



Red Hat Hyperconverged Infrastructure for Virtualization 1.8

Red Hat Hyperconverged Infrastructure for Virtualizationのアップグレード

Red Hat Enterprise Linux 8ベースの RHHI for Virtualization環境への移行方法

Red Hat Hyperconverged Infrastructure for Virtualization 1.8 Red Hat Hyperconverged Infrastructure for Virtualizationのアップグレード

Red Hat Enterprise Linux 8ベースの RHHI for Virtualization環境への移行方法

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 © 2022 | You need to change the HOLDER entity in the en-US/Upgrading_Red_Hat_Hyperconverged_Infrastructure_for_Virtualization.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

本書では、Red Hat Hyperconverged Infrastructure for Virtualization の最新バージョンへのアップグレード方法について説明します。

目次

多様性を受け入れるオープンソースの強化	4
第1章 RHHI FOR VIRTUALIZATIONの概要	5
第2章 バージョン 1.8 の主な変更点	6
第3章 RED HAT HYPERCONVERGED INFRASTRUCTURE FOR VIRTUALIZATION 1.5 および1.6 からのアップグレード	8
3.1. アップグレードのワークフロー	8
3.2. RED HAT HYPERCONVERGED INFRASTRUCTURE FOR VIRTUALIZATION 1.7 へのアップグレード	8
第4章 RED HAT HYPERCONVERGED INFRASTRUCTURE FOR VIRTUALIZATION 1.8 へのアップグレード ..	9
4.1. アップグレードのワークフロー	9
4.2. 前提条件	9
4.3. 制限	9
4.4. 手順	9
4.4.1. Red Hat Virtualization 4.4 ホストエンジンデプロイメント用の新規 gluster ボリュームの作成	10
4.4.2. Gluster 設定のバックアップ	10
4.4.3. 仮想マシンの移行	11
4.4.4. ホストエンジンの設定のバックアップ	11
4.4.5. 自己修復ステータスの確認	12
4.4.6. Red Hat Virtualization Host 4.4 を使用した最初のホストの再インストール	12
4.4.7. バックアップファイルの新規インストールホストへのコピー	13
4.4.8. 新規インストールしたホストへのgluster設定ファイルのリストア	13
4.4.9. 新たにインストールしたホストへの、ホストエンジンのデプロイ	14
4.4.10. 次のホストのアップグレード	16
4.4.11. gluster論理ネットワークのアタッチ	17
4.4.12. 古いホストエンジンのストレージドメインの削除	17
4.4.13. クラスタの互換性の更新	17
4.4.14. データセンターの互換性の更新	18
4.4.15. RHV 4.4で利用可能な新しいglusterボリュームオプションの追加	18
4.4.16. アーカイブと展開したコンテンツの削除	18
4.4.17. トラブルシューティング	18
4.5. アップグレードの検証	20
第5章 マイナーリリース間の更新	21
5.1. 更新ワークフロー	21
5.2. アップデートするシステムの準備	21
5.2.1. サブスクリプションの更新	21
5.2.2. ジオレプリケーションによるデータの同期が行われていないことの確認	21
5.3. ホストエンジン仮想マシンと RED HAT VIRTUALIZATION MANAGER 4.4 の更新	22
5.3.1. ホストエンジン仮想マシンの更新	22
5.3.2. Red Hat Virtualization Manager の更新	22
5.4. ハイパーコンバージドホストのアップグレード	23
5.4.1. Red Hat Virtualization 4.4.2 以降からのアップグレード	24
5.4.2. Red Hat Virtualization 4.4.1 以前からのアップグレード	25
パート I. 参考資料	28
付録A ANSIBLE VAULT で暗号化されたファイルの使用	29
A.1. ファイルの暗号化	29
A.2. 暗号化されたファイルの編集	29
A.3. 暗号化されたファイルの新規パスワードへのキーの指定	29

