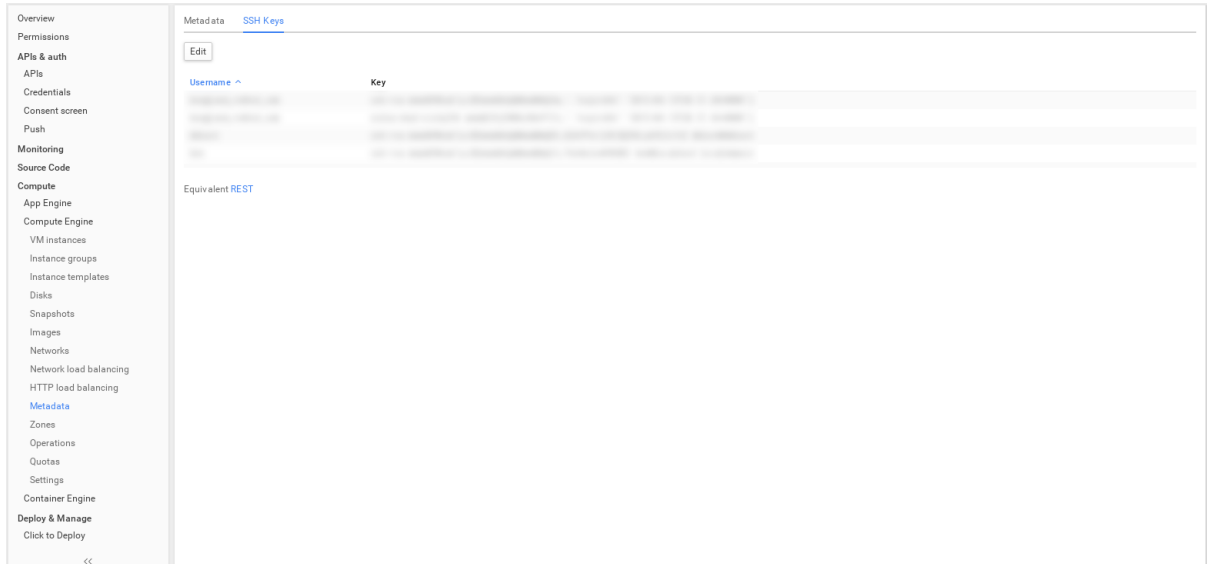
4.2.1. SSH キー

標準的な SSH 経由で接続するには、SSH 鍵を生成して、Google Compute Engine プロジェクトに登録する必要があります。インスタンスのパブリック IP アドレスには、生成後に直接 SSH 接続することができます。

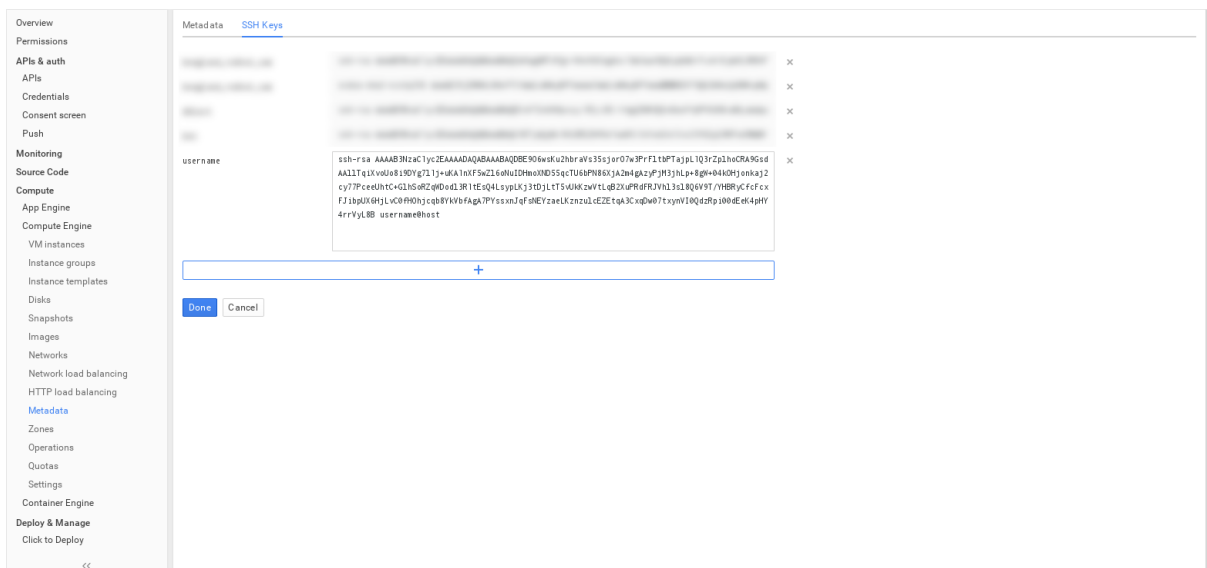
1. 以下のコマンドを使用して、Google Compute Engine で使用する SSH キーペアを生成します。

```
# ssh-keygen -t rsa -f ~/.ssh/google_compute_engine
```

2. Google Developers コンソールで、**Computer > Compute Engine > Metadata > SSH Keys > Edit** の順にクリックします。



3. `~/.ssh/google_compute_engine.pub` ファイルから生成された出力を入力し、**保存** をクリックします。



4. SSH エージェントが新規ローカルコンソールセッションごとにこの ID ファイルを使用するには、コンソールで以下のコマンドを実行します。

```
# ssh-add ~/.ssh/google_compute_engine
```

5. 以下の行を `~/.ssh/config` ファイルに追加すると、このコマンドの自動化に役立ちます。

```
IdentityFile ~/.ssh/google_compute_engine
```

6. これで、Google Compute Engine プロジェクトで作成した新しい仮想マシンインスタンスへの標準の SSH 経由で接続できるようになりました。

```
# ssh -i ~/.ssh/google_compute_engine <username>@<instance_external_ip>
```

Google Cloud SDK の `gcloud compute config-ssh` コマンドは、`~/.ssh/config` ファイルに、インスタンス名による単純な SSH 接続を可能にするエイリアスを設定します。

4.2.2. クォータの設定

以下の最小永続ディスククォータは、このデプロイメントに必要です。Google からクォータの増加を要求する必要がある場合があります。

- ローカルリージョン (「[プライマリストレージプールの設定](#)」の US-CENTRAL1 図を参照)
 - 永続ディスクの合計 (GB) \geq 206,000
 - 100 を超える CPU
- リモートリージョン (「[セカンダリストレージプールの設定](#)」の EUROPE-WEST1 を参照してください)
 - 103,000 を超える永続ディスクの合計(GB)
 - CPU \geq 40

4.3. QCOW2 から .RAW 形式への変換

以下のコマンドを使用して、ダウンロードした `qcow2` イメージを `.raw` 形式に変換します。

```
# qemu-img convert image_name disk.raw
```

以下は例になります。

```
# qemu-img convert RHGS-3.5.0.x86_64.qcow2 disk.raw
```

4.4. GOOGLE COMPUTE ENGINE のイメージのパッケージ化

以下のコマンドを使用して、`gzip sparse tar` アーカイブを作成して Google Compute Engine のイメージをパッケージ化します。

```
# tar -czSf disk.raw.tar.gz disk.raw
```

4.5. GOOGLE CLOUD STORAGE へのイメージのアップロード

Google クラウドにイメージをアップロードする前に、`gcloud auth login` コマンドを使用してログインする必要があります。コマンドを実行するとブラウザが開き、Google アカウントの認証情報の入力が必要とされます。PROJECT_ID はデフォルトで設定され、後続の CLI 命令に従い、必要に応じて変更を行います。

Google の `gsutil` コマンドを使用してストレージバケットを作成し、イメージをアップロードします。

```
# gsutil mb gs://rhgs_image_upload
# gsutil cp disk.raw.tar.gz gs://rhgs_image_upload
```