付録B GLUSTER_VOLUME_INVENTORY.YML ファイルについて	31
B.1. 論理ボリュームを作成するための設定パラメータ	31
付録C ARCHIVE_CONFIG_INVENTORY.YML ファイルについて	33
C.1. ARCHIVE_CONFIG_INVENTORY.YML のバックアップとリストアの設定パラメータ	33
C.2. PLAYBOOK ファイルARCHIVE_CONFIG.YMLの作成	33
C.3. PLAYBOOK ファイルTASKS/BACKUP.YMLの作成	33
付録D HE_GLUSTER_VARS.JSON ファイルについて	36
D.1. 必須の変数	36
D.2. 静的ネットワーク設定に必要な変数	37

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。

第1章 RHHI FOR VIRTUALIZATIONの概要

アップグレードとは、あるバージョンの製品から、同じ製品の新しいリリースに移行することです。このセクションでは、Red Hat Hyperconverged Infrastructure for Virtualization 1.8 をバージョン 1.5、1.6、1.7 からアップグレードする方法を説明します。

コンポーネントの視点からすると、アップグレードでは、以下を実行します。

- ホストエンジン仮想マシンを Red Hat Virtualization Manager バージョン 4.4 にアップグレードする
- 物理ホストを Red Hat Virtualization 4.4 にアップグレードする

第2章 バージョン 1.8 の主な変更点

Red Hat Hyperconverged Infrastructure for Virtualization 1.8 と以前のバージョンとの間には、以下のよう
な違いがありますのでご注意ください。

変更された動作

- Red Hat Hyperconverged Infrastructure for Virtualization 1.8 および Red Hat Virtualization 4.4 は Red Hat Enterprise Linux 8 をベースにしています。Red Hat Enterprise Linux 8 の主な相違点は、『[RHEL 8 の導入における検討事項](#)』を参照してください。
- クラスタのアップグレードでは、アップグレードの途中で容量不足に陥るリスクを軽減するため、glusterディスクに10%以上の空き容量が必要になりました。これは、Red Hat Hyperconverged Infrastructure for Virtualization 1.8 へのアップグレード後に利用できます。[\(BZ#1783750\)](#)
- Webコンソールの「ホスト」タブと「追加のホスト」タブが統合され、従来は両方に表示されていた情報が新たに「ホスト」タブに表示されるようになりました。[\(BZ#1762804\)](#)
- Readcache および readcachesize オプションは、Red Hat Enterprise Linux 8 ベースのオペレーティングシステムではサポートされていないため、VDO ボリュームから削除されました。[\(BZ#1808081\)](#)
- Red Hat Virtualization のサポートに合わせて Quartz スケジューラが標準の Java スケジューラに変更されました。[\(BZ#1797487\)](#)

機能拡張

- 管理者ポータルで、クラスタ内のすべてのホストをワンクリックでアップグレードできるようになりました。これは、Red Hat Hyperconverged Infrastructure for Virtualization 1.8 へのアップグレード後に利用できます。[\(BZ#1721366\)](#)
- 新しい Red Hat Hyperconverged Infrastructure for Virtualization のデプロイメントでは、Network-Bound Disk Encryption (NBDE) を使用した、保存データの暗号化がサポートされるようになりました。[\(BZ#1821248, BZ#1781184\)](#)
- IPv6 ネットワークに対応しました。IPv4 アドレスと IPv6 アドレスの両方を備えた環境はサポートされません。[\(BZ#1721383\)](#)
- 新しいロール、Playbook、インベントリの例が用意されており、以下のタスクを簡素化、自動化できます。
 - アップグレード [\(BZ#1500728, BZ#1832654\)](#)
 - 設定のバックアップおよび復元 [\(BZ#1850488\)](#)
 - ホストの置き換え [\(BZ#1840123\)](#)
 - マルチパスデバイスのブラックリスト [\(BZ#1807808\)](#)
 - gluster論理ネットワークの作成 [\(BZ#1832966\)](#)
 - IPv6 ネットワークへのデプロイ [\(BZ#1688217\)](#)
- Webコンソールで、IPv4 または IPv6 ベースのデプロイメントを選択するオプションを追加しました。[\(BZ#1688798\)](#)