4.6. GOOGLE COMPUTE ENGINE へのイメージのインポート

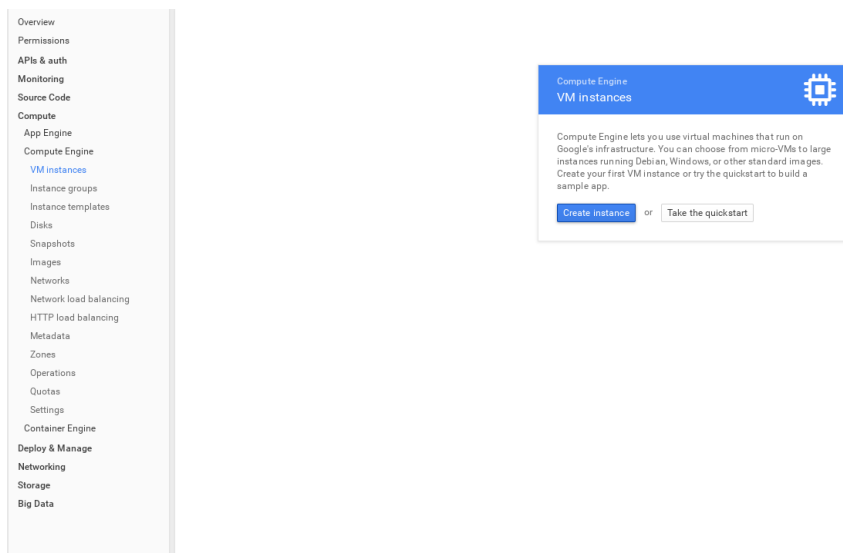
以下のコマンドを使用して、イメージを Google Compute Engine にインポートします。

```
# gcloud compute images create rhgs31 --source-uri gs://rhgs_image_upload/disk.raw.tar.gz
```

保存されたイメージの使用方法については、https://cloud.google.com/compute/docs/images/create-delete-deprecate-private-images#use_saved_image を参照してください。

4.7. RED HAT GLUSTER STORAGE インスタンスのディスクを設定するための仮想マシンインスタンスの作成

1. Google Developers コンソールで、**Compute > Compute Engine > VM instances > Create Instance** をクリックします。



Create Instance ウィンドウが表示されます。

Home
Permissions
APIs & auth
Monitoring
Source Code
Cloud Launcher
Deployments
Compute
App Engine
Compute Engine
VM instances
Instance groups
Instance templates
Disks
Snapshots
Images
Networks
Network load balancing
HTTP load balancing
Metadata
Health checks
Zones
Operations
Quotas
Settings
Container Engine
Networking
Storage
Big Data


←

Create a new instance


Name [?]
rhg-primary-n1

Zone [?]
us-central1-a

Machine type [?]

	n1-highmem-4	Memory	Change
	vCPUs	26 GB	
	4		

Boot disk [?]

	New 20 GB standard persistent disk	Change
	Image	
	rhgs31	

Firewall [?]
Add tags and firewall rules to allow specific network traffic from the Internet

Allow HTTP traffic
 Allow HTTPS traffic

Project access

Allow API access to all Google Cloud services in the same project. [Learn more](#)

∨ [Management, disk, networking, access & security options](#)

You will be billed for this instance. [Learn more](#)

Create Cancel

Equivalent [REST](#) or [command line](#)

2. **Create a new instance** ウィンドウで以下を入力し、**Create** をクリックします。

- 名前: rhgs-primary-n01
- ゾーン: us-central1-a
- マシンタイプ: n1-highmem-4 (4 vCPU、26 GB メモリー)
- ブートディスク: 20 GB 標準永続ディスク
- イメージ: rhgs31 (4 つのアップロードイメージファイル)

4.8. 初期データディスクの作成

インスタンスと同じゾーンに rhgs-primary-n01 仮想マシンインスタンス用に 10,220 GB の標準永続ディスクを作成します。

1. Google Developers コンソールで、**Compute > Compute Engine > Disks > New disk** をクリックします。

Overview
Permissions
APIs & auth
APIs
Credentials
Consent screen
Push
Monitoring
Source Code
Compute
App Engine
Compute Engine
VM instances
Instance groups
Instance templates
Disks
Snapshots
Images
Networks
Network load balancing
HTTP load balancing
Metadata
Zones
Operations
Quotas
Settings
Container Engine
Deploy & Manage
Click to Deploy

←

Create a new disk

Name

Description (Optional)

Zone

Disk Type

Source type

Size (GB)

Estimated performance

Operation Type	Read	Write
Sustained random IOPS limit	3,000	15,000
Sustained throughput limit (MB/s)	180	120

[Equivalent REST or command line](#)

2. **New Disk** ウィンドウで以下を入力し、**Create** をクリックします。

- Name: rhgs-primary-n01-data
- ゾーン: us-central1-a
- ディスクタイプ: 標準の永続ディスク
- ソースタイプ: None (空白ディスク)
- サイズ (GB): 10220

4.9. データディスクのアタッチおよび設定

1. Google Developers コンソールから、**Compute > Compute Engine > VM instances > rhgs-primary-n01 > Attach > rhgs-primary-n01-data** をクリックします。

Overview
Permissions
APIs & auth
APIs
Credentials
Consent screen
Push
Monitoring
Source Code
Compute
App Engine
Compute Engine
VM instances
Instance groups
Instance templates
Disks
Snapshots
Images
Networks
Network load balancing
HTTP load balancing
Metadata
Zones
Operations
Quotas
Settings
Container Engine
Deploy & Manage
Click to Deploy

External IP [Edit](#)

Internal IP

IP forwarding

Disks [Attach](#)

Delete boot disk when instance is deleted

Name	Size (GB)	Type	Mode
rhgs-primary-n01	10	Standard persistent disk	Boot, read/write

Network

Allow HTTP traffic Allow HTTPS traffic

Availability policies

Automatic restart

On host maintenance

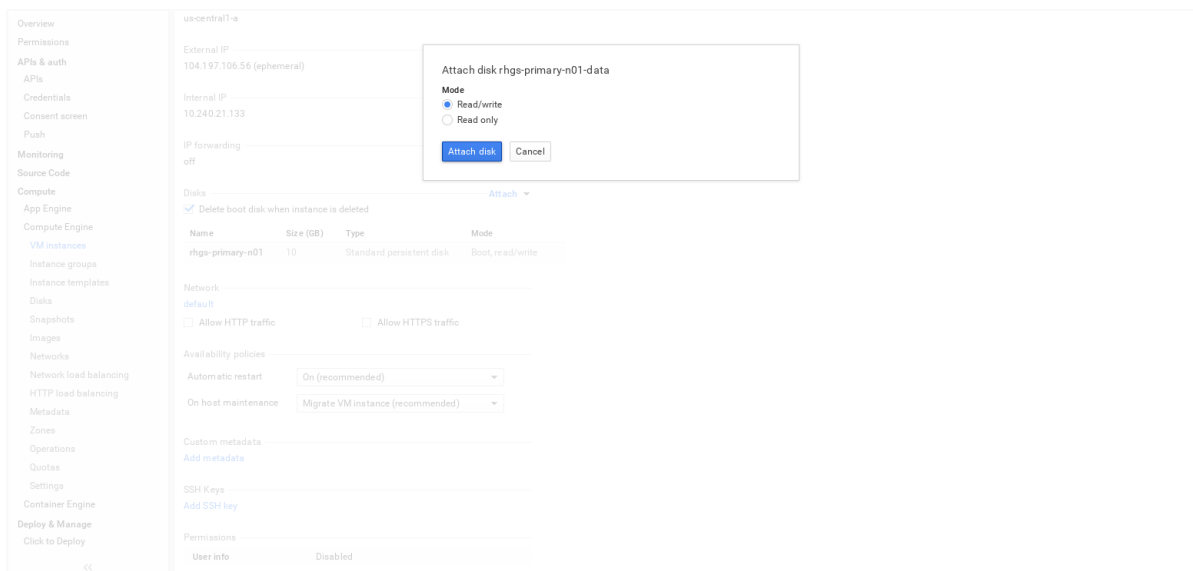
Custom metadata

SSH Keys

Permissions

User info	Disabled
Compute	Disabled

2. モードを **Read/write** として選択します。



- SSH 経由で rhgs-primary-n01 インスタンスに接続し、データディスクを設定します。

```
# ssh username@instance_external_ip
```

- データディスクが /dev/sdb として表示されることを確認します。

```
# fdisk -l /dev/sdb
```

```
Disk /dev/sdb: 10984.4 GB, 10984378859520 bytes
255 heads, 63 sectors/track, 1335441 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
Disk identifier: 0x00000000
```

- LVM を構成し、ファイルシステムのフォーマットを作成して、データディスクをマウントします。

以下のスクリプトを使用して、文書化された推奨事項に従ってこのプロセスを完了できます。



警告

このスクリプトは、スナップショットに必要な LVM シンプロビジョニングに、対応している最大メタデータ LV サイズ 16 GB に対応するのに十分なブロックデバイスをカバーすることを前提としています。スクリプトの各ステップを使用する前に、これを実行する内容を理解する必要があります。

```
#!/bin/bash
pvcreate /dev/sdb
vgcreate rhgs_vg /dev/sdb
# Create metadata LV with the maximum supported size of 16GB
```

```
lvcreate -L 16777216K --name rhgs_pool_meta rhgs_vg
# Create data LV with the remainder of the VG space
lvcreate -l 100%FREE --name rhgs_pool rhgs_vg
# The lvconvert command below required 4096 free extents, so reduce the LV
lvreduce -f -l 4096 /dev/rhgs_vg/rhgs_pool
# Convert our LVs to a thin pool
lvconvert --yes --thinpool rhgs_vg/rhgs_pool --poolmetadata rhgs_vg/rhgs_pool_meta
# Disable zeroing of thin pool chunks for performance boost
lvchange --zero n rhgs_vg/rhgs_pool
# The -V flag for lvcreate does not allow a '100%FREE' option like -l does.
# So we'll get the size of rhgs_pool from lvs for maximum efficiency
LVSIZE=$(lvs --units g | grep rhgs_pool | awk '{print $4}' | awk -F. '{print $1}')
# Create the thin LV for the bricks
lvcreate -V ${LVSIZE}G -T rhgs_vg/rhgs_pool -n rhgs_lv
# Create the XFS filesystem with 512B inode size and 8KB directory block size
# This step may take a while...
mkfs.xfs -f -i size=512 -n size=8192 -L rhgs_lv /dev/rhgs_vg/rhgs_lv
# Create mountpoint and fstab entry
mkdir -p /rhgs/bricks
echo "LABEL=rhgs_lv /rhgs/bricks xfs rw,inode64,noatime,nouuid 1 2" >> /etc/fstab
mount /rhgs/bricks
df -h /rhgs/bricks
```

4.10. イメージ作成プロセスのディスクの割り当て解除

ディスクが設定されたので、仮想マシンインスタンスから切り離し、次のステップで使用できるようにする必要があります。仮想マシンインスタンスはプロセス内で削除されます。

- Google Developers コンソールで、**Compute > Compute Engine > VM instances > rhgs-primary-n01** をクリックします。

←
SSH
⋮
Reset
Stop
Clone
Delete

✔ rhgs-primary-n01

CPU utilization ▾

1 hour
6 hours
12 hours
1 day
2 days
4 days
7 days
14 days
30 days

CPU

% CPU

Tags

glusterfs-deployed-from-google-developer-console

Edit

Machine type

n1-highmem-4 (4 vCPUs, 26 GB memory)

CPU platform

Intel Sandy Bridge

Zone

us-central1-a

External IP

130.211.127.252 (ephemeral)

Edit

Internal IP

10.240.214.249

IP forwarding

off

Boot disk and local disks

Name	Size (GB)	Type	Mode
rhgs-primary-n01	10	Standard persistent disk	Boot, read/write

Delete boot disk when instance is deleted

Save Cancel

Additional disks (Optional)

Name	Mode	When deleting instance
rhgs-primary-data-disk-n01	Read/write	Keep disk

+ Add item

Save Cancel

Network

default

Allow HTTP traffic

Allow HTTPS traffic

Disks セクションまで下方方向にスクロールします (ブートディスク用および追加ディスク用のものである必要があります)。インスタンスが削除されても、チェックボックスのチェックマークが外れ、インスタンスの削除時に追加のディスクには **Keep disk** が表示されることを確認します。

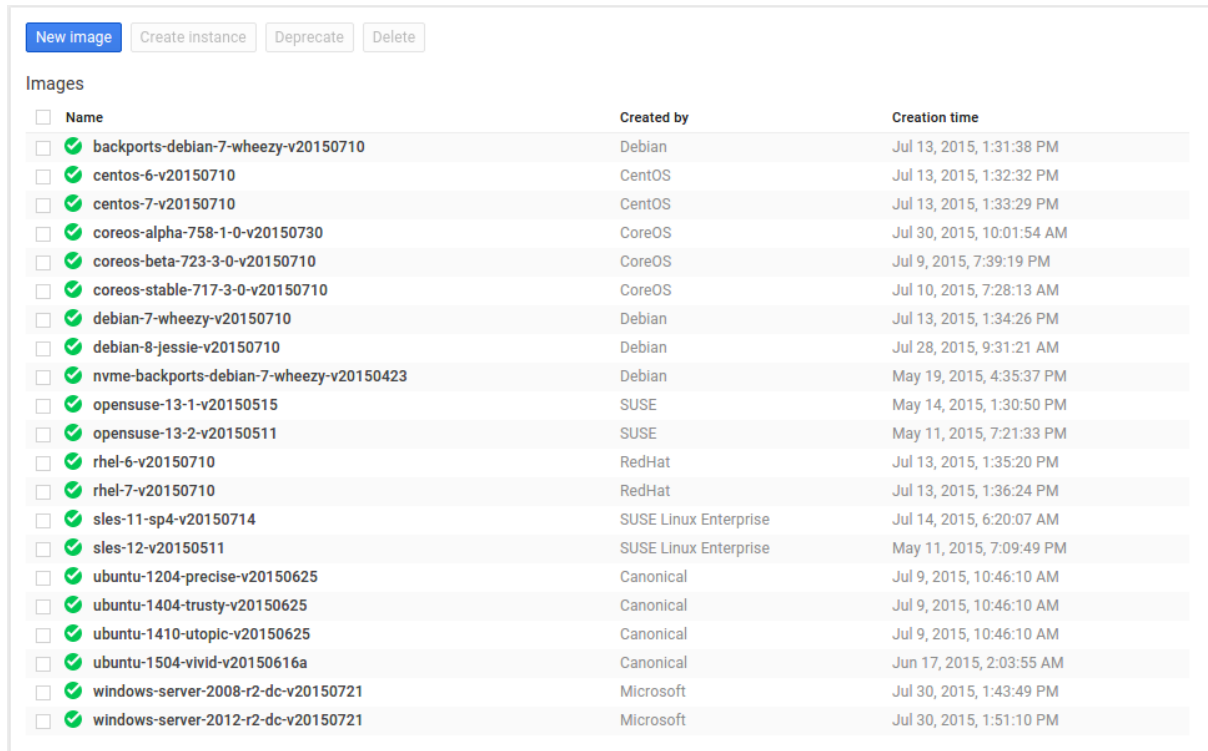
次に、上部の **Delete** をクリックし、仮想マシンインスタンスを削除します。

4.11. イメージを使用した複数の RED HAT GLUSTER STORAGE インスタンスの作成

本章では、ローカルリージョン、US-CENTRAL1、および 10 の仮想マシンインスタンス (EUROPE-WEST1) で 20 台の仮想マシンインスタンスを使用しています。