- レプリケーションモジュールの**fsync**が**イーガーロック**機能を使用するようになり、Red Hat Hyperconverged Infrastructure for Virtualization 1.8 で、サイズが 4k にほぼ等しいスモールブロックの書き込みの多いワークロードのパフォーマンスが 50% 以上向上しました。
([BZ#1836164](#))
- Webコンソールでマルチパスデバイスのブラックリスト化に対応しました。([BZ#1814120](#))
- 新しいフェンシングポリシー**skip_fencing_if_gluster_bricks_up**および**skip_fencing_if_gluster_quorum_not_met**が追加され、デフォルトで有効になりました。
([BZ#1775552](#))
- Red Hat Hyperconverged Infrastructure for Virtualization は、ストレージドメインを作成する前に、Red Hat Gluster Storage の「performance.strict-o-direct」オプションが有効になっていることを確認するようになりました。([BZ#1807400](#))
- Red Hat Gluster Storage のボリュームオプションは、管理者ポータルでボリューム名に「all」を使用してすべてのボリュームに設定できるようになりました。([BZ#1775586](#))
- 読み取り専用のフィールドが Web コンソールのユーザーインターフェースに含まれなくなり、インターフェースがシンプルで読みやすくなりました。([BZ#1814553](#))

第3章 RED HAT HYPERCONVERGED INFRASTRUCTURE FOR VIRTUALIZATION 1.5 および1.6 からのアップグレード

3.1. アップグレードのワークフロー

Red Hat Hyperconverged Infrastructure for Virtualization 1.8 にアップグレードする場合の主な要件として、Red Hat Virtualization 4.3 の最新バージョンを使用して Red Hat Hyperconverged Infrastructure for Virtualization 1.7 にアップグレードしてください。RHHI for Virtualizationバージョン1.5、1.6、1.7からRHHI for Virtualization 1.8へのアップグレード手順は以下の通りです。

RHHI for Virtualization 1.5 (RHV 4.2 ベース)

1.5から1.7へのアップグレードを行った後、RHHI for Virtualization 1.8へのアップグレードを行ってください。

RHHI for Virtualization 1.6 (RHV 4.3 ベース)

1.6から1.7へのアップグレードを行った後、RHHI for Virtualization 1.8へのアップグレードを行ってください。

RHHI for Virtualization 1.7 (RHV 4.3.8以降をベース)

現在のセットアップを最新の Red Hat Virtualization 4.3 に更新した後に、RHHI for Virtualization 1.8 にアップグレードします。

3.2. RED HAT HYPERCONVERGED INFRASTRUCTURE FOR VIRTUALIZATION 1.7 へのアップグレード

RHHI for Virtualization 1.5, 1.6 から 1.7 へのアップグレード、および RHHI for Virtualization 1.7 から最新の Red Hat Virtualization 4.3.z へのアップグレードの方法は、「[Upgrading to RHHI for Virtualization 1.7](#)」を参照してください。

第4章 RED HAT HYPERCONVERGED INFRASTRUCTURE FOR VIRTUALIZATION 1.8 へのアップグレード

4.1. アップグレードのワークフロー

Red Hat Hyperconverged Infrastructure for Virtualization (RHVI for Virtualization) 1.8 へのアップグレード手順は、RHVI for Virtualization 1.7 が Red Hat Enterprise Linux 7 プラットフォームを使用しているのに対し、新バージョン 1.8 は Red Hat Enterprise Linux 8 プラットフォームを使用しているため、「yum update」を使用して旧バージョンの RHVI for Virtualization から直接アップグレードできません。

今回は直接のアップグレードではないので、glusterの設定と一緒にエンジンのバックアップを行い、ノードを再インストールします。その後、ホストエンジン用に作成した新しいglusterボリュームを使用して、設定が復元されます。新たにインストールされたノードは、クラスタ内の他のノードと同期ができ、順番にすべてのノードでこの手順を繰り返します。