1. rhgs-primary-n01 のイメージをルート永続ディスクとして作成する。

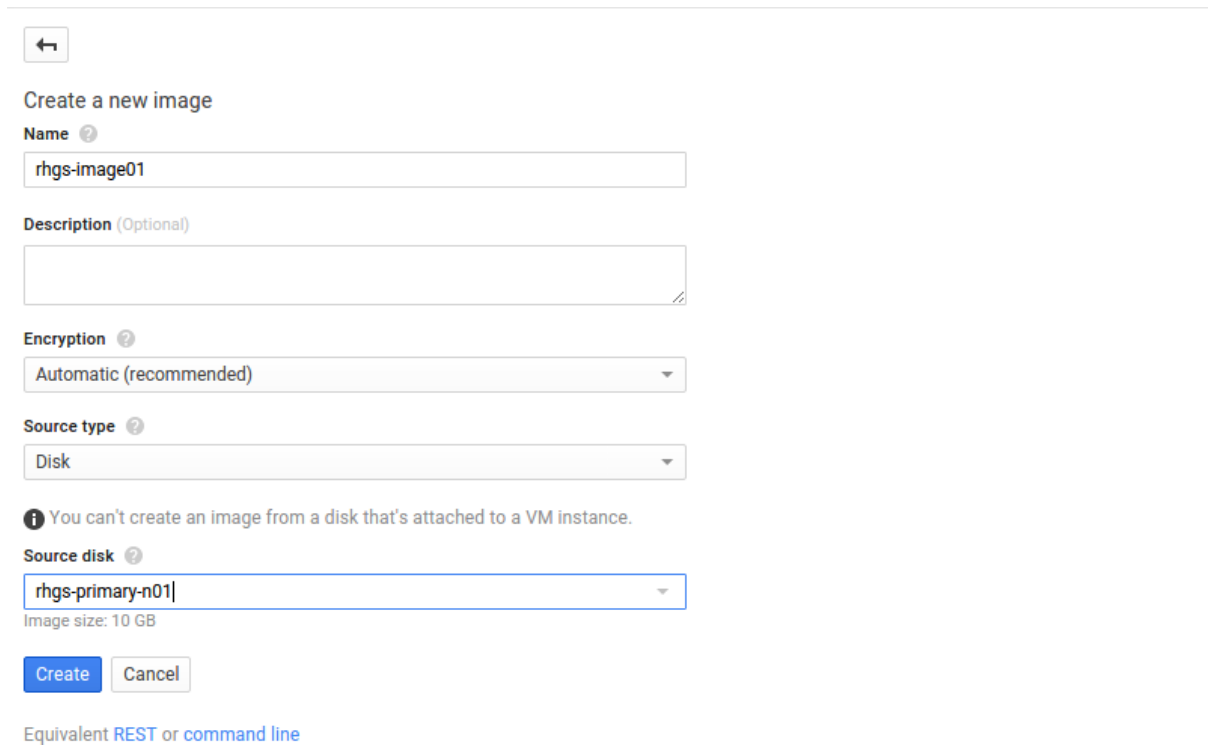
Google Developers コンソールで、**Compute > Compute Engine > Images > New image**をクリックします。



The screenshot shows the 'New image' page in the Google Cloud Platform console. At the top, there are buttons for 'New image', 'Create instance', 'Deprecate', and 'Delete'. Below this is a table of existing images with columns for 'Name', 'Created by', and 'Creation time'. Each row has a checkbox on the left and a green checkmark in the 'Name' column.

Name	Created by	Creation time
backports-debian-7-wheezy-v20150710	Debian	Jul 13, 2015, 1:31:38 PM
centos-6-v20150710	CentOS	Jul 13, 2015, 1:32:32 PM
centos-7-v20150710	CentOS	Jul 13, 2015, 1:33:29 PM
coreos-alpha-758-1-0-v20150730	CoreOS	Jul 30, 2015, 10:01:54 AM
coreos-beta-723-3-0-v20150710	CoreOS	Jul 9, 2015, 7:39:19 PM
coreos-stable-717-3-0-v20150710	CoreOS	Jul 10, 2015, 7:28:13 AM
debian-7-wheezy-v20150710	Debian	Jul 13, 2015, 1:34:26 PM
debian-8-jessie-v20150710	Debian	Jul 28, 2015, 9:31:21 AM
nvme-backports-debian-7-wheezy-v20150423	Debian	May 19, 2015, 4:35:37 PM
opensuse-13-1-v20150515	SUSE	May 14, 2015, 1:30:50 PM
opensuse-13-2-v20150511	SUSE	May 11, 2015, 7:21:33 PM
rhel-6-v20150710	RedHat	Jul 13, 2015, 1:35:20 PM
rhel-7-v20150710	RedHat	Jul 13, 2015, 1:36:24 PM
sles-11-sp4-v20150714	SUSE Linux Enterprise	Jul 14, 2015, 6:20:07 AM
sles-12-v20150511	SUSE Linux Enterprise	May 11, 2015, 7:09:49 PM
ubuntu-1204-precise-v20150625	Canonical	Jul 9, 2015, 10:46:10 AM
ubuntu-1404-trusty-v20150625	Canonical	Jul 9, 2015, 10:46:10 AM
ubuntu-1410-utopic-v20150625	Canonical	Jul 9, 2015, 10:46:10 AM
ubuntu-1504-vivid-v20150616a	Canonical	Jun 17, 2015, 2:03:55 AM
windows-server-2008-r2-dc-v20150721	Microsoft	Jul 30, 2015, 1:43:49 PM
windows-server-2012-r2-dc-v20150721	Microsoft	Jul 30, 2015, 1:51:10 PM

Create a new image ウィンドウが表示されます。



The screenshot shows the 'Create a new image' window. It has a back arrow at the top left. The title is 'Create a new image'. Below the title are several fields:

- Name**: rhgs-image01
- Description (Optional)**: (empty text area)
- Encryption**: Automatic (recommended)
- Source type**: Disk
- Source disk**: rhgs-primary-n01

Below the 'Source disk' field, it says 'Image size: 10 GB'. At the bottom, there are 'Create' and 'Cancel' buttons. A note below the buttons says 'Equivalent REST or command line'.

Create a new image ウィンドウで以下を入力し、**Create** をクリックします。

- 名前: rhgs-image01
- ソースディスク : rhgs-primary-n01

2. rhgs-primary-n01-data のイメージをデータ永続ディスクとして作成する。

Google Developers コンソールで、**Compute > Compute Engine > Images > New image**をクリックします。

Create a new image ウィンドウで以下を入力し、**Create** をクリックします。

- 名前: rhgs-data-image01
- ソースディスク: rhgs-primary-n01-data

4.12. GOOGLE CLOUD DEPLOYMENT MANAGER を使用した複数インスタンスのデプロイ

google クラウドデプロイメントマネージャーを使用すると、複数の RGHS インスタンスの作成が容易になります。

1. <https://cloud.google.com/sdk/gcloud/#gcloud.auth> の手順に従って、ログインして Google Cloud に対して認証します。
2. 「付録：Red Hat Gluster Storage デプロイメントの設定ファイル」 の設定内容をコピーします。
3. 以下の gcloud コマンドを実行します。

```
# gcloud deployment-manager deployments create rghs --config glusterfs-config.yaml
```

4.13. RED HAT GLUSTER STORAGE の設定

4.13.1. ノードのピア

rhgs-primary-n01 から、ピアノード rhgs-primary-n{02..20}:

```
# for i in {02..20};
do gluster peer probe rhgs-primary-n${i};
done
```

rhgs-primary-n02 から、ピアノード rhgs-primary-n01 から、以下を実行します。

問題を書き留めておきます。この手順は、最初のピアプロセスをクリーンアップし、他のピアが Gluster の信頼できるプール設定ファイルの IP アドレスで定義される rhgs-primary-n01 のままとまります。これは、IP アドレスは一時的なものであるため、重要になります。

```
# gluster peer status | grep Hostname | grep -v rhgs
Hostname: 10.240.21.133
```

そして修正します。

```
# gluster peer probe rhgs-primary-n01
peer probe: success.
# gluster peer status | grep Hostname | grep n01
Hostname: rhgs-primary-n01
```

rhgs--n01 以降、ピアノード rhgs-secondary-n{02..10}:

```
# for i in {02..10};
do gluster peer probe rhgs-secondary-n${i};
done
```

from rhgs-secondary-n02, peer node rhgs-secondary-n01:

```
# gluster peer probe rhgs-secondary-n01
```

4.13.2. 分散ボリュームの作成



警告

双方向レプリケーションのサポートは、Red Hat Gluster Storage の今後のバージョンで廃止および削除される予定です。これにより、レプリケートされたボリュームと分散複製ボリュームの両方に影響します。

双方向レプリケーションはスプリットブレイン条件から十分な保護を提供しないため、サポートは削除されます。ダミーノードは、この問題のソリューションとして使用できますが、Red Hat では、現在 2 方向のレプリケーションを使用するすべてのボリュームを、Arbitrated レプリケーションまたは 3 方向レプリケーションのいずれかを使用するように移行することが推奨されます。

双方向の複製されたボリュームを arbitrated レプリケートされたボリュームに移行する手順は、「[Arbitrated ボリュームへの変換](#)」を参照してください。