接続されたホストと仮想マシンは、Manager のアップグレード中も引き続き動作可能です。

4.2. 前提条件

- Red Hat では、仮想マシンのワークロードを最小限に抑えることでアップグレードの時間を短縮することを推奨しています。書き込みの多いワークロードがある場合に、データの同期にかかる時間が長くなるため、アップグレードにかかる時間が長くなります。
- ストレージドメインにスケジュールされたジオレプリケーションセッションがある場合は、Red Hat では、アップグレードウィンドウと重ならないように、これらのスケジュールを削除することを推奨します。
- ジオレプリケーションが進行中の場合は、同期が完了するのを待ってアップグレードを開始します。
- 手順の開始前に、環境内のすべてのデータセンターおよびクラスタにおいて、クラスタの互換レベルがバージョン 4.3 に設定されているようにしてください。

4.3. 制限

- 前バージョンのRHVI for Virtualization環境で重複排除および圧縮機能が有効になっていない場合には、RHVI for Virtualization 1.8へのアップグレード時にこの機能を有効にすることはできません。
- NBDE (Network-Bound Disk Encryption) は、RHVI for Virtualization 1.8の新規導入時にのみサポートされます。この機能はアップグレード時には有効にできません。

4.4. 手順

このセクションでは、RHVI for Virtualization 1.7からRHVI for Virtualization 1.8にアップグレードする手順を説明します。



重要

本セクションで紹介する Playbook は、RHVI for Virtualization 1.7環境でのみ利用可能です。RHVI for virtualizationのバージョン1.5および1.6が最新のRHVI for Virtualization 1.7にアップグレードされていることを確認してください。

4.4.1. Red Hat Virtualization 4.4 ホストエンジンデプロイメント用の新規 gluster ボリュームの作成

手順

エンジン ブリックマウント (`/gluster_bricks/engine`) に、ホストごとのブリックで、新しい Red Hat Virtualization 4.4 ホストエンジンデプロイメント向けに新しい Gluster ボリュームを作成します。

- 各ホストの既存のエンジンブリックのマウントパス/`gluster_bricks/engine`の空き領域を使用して、新しいレプリカ3ボリュームを作成します。

```
# gluster volume create newengine replica 3 host1:/gluster_bricks/engine/newengine
host2:/gluster_bricks/engine/newengine host3:/gluster_bricks/engine/newengine
# gluster volume set newengine group virt
# gluster volume set newengine storage.owner-uid 36
# gluster volume set newengine storage.owner-gid 36
# gluster volume set newengine cluster.granular-entry-heal on
# gluster volume set newengine performance.strict-o-direct on
# gluster volume set newengine network.remote-dio off
# gluster volume start newengine
```

検証

- ブリックの状態は、以下のコマンドで確認できます。

```
# gluster volume status newengine
```

4.4.2. Gluster 設定のバックアップ

前提条件

`tasks/backup.yml` および `archive_config.yml` Playbook は、最新バージョンの RHV 4.3.z(`/etc/ansible/roles/gluster.ansible/playbooks/hc-ansible-deployment`) で利用できます。



重要

`tasks/backup.yml` および `archive_config.yml` が `/etc/ansible/roles/gluster.ansible/playbooks/hc-ansible-deployment` で利用できない場合、これらの Playbook は [archive_config_inventory.yml](#) ファイルの理解から作成できます。

手順

- `/etc/ansible/roles/gluster.ansible/playbooks/hc-ansible-deployment/archive_config_inventory.yml` にある `archive_config_inventory.yml` インベントリーファイルを編集します。

ホスト

クラスター内のすべてのホストの FQDN。

共通変数

デフォルト値は、一般的な変数 `backup_dir`、`nbde_setup`、および `upgrade` については適切です。

```

all:
  hosts:
    host1.example.com:
    host2.example.com:
    host3.example.com:
  vars:
    backup_dir: /archive
    nbde_setup: false
    upgrade: true