3 方向のレプリケーションに関する情報は、「[3 方向の複製ボリュームの作成](#)」および「[3 方向の分散複製ボリュームの作成](#)」を参照してください。

プライマリ信頼できるプールで 10x2 Distribute-Replicate ボリュームを作成し、ブリックが「[プライマリストレージプールの設定](#)」で定義されているようにレプリカピアと適切にペアになるようにします。

```
# gluster volume create myvol replica 2 \
rhgs-primary-n01:/rhgs/bricks/myvol rhgs-primary-n02:/rhgs/bricks/myvol \
rhgs-primary-n03:/rhgs/bricks/myvol rhgs-primary-n04:/rhgs/bricks/myvol \
rhgs-primary-n05:/rhgs/bricks/myvol rhgs-primary-n06:/rhgs/bricks/myvol \
rhgs-primary-n07:/rhgs/bricks/myvol rhgs-primary-n08:/rhgs/bricks/myvol \
rhgs-primary-n09:/rhgs/bricks/myvol rhgs-primary-n10:/rhgs/bricks/myvol \
rhgs-primary-n11:/rhgs/bricks/myvol rhgs-primary-n12:/rhgs/bricks/myvol \
rhgs-primary-n13:/rhgs/bricks/myvol rhgs-primary-n14:/rhgs/bricks/myvol \
rhgs-primary-n15:/rhgs/bricks/myvol rhgs-primary-n16:/rhgs/bricks/myvol \
rhgs-primary-n17:/rhgs/bricks/myvol rhgs-primary-n18:/rhgs/bricks/myvol \
rhgs-primary-n19:/rhgs/bricks/myvol rhgs-primary-n20:/rhgs/bricks/myvol
volume create: myvol: success: please start the volume to access data
```

```
# gluster volume start myvol
volume start: myvol: success
```



```
# gluster volume info myvol
Volume Name: myvol
Type: Distributed-Replicate
Volume ID: f093e120-b291-4362-a859-8d2d4dd87f3a
Status: Started
Snap Volume: no
Number of Bricks: 10 x 2 = 20
Transport-type: tcp
Bricks:
Brick1: rhgs-primary-n01:/rhgs/bricks/myvol
Brick2: rhgs-primary-n02:/rhgs/bricks/myvol
Brick3: rhgs-primary-n03:/rhgs/bricks/myvol
Brick4: rhgs-primary-n04:/rhgs/bricks/myvol
Brick5: rhgs-primary-n05:/rhgs/bricks/myvol
Brick6: rhgs-primary-n06:/rhgs/bricks/myvol
Brick7: rhgs-primary-n07:/rhgs/bricks/myvol
Brick8: rhgs-primary-n08:/rhgs/bricks/myvol
Brick9: rhgs-primary-n09:/rhgs/bricks/myvol
Brick10: rhgs-primary-n10:/rhgs/bricks/myvol
Brick11: rhgs-primary-n11:/rhgs/bricks/myvol
Brick12: rhgs-primary-n12:/rhgs/bricks/myvol
Brick13: rhgs-primary-n13:/rhgs/bricks/myvol
Brick14: rhgs-primary-n14:/rhgs/bricks/myvol
Brick15: rhgs-primary-n15:/rhgs/bricks/myvol
Brick16: rhgs-primary-n16:/rhgs/bricks/myvol
Brick17: rhgs-primary-n17:/rhgs/bricks/myvol
Brick18: rhgs-primary-n18:/rhgs/bricks/myvol
Brick19: rhgs-primary-n19:/rhgs/bricks/myvol
Brick20: rhgs-primary-n20:/rhgs/bricks/myvol
Options Reconfigured:
performance.readdir-ahead: on
auto-delete: disable
snap-max-soft-limit: 90
snap-max-hard-limit: 256
```

作成される Gluster ポリユーメントポロジューは次のとおりです。

```
Distribute set
|
+-- Replica set 0
| |
| +-- Brick 0: rhgs-primary-n01:/rhgs/bricks/myvol
| |
| +-- Brick 1: rhgs-primary-n02:/rhgs/bricks/myvol
|
+-- Replica set 1
| |
| +-- Brick 0: rhgs-primary-n03:/rhgs/bricks/myvol
| |
| +-- Brick 1: rhgs-primary-n04:/rhgs/bricks/myvol
|
+-- Replica set 2
| |
| +-- Brick 0: rhgs-primary-n05:/rhgs/bricks/myvol
| |
```

```

|   +-- Brick 1: rhgs-primary-n06:/rhgs/bricks/myvol
|
+-- Replica set 3
|   |
|   +-- Brick 0: rhgs-primary-n07:/rhgs/bricks/myvol
|   |
|   +-- Brick 1: rhgs-primary-n08:/rhgs/bricks/myvol
|
+-- Replica set 4
|   |
|   +-- Brick 0: rhgs-primary-n09:/rhgs/bricks/myvol
|   |
|   +-- Brick 1: rhgs-primary-n10:/rhgs/bricks/myvol
|
+-- Replica set 5
|   |
|   +-- Brick 0: rhgs-primary-n11:/rhgs/bricks/myvol
|   |
|   +-- Brick 1: rhgs-primary-n12:/rhgs/bricks/myvol
|
+-- Replica set 6
|   |
|   +-- Brick 0: rhgs-primary-n13:/rhgs/bricks/myvol
|   |
|   +-- Brick 1: rhgs-primary-n14:/rhgs/bricks/myvol
|
+-- Replica set 7
|   |
|   +-- Brick 0: rhgs-primary-n15:/rhgs/bricks/myvol
|   |
|   +-- Brick 1: rhgs-primary-n16:/rhgs/bricks/myvol
|
+-- Replica set 8
|   |
|   +-- Brick 0: rhgs-primary-n17:/rhgs/bricks/myvol
|   |
|   +-- Brick 1: rhgs-primary-n18:/rhgs/bricks/myvol
|
+-- Replica set 9
|   |
|   +-- Brick 0: rhgs-primary-n19:/rhgs/bricks/myvol
|   |
|   +-- Brick 1: rhgs-primary-n20:/rhgs/bricks/myvol

```

セカンダリーの信頼されるプールで、10 ブリックの分散ボリュームを作成します。

```

# gluster volume create myvol-slave \
rhgs-secondary-n01:/rhgs/bricks/myvol-slave \
rhgs-secondary-n02:/rhgs/bricks/myvol-slave \
rhgs-secondary-n03:/rhgs/bricks/myvol-slave \
rhgs-secondary-n04:/rhgs/bricks/myvol-slave \
rhgs-secondary-n05:/rhgs/bricks/myvol-slave \
rhgs-secondary-n06:/rhgs/bricks/myvol-slave \
rhgs-secondary-n07:/rhgs/bricks/myvol-slave \
rhgs-secondary-n08:/rhgs/bricks/myvol-slave \

```

```
rhgs-secondary-n09:/rhgs/bricks/myvol-slave \  
rhgs-secondary-n10:/rhgs/bricks/myvol-slave  
volume create: myvol-slave: success: please start the volume to access data
```

```
# gluster volume start myvol-slave  
volume start: myvol-slave: success
```

```
# gluster volume info myvol-slave  
Volume Name: myvol-slave  
Type: Distribute  
Volume ID: 64295b00-ac19-436c-9aac-6069e0a5b8cf  
Status: Started  
Snap Volume: no  
Number of Bricks: 10  
Transport-type: tcp  
Bricks:  
Brick1: rhgs-secondary-n01:/rhgs/bricks/myvol-slave  
Brick2: rhgs-secondary-n02:/rhgs/bricks/myvol-slave  
Brick3: rhgs-secondary-n03:/rhgs/bricks/myvol-slave  
Brick4: rhgs-secondary-n04:/rhgs/bricks/myvol-slave  
Brick5: rhgs-secondary-n05:/rhgs/bricks/myvol-slave  
Brick6: rhgs-secondary-n06:/rhgs/bricks/myvol-slave  
Brick7: rhgs-secondary-n07:/rhgs/bricks/myvol-slave  
Brick8: rhgs-secondary-n08:/rhgs/bricks/myvol-slave  
Brick9: rhgs-secondary-n09:/rhgs/bricks/myvol-slave  
Brick10: rhgs-secondary-n10:/rhgs/bricks/myvol-slave  
Options Reconfigured:  
performance.readdir-ahead: on  
snap-max-hard-limit: 256  
snap-max-soft-limit: 90  
auto-delete: disable
```

作成される Gluster ポリームトポロジーは次のとおりです。

```
Distribute set  
|  
+-- Brick 0: rhgs-secondary-n01:/rhgs/bricks/myvol-slave  
|  
+-- Brick 1: rhgs-secondary-n02:/rhgs/bricks/myvol-slave  
|  
+-- Brick 2: rhgs-secondary-n03:/rhgs/bricks/myvol-slave  
|  
+-- Brick 3: rhgs-secondary-n04:/rhgs/bricks/myvol-slave  
|  
+-- Brick 4: rhgs-secondary-n05:/rhgs/bricks/myvol-slave  
|  
+-- Brick 5: rhgs-secondary-n06:/rhgs/bricks/myvol-slave  
|  
+-- Brick 6: rhgs-secondary-n07:/rhgs/bricks/myvol-slave  
|  
+-- Brick 7: rhgs-secondary-n08:/rhgs/bricks/myvol-slave  
|  
+-- Brick 8: rhgs-secondary-n09:/rhgs/bricks/myvol-slave  
|  
+-- Brick 9: rhgs-secondary-n10:/rhgs/bricks/myvol-slave
```

4.13.3. プライマリーからセカンダリーリージョンへの Geo レプリケーションの設定

プライマリーリージョンノードから、ローカルの **myvol** ボリュームからリモートリージョン **myvol-slave** ボリュームに geo レプリケーションを確立します。

1. 前提条件として、すべてのセカンダリー/スレーブサイドノードで SSH 経由での root ユーザーのログインを許可する必要があります。以下のコマンドは、すべての `rhgs-secondary-n{01..10}` で実行する必要があります。

```
# sed -i s/PermitRootLogin\ no/PermitRootLogin\ yes/\
/etc/ssh/sshd_config
# service sshd restart
```

2. `rhgs-primary-n01` で root ユーザーの SSH キーペアを作成し、パブリックキーの内容をコピーします。

```
# ssh-keygen
# cat ~root/.ssh/id_rsa.pub
ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAQEAmtzZdIR+pEI16LqH0kbGQfA7sTe1iWHhV/x+5zVDb91
Z+gzMVdBTBaLyugeoBlxzOeFFnc/7a9TwNSr7YWt/yKZxh+lnqq
/9xcWtONUrfvLH4TEWu4dlRwCvXGsdv23IQK0YabaY9hqzshscFtSnQTmzT13LPc9drH+k7IH
Bu4KjA4igDvX/j41or0weneg1vcqAP9vRyh4xXgtocqBiAqJegBZ5O
/QO1ynyJBysp7tIHf7HZuh3sFCxtqEPPsJkVJDiQZ/NqTr3hAqDzmn4USOX3FbSOvomlWa8W
e6tGb9nfUH6vBQGyKbWk4YOzm6E5oTzuRBGA1vCPmwpwR/cw==
root@rhgs-primary-n01
```

3. `rhgs-secondary-n01` で、SSH 公開鍵を `rhgs-primary-n01` から root ユーザーの `authorized_keys` ファイルに追加します。

```
# echo "ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAQEAmtzZdIR+pEI16LqH0kbGQfA7sTe1iWHhV/x+5zVDb91
Z+gzMVdBTBaLyugeoBlxzOeFFnc/7a9TwNSr7YWt
/yKZxh+lnqq7/9xcWtONUrfvLH4TEWu4dlRwCvXGsdv23IQK0YabaY9hqzshscFtSnQTmzT13L
Pc9drH+k7IHBu4KjA4igDvX
j41or0weneg1vcqAP9vRyh4xXgtocqBiAqJegBZ5O/QO1ynyJBysp7tIHf7HZuh3sFCxtqEPPsJk
VJDiQZ
/NqTr3hAqDzmn4USOX3FbSOvomlWa8We6tGb9nfUH6vBQGyKbWk4YOzm6E5oTzuRBGA1
vCPmwpwR/cw== root@rhgs-primary-n01" | sudo tee ~root/.ssh
/authorized_keys > /dev/null
```

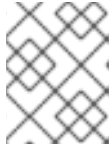


注記

上記の SSH 公開鍵は、説明のみを目的としています。 `rhgs-primary-n01` で独自の `id_rsa.pub` ファイルからのキーを使用します。

この時点で、`rhgs-primary-n01` の root ユーザーは、`rhgs-secondary-n01` へのパスワードなしの SSH アクセスを持つ必要があります。これは、geo レプリケーションの設定の前提条件です。

1. `rhgs-primary-n01` に共通の pem pub ファイルを作成します。



注記

これは、パスワードなしの SSH がセカンダリーノードに設定されたノードで行う必要があります。

```
# gluster system:: execute gsec_create
```

- プライマリーサイトからセカンダリーサイトへのジオレプリケーションセッションを作成します。セカンダリーノードに必要な pem-file を設定するには、push-pem オプションが必要です。

```
# gluster volume geo-replication myvol \
rhgs-secondary-n01::myvol-slave create push-pem
```

```
# gluster volume geo-replication myvol \
rhgs-secondary-n01::myvol-slave start
```

- geo レプリケーションのステータスを確認します。数分後に初期化ステージを完了し、各接続はそのステータスに Active または Passive と表示されるはずで

```
# gluster volume geo-replication myvol rhgs-secondary-n01::myvol-slave status
MASTER NODE      MASTER VOL  MASTER BRICK      SLAVE USER  SLAVE
STATUS  CHECKPOINT STATUS  CRAWL STATUS
-----
rhgs-primary-n01  myvol      /rhgs/bricks/myvol  root      rhgs-secondary-n10::myvol-
slave Active  N/A      Changelog Crawl
rhgs-primary-n18  myvol      /rhgs/bricks/myvol  root      rhgs-secondary-n05::myvol-
slave Passive N/A      N/A
rhgs-primary-n06  myvol      /rhgs/bricks/myvol  root      rhgs-secondary-n07::myvol-
slave Passive N/A      N/A
rhgs-primary-n02  myvol      /rhgs/bricks/myvol  root      rhgs-secondary-n02::myvol-
slave Passive N/A      N/A
rhgs-primary-n10  myvol      /rhgs/bricks/myvol  root      rhgs-secondary-n09::myvol-
slave Passive N/A      N/A
rhgs-primary-n14  myvol      /rhgs/bricks/myvol  root      rhgs-secondary-n01::myvol-
slave Passive N/A      N/A
rhgs-primary-n03  myvol      /rhgs/bricks/myvol  root      rhgs-secondary-n03::myvol-
slave Active  N/A      Changelog Crawl
rhgs-primary-n09  myvol      /rhgs/bricks/myvol  root      rhgs-secondary-n08::myvol-
slave Active  N/A      Changelog Crawl
rhgs-primary-n11  myvol      /rhgs/bricks/myvol  root      rhgs-secondary-n10::myvol-
slave Active  N/A      Changelog Crawl
rhgs-primary-n13  myvol      /rhgs/bricks/myvol  root      rhgs-secondary-n03::myvol-
slave Active  N/A      Changelog Crawl
rhgs-primary-n19  myvol      /rhgs/bricks/myvol  root      rhgs-secondary-n08::myvol-
slave Active  N/A      Changelog Crawl
rhgs-primary-n17  myvol      /rhgs/bricks/myvol  root      rhgs-secondary-n04::myvol-
slave Active  N/A      Changelog Crawl
rhgs-primary-n05  myvol      /rhgs/bricks/myvol  root      rhgs-secondary-n06::myvol-
slave Active  N/A      Changelog Crawl
rhgs-primary-n15  myvol      /rhgs/bricks/myvol  root      rhgs-secondary-n06::myvol-
slave Active  N/A      Changelog Crawl
rhgs-primary-n16  myvol      /rhgs/bricks/myvol  root      rhgs-secondary-n07::myvol-
```

slave	Passive	N/A	N/A		
rhgs-primary-n07	myvol	/rhgs/bricks/myvol	root	rhgs-secondary-n04::myvol-	
slave	Active	N/A	Changelog Crawl		
rhgs-primary-n20	myvol	/rhgs/bricks/myvol	root	rhgs-secondary-n09::myvol-	
slave	Passive	N/A	N/A		
rhgs-primary-n12	myvol	/rhgs/bricks/myvol	root	rhgs-secondary-n02::myvol-	
slave	Passive	N/A	N/A		
rhgs-primary-n04	myvol	/rhgs/bricks/myvol	root	rhgs-secondary-n01::myvol-	
slave	Passive	N/A	N/A		
rhgs-primary-n08	myvol	/rhgs/bricks/myvol	root	rhgs-secondary-n05::myvol-	
slave	Passive	N/A	N/A		