```

2. backupfiles タグが含まれる更新されたインベントリーファイルを使用して、**archive_config.yml** Playbook を実行します。

```

# cd /etc/ansible/roles/gluster.ansible/playbooks/hc-ansible-deployment

# ansible-playbook -i archive_config_inventory.yml archive_config.yml --tags backupfiles

```

3. バックアップ設定の tar ファイルは、**/root** 下のすべてのホストに **rhvh-node-<HOSTNAME>-backup.tar.gz** という名前で生成されます。このバックアップ設定 tar ファイルをすべてのホストから別のマシン（バックアップホスト）にコピーします。

検証

- すべてのホストにバックアップ設定ファイルが生成され、別のマシン（バックアップホスト）にコピーされていることを確認します。

4.4.3. 仮想マシンの移行

1. **Compute→Hosts** をクリックし、→最初のホストを選択します。
2. ホスト名をクリックして→「**仮想マシン**」タブを選択します。
3. すべての**仮想マシン**を選択して→移行します。
4. クラスタ内の他のホストに、仮想マシンがすべて移行されるのを待ちます。

4.4.4. ホストエンジンの設定のバックアップ

1. ホストエンジンの**グローバルメンテナンス**を有効にします。**Hosted-engine --deploy** でデプロイしたクラスタ内にあるアクティブなホストの1つで以下のコマンドを実行します。

```
# hosted-engine --set-maintenance --mode=global
```

2. SSH を使用して Hosted Engine 仮想マシンにログインし、**ovirt-engine** サービスを停止します。

```
# systemctl stop ovirt-engine
```

3. ホストエンジン仮想マシンで以下のコマンドを実行し、ホストエンジン仮想マシンからエンジンのバックアップを作成します。

```
# engine-backup --mode=backup --scope=all --file=<backup-file.tar.gz> --log=<logfile>
```

Example:

```
# engine-backup --mode=backup --scope=all --file=engine-backup.tar.gz --log=backup.log
Start of engine-backup with mode 'backup'
scope: all
archive file: engine-backup.tar.gz
log file: backup.log
Backing up:
Notifying engine
- Files
- Engine database 'engine'
- DWH database 'ovirt_engine_history'
Packing into file 'engine-backup.tar.gz'
Notifying engine
Done.
```

4. ホストエンジン仮想マシンのバックアップファイルを別のマシン（バックアップホスト）にコピーします。

```
# scp <backup-file.tar.gz> root@backup-host.example.com:/backup/
```

5. ホストエンジン仮想マシンから **poweroff** コマンドを実行して、ホストエンジン仮想マシンをシャットダウンします。

4.4.5. 自己修復ステータスの確認

1. すべてのレプリカ3ボリュームで保留中の自己修復がないかを確認し、修復が完了するのを待ちます。ホストの1つで以下のコマンドを実行してください。

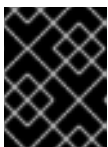
```
# gluster volume heal <volume> info summary
```

2. 保留中の自己修復がないことを確認したら、glusterfsのブリックプロセスを停止して、最初のホスト(現在作業中のホスト)のブリックをすべてアンマウントして、ファイルシステムの整合性を確保します。最初のホストで以下を実行します。

```
# pkill glusterfsd; pkill glusterfs
# systemctl stop glusterd
# umount /gluster_bricks/*
```

4.4.6. Red Hat Virtualization Host 4.4 を使用した最初のホストの再インストール

1. [Red Hat Virtualization Host のインストール](#) ガイドを使用して、OS ディスクのみをフォーマットし、Red Hat Virtualization Host 4.4 ISO でホストを再インストールします。



重要

ブリックはこれらのディスクの上に作成されるので、インストールで他のディスクがフォーマットされないようにしてください。

2. RHVH 4.4のインストール後にノードが稼働したら、Red Hat Virtualization Host(RHVH)4.4のリポジトリに登録するか、カスタマーポータルからダウンロードしたRHV 4.4アプライアンスをインストールしてください。

```
# yum install rhvm-appliance
```