この時点で、100 TB Gluster ボリュームが完全に使用でき、プライマリーサイドでゾーン間同期データレプリケーションと、別のリージョンにあるセカンダリー側の読み取り専用ボリュームへのリモート非同期データレプリケーションで完全に準備されます。

4.14. データにアクセスするためのクライアントの設定

- NFS および SMB クライアント

NFS プロトコルおよび SMB プロトコルを使用することができますが、ネットワーク設定の制限により、通常通り、CTDB または Pacemaker ではこれらのプロトコルを高可用性にすることはできません。

前提条件

Native Client をインストールする前に、システムを登録およびサブスクライブする必要があります。システムを登録する方法は、『『Red Hat Gluster Storage 管理ガイド』』の「『ボリュームへのアクセスの作成』」の章を参照してください。

4.14.1. ネイティブクライアントのインストール

以下のコマンドを実行してネイティブクライアントをインストールします。

```
# yum -y install glusterfs-fuse attr
```

4.14.2. Red Hat Gluster Storage ボリュームのマウント

ネイティブクライアントをインストールした後に、データにアクセスするために Red Hat Gluster Storage ボリュームがマウントされている必要があります。myvol Gluster ボリュームをローカルディレクトリー /rhgs/client/myvol にマウントします。

```
# mkdir -p /rhgs/client/myvol
```

```
# sh -c 'echo "rhgs-primary-n01:myvol /rhgs/client/myvol \
glusterfs defaults 0 0" >> /etc/fstab'
```

4.14.3. マウントしたボリュームのテスト

以下のコマンドを実行して、マウントされたボリュームをテストします。

```
# mount /rhgs/client/myvol'
```

4.15. 付録: スクラットから RED HAT GLUSTER STORAGE COMPUTE ENGINE イメージのビルド

既存の Red Hat Enterprise Linux パブリックイメージをデプロイし、Red Hat Gluster Storage の階層的なインストールを行うことが可能です。これにより、各 Red Hat Enterprise Linux インスタンスに対して有効な「シブル課金」が作成されます。



注記

Google Compute Engine は、Red Hat サブスクリプションの使用に対する Red Hat Enterprise Linux の公開イメージの使用に対するプレミアム料金を請求します。

レイヤードインストールをデプロイする場合は、Red Hat Subscription Manager でインスタンスを再登録する必要があるため、別途有料の Red Hat Enterprise Linux エンタイトルメントを消費する必要があります。ただし、Subscription Manager に登録した後に、Google Compute Engine はインスタンスのプレミアム料金を課金します。

これを回避するには、Google Compute Engine のプレミアムフェーンにはないカスタムイメージを構築します。ゼロからカスタムイメージを構築する情報については、<https://cloud.google.com/compute/docs/tutorials/building-images> を参照してください。

4.15.1. ISO から RAW ディスクイメージファイルへの Red Hat Gluster Storage のインストール

ローカルの仮想化マネージャーを使用して、RAW 形式のスパースなフラットファイルで仮想マシンを作成して、システムディスクをバックアップします。Red Hat Gluster Storage 3.4 (以降) のシステムディスクに推奨される最小ディスクサイズは 20 GB で、Google Compute Engine にインポートするための最大ディスクサイズは 100 GB です。Google Compute Engine では、ディスクのサイズが GB 全体 (20 GB または 21 GB の増分) が必要ですが、20.5 GB ではありません。RAW ディスクファイルには disk.raw ファイル名が必要です。disk.raw ファイルには、MS-DOS (MBR) パーティションテーブルが含まれている必要があります。

たとえば、以下の **dd** コマンドを実行して、RAW ディスクイメージとして機能する 20 GB のスパースファイルを作成します。

```
# dd if=/dev/zero of=disk.raw bs=1 count=0 seek=20G
```

仮想マシンイメージが Google Compute Engine プラットフォームと互換性があることを確認するには、<https://cloud.google.com/compute/docs/tutorials/building-images#hardwaremanifest> 『Google Compute Engine Hardware Manifest Guide』を参照してください。



注記

以下の手順は、KVM/QEMU をローカルの仮想化プラットフォームとして想定します。

Red Hat カスタマーポータルから利用可能な Red Hat Gluster Storage ISO を起動可能な CD-ROM デバイスとしてイメージにアタッチします。仮想マシンを ISO にブートし、『[Red Hat Gluster Storage 3.5 インストールガイド](#)』の説明に従って、Red Hat Gluster Storage のインストールを行います。

4.15.2. ネットワークインターフェースの有効化と起動

ネットワークインターフェースを有効にして起動するには、次のコマンドを実行します。

- システムの起動時に、デフォルトの eth0 ネットワークインターフェースを有効にします。

```
# sed -i s/ONBOOT=no/ONBOOT=yes/ /etc/sysconfig/network-scripts/ifcfg-eth0
```

- eth0 ネットワークインターフェースを起動します。

```
# ifup eth0
```

4.15.3. Red Hat Gluster Storage Server チャンネルのサブスクライブ

システムを登録し、Red Hat Gluster Storage に必要なチャンネルを有効にする必要があります。適切なプールおよびリポジトリをサブスクライブして接続する方法は、『[Red Hat Gluster Storage 3.5 インストールガイド](#)』の「[Installing Red Hat Gluster Storage chapter](#)」を参照してください。

4.15.4. システムの更新

以下のコマンドを使用してシステムを更新します。

```
# yum -y update
```

4.15.5. チューニングとその他の設定

以下のコマンドを使用して、tuned プロファイルを **rhgs-sequential-io** に設定します。

```
# tuned-adm profile rhgs-sequential-io
```



注記

rhgs-sequential-io プロファイルはこの環境に適していますが、**rhgs-random-io** プロファイルは異なるワークロードに適しています。

SELinux を無効にします。

```
# setenforce 0
```

SELinux サポートが必要な場合は、『[Red Hat Gluster Storage 3.5 インストールガイド](#)』の「[SELinux の有効化](#)」の章を参照してください。

4.15.6. Google Compute Engine の仮想マシンのカスタマイズ

Google Compute Engine の「[Build a Compute Engine Image from Scratch](#)」のドキュメントには、仮想マシンのカーネル、ネットワーク、パッケージ、SSH、およびセキュリティーを設定する具体的な手順が含まれています。Google Compute Engine とイメージの互換性を確保するために、ドキュメントを直接更新して更新しておくことを推奨します。

インスタンスの電源を切ることで、すべての変更を適用し、イメージのインポートを準備します。

```
# init 0
```


4.16. 付録 : RED HAT GLUSTER STORAGE デプロイメントの設定ファイル

ファイル名: glusterfs-config.yaml

```
# Copyright 2015 Google Inc. All Rights Reserved.
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
# http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.

#
# glusterfs-config.yaml
#
# The Gluster FS deployment consists of a primary pool and a secondary pool
# of resources, each on a separate zone.
#

imports:
  - path: gluster_instance.jinja
  - path: path_utils.jinja
resources:
  - name: gluster_instance
    type: gluster_instance.jinja
    properties:
      namePrefix: rhgs
      numPrimaryReplicas: 10
      primaryZone: us-central1-a
      secondaryZone: us-central1-b
      numSecondaryReplicas: 10
      backupZone: europe-west1-b
      sourceImage: global/images/rhgs-image01
      dataSourceImage: global/images/rhgs-data-image01
      machineType: n1-highmem-4
      network: default
      bootDiskType: pd-standard
      dataDiskType: pd-standard
      dataDiskSizeGb: 10230
```

ファイル名: gluster_instance.jinja

```
# Copyright 2015 Google Inc. All Rights Reserved.
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
```

```

# http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.

# GlusterFs configuration variables
#
# Required Cloud resource input parameters:
# * numPrimaryReplicas - number of instances to create in the primary zone
# * numSecondaryReplicas - number of instances to create in the secondary zone
# * namePrefix - hostname prefix
# The instance number (0 based) will be appended ("-n<#><#>")
# * primaryZone - Compute Engine zone for the instance (short name)
# * secondaryZone - Compute Engine zone for the instance (short name)
# * network - Compute Engine network for the instance (full URI)
# * image - Compute Engine image for the instance (full URI)
# * machineType - Compute Engine machine type for the instance (full URI)
# * bootDiskType - Compute Engine boot disk type for the instance (full URI)
# * dataDiskType: Compute Engine data disk type for the instance (full URI)
# * dataDiskSizeGb: Data disk size in Gigabytes

{% import 'path_utils.jinja' as path_utils with context %}

# Grab the config properties
{% set numPrimaryReplicas = properties["numPrimaryReplicas"] + 1%}
{% set numSecondaryReplicas = properties["numSecondaryReplicas"] + 1 %}
{% set image = properties["image"] %}

# Macros and variables dealing with naming
{% set prefix = properties["namePrefix"] %}

{% macro hostname(prefix, id) -%}
{{ "%s-n%02d"|format(prefix, id) }}
{%- endmacro %}

{% macro diskname(prefix, id) -%}
{{ "%s-data-disk-n%02d"|format(prefix, id) }}
{%- endmacro %}

# Expand resource input parameters into full URLs
{% set network = path_utils.networkPath(properties["network"]) %}
{% set primaryZone = properties["primaryZone"] %}
{% set bootDiskType = path_utils.diskTypePath(
    primaryZone, properties["bootDiskType"]) %}
{% set dataDiskType = path_utils.diskTypePath(
    primaryZone, properties["dataDiskType"]) %}
{% set machineType = path_utils.machineTypePath(
    primaryZone, properties["machineType"]) %}

resources:
# Add clone instances in the local Zone
{% for n_suffix in range(1, numPrimaryReplicas) %}

```

```

{% set namePrefix = prefix + '-primary' %}

- type: compute.v1.disk
name: {{ diskname(namePrefix, n_suffix) }}
properties:

  zone: {{ primaryZone }}
  type: {{ dataDiskType }}
  sizeGb: {{ properties["dataDiskSizeGb"] }}
  sourceImage: {{ properties["dataSourceImage"] }}

- type: compute.v1.instance
name: {{ hostname(namePrefix, n_suffix) }}
properties:
  zone: {{ primaryZone }}
  machineType: {{ machineType }}

disks:
# Request boot disk creation (mark for autodelete)
- deviceName: boot
  type: PERSISTENT
  boot: true
  autoDelete: true
  initializeParams:
    sourceImage: {{ properties["sourceImage"] }}
    diskType: {{ bootDiskType }}
    diskSizeGb: 10

# Attach the existing data disk (mark for autodelete)
- deviceName: {{ diskname(namePrefix, n_suffix) }}
  source: $(ref.{{ diskname(namePrefix, n_suffix) }}.selfLink)
  autoDelete: true
  type: PERSISTENT

networkInterfaces:
- network: {{ network }}
  accessConfigs:
  - name: External NAT
    type: ONE_TO_ONE_NAT

tags:
  items:
  - "glusterfs-deployed-from-google-developer-console"

{% endfor %}

# Setup in-region replicas
{% set network = path_utils.networkPath(properties["network"]) %}
{% set secondaryZone = properties["secondaryZone"] %}
{% set bootDiskType = path_utils.diskTypePath(
  secondaryZone, properties["bootDiskType"]) %}
{% set dataDiskType = path_utils.diskTypePath(
  secondaryZone, properties["dataDiskType"]) %}
{% set machineType = path_utils.machineTypePath(
  secondaryZone, properties["machineType"]) %}
{% for n_suffix in range(1, numPrimaryReplicas) %}

```

```

{% set namePrefix = prefix + '-secondary' %}

- type: compute.v1.disk
name: {{ diskname(namePrefix, n_suffix) }}
properties:
  zone: {{ secondaryZone }}
  type: {{ dataDiskType }}
  sizeGb: {{ properties["dataDiskSizeGb"] }}
  sourceImage: {{ properties["dataSourceImage"] }}

- type: compute.v1.instance
name: {{ hostname(namePrefix, n_suffix) }}
properties:
  zone: {{ secondaryZone }}
  machineType: {{ machineType }}

disks:
# Request boot disk creation (mark for autodelete)
- deviceName: boot
  type: PERSISTENT
  boot: true
  autoDelete: true
  initializeParams:
    sourceImage: {{ properties["sourceImage"] }}
    diskType: {{ bootDiskType }}
    diskSizeGb: 10

# Attach the existing data disk (mark for autodelete)
- deviceName: {{ diskname(namePrefix, n_suffix) }}
  source: $(ref.{{ diskname(namePrefix, n_suffix) }}.selfLink)
  autoDelete: true
  type: PERSISTENT

networkInterfaces:
- network: {{ network }}
  accessConfigs:
  - name: External NAT
    type: ONE_TO_ONE_NAT

tags:
  items:
  - "glusterfs-deployed-from-google-developer-console"

{% endfor %}

# Add clone instances in the remote Zone
{% set backupZone = properties["backupZone"] %}
{% set bootDiskType = path_utils.diskTypePath(
  backupZone, properties["bootDiskType"]) %}
{% set dataDiskType = path_utils.diskTypePath(
  backupZone, properties["dataDiskType"]) %}
{% set machineType = path_utils.machineTypePath(
  backupZone, properties["machineType"]) %}
{% for n_suffix in range(1, numSecondaryReplicas) %}

```

```

{% set namePrefix = prefix + '-backup' %}

- type: compute.v1.disk
name: {{ diskname(namePrefix, n_suffix) }}
properties:
  zone: {{ backupZone }}
  type: {{ dataDiskType }}
  sizeGb: {{ properties["dataDiskSizeGb"] }}
#  sourceImage: {{ properties["dataSourceImage"] }}

- type: compute.v1.instance
name: {{ hostname(namePrefix, n_suffix) }}
properties:
  zone: {{ backupZone }}
  machineType: {{ machineType }}

disks:
# Request boot disk creation (mark for autodelete)
- deviceName: boot
  type: PERSISTENT
  boot: true
  autoDelete: true
  initializeParams:
    sourceImage: {{ properties["sourceImage"] }}
    diskType: {{ bootDiskType }}
    diskSizeGb: 10

# Attach the existing data disk (mark for autodelete)
- deviceName: {{ diskname(namePrefix, n_suffix) }}
  source: $(ref.{{ diskname(namePrefix, n_suffix) }}.selfLink)
  autoDelete: true
  type: PERSISTENT

networkInterfaces:
- network: {{ network }}
  accessConfigs:
  - name: External NAT
    type: ONE_TO_ONE_NAT

tags:
  items:
  - "glusterfs-deployed-from-google-developer-console"

{% endfor %}

```

ファイル名: path_utils.jinja

```

# Copyright 2015 Google Inc. All rights reserved.
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
# http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software

```

```
# distributed under the License is distributed on an "AS IS" BASIS,  
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
# See the License for the specific language governing permissions and  
# limitations under the License.  
  
# path_utils.jinja  
#  
# Jinja macros for expanding short resource names into full paths  
# Must have reference to the global env object, so when including this file,  
# use the jinja import "with context" option.  
  
{% macro projectPrefix() -%}  
{{ "https://www.googleapis.com/compute/v1/projects/%s"|format(env["project"]) }}  
{%- endmacro %}  
  
{% macro imagePath(image) -%}  
{% if image.startswith("https://") -%}  
{{ image }}  
{% elif image.startswith("debian-") -%}  
{{ "https://www.googleapis.com/compute/v1/projects/debian-cloud/global/images/" + image }}  
{% elif image.startswith("windows-") -%}  
{{ "https://www.googleapis.com/compute/v1/projects/windows-cloud/global/images/" + image }}  
{% endif -%}  
{%- endmacro %}  
  
{% macro machineTypePath(zone, machineType) -%}  
{% if machineType.startswith("https://") -%}  
{{ machineType }}  
{% else -%}  
{{ "%s/zones/%s/machineTypes/%s"|format(projectPrefix(), zone, machineType) }}  
{% endif -%}  
{%- endmacro %}  
  
{% macro networkPath(network) -%}  
{% if network.startswith("https://") -%}  
{{ network }}  
{% else -%}  
{{ "%s/global/networks/%s"|format(projectPrefix(), network) }}  
{% endif -%}  
{%- endmacro %}  
  
{% macro diskTypePath(zone, diskType) -%}  
{% if diskType.startswith("https://") -%}  
{{ diskType }}  
{% else -%}  
{{ "%s/zones/%s/diskTypes/%s"|format(projectPrefix(), zone, diskType) }}  
{% endif -%}  
{%- endmacro %}
```

付録A 改訂履歴

改訂 3.5-0

Wed Oct 30 2019

Red Hat Gluster Storage
Documentation Team

Red Hat Gluster Storage 3.5 のドキュメントを更新。