ソフトウェアリポジトリアクセスの設定を参照して、Red Hat Virtualization Host にサブスクライブしてください。

4.4.7. バックアップファイルの新規インストールホストへのコピー

- エンジンのバックアップとホスト設定のtarファイルを、バックアップホストから新しくインストールしたホストにコピーし、コンテンツを展開します。

```
# scp root@backuphost.example.com:/backupdir/engine-backup.tar.gz /root/
# scp root@backuphost.example.com:/backupdir/rhvh-node-host1.example.com-
backup.tar.gz /root/
```

4.4.8. 新規インストールしたホストへのgluster設定ファイルのリストア



注記

バックアップをリストアする前に既存のLVMフィルターを削除し、リストア後にLVMフィルターを再生成するようにしてください。

- 既存のLVMフィルターを削除して、既存の物理ボリューム（PV）を使用できるようにします。

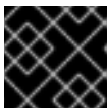
```
# sed -i /^filter/d /etc/lvm/lvm.conf
```

- gluster設定ファイルの内容を展開します。

```
# mkdir /archive
# tar -xvf /root/rhvh-host-host1.example.com.tar.gz -C /archive/
```

- archive_config_inventory.yml** ファイルを編集して、設定ファイルを復元します。**archive_config_inventory.yml** ファイルは、**/etc/ansible/roles/gluster.ansible/playbooks/hc-ansible-deployment/archive_config_inventory.yml** にあります。

```
all:
  hosts:
    host1.example.com:
  vars:
    backup_dir: /archive
    nbde_setup: false
    upgrade: true
```



重要

復元 Playbook の **hosts** セクションでは、ホストを1つだけ使用します。

- Playbook を実行して、設定ファイルを復元します。

```
# cd /etc/ansible/roles/gluster.ansible/playbooks/hc-ansible-deployment
# ansible-playbook -i archive_config_inventory.yml archive_config.yml --tags restorefiles
```

- 新たに識別されたPVに対して、新しいLVMフィルタを再生成します。

-

```
# vdsm-tool config-lvm-filter -y
```

4.4.9. 新たにインストールしたホストへの、ホストエンジンのデプロイ

エンジンからバックアップしたアーカイブを参照する **hosted-engine --deploy --restore-from-file=<engine-backup.tar.gz>** オプションを指定して、ホストエンジンをデプロイします。

ホストエンジンは、**hosted-engine --deploy** コマンドを使用して対話式でデプロイでき、新しく作成されたエンジンボリュームに対応するストレージを指定できます。

ホストエンジンは、**ovirt-ansible-hosted-engine-setup** ロールを使って自動化された方法でデプロイすることもできますが、Red Hatでは、自動化での方法を使用してエラーを回避することを推奨しています。以下の手順では、ホストエンジン仮想マシンを自動でデプロイする方法を説明します。

1. 新たにインストールしたホストに、ホストエンジンデプロイメント用の Playbook を **/etc/ansible/roles/gluster.ansible/playbooks/hc-ansible-deployment/he.yml** に作成します。

```
---
- name: Deploy oVirt hosted engine
  hosts: localhost
  roles:
    - role: ovirt.ovirt.hosted_engine_setup
```

2. **/etc/ansible/roles/gluster.ansible/playbooks/hc-ansible-deployment/he_gluster_vars.json** にある **he_gluster_vars.json** のテンプレートファイルを使用して、ホストエンジン関連の情報を更新します。

```
# cat /etc/ansible/roles/gluster.ansible/playbooks/hc-ansible-deployment/he_gluster_vars.json

{
  "he_appliance_password": "password",
  "he_admin_password": "password",
  "he_domain_type": "glusterfs",
  "he_fqdn": "hostedengine.example.com",
  "he_vm_mac_addr": "00:18:15:20:59:01",
  "he_default_gateway": "19.70.12.254",
  "he_mgmt_network": "ovirtmgmt",
  "he_storage_domain_name": "HostedEngine",
  "he_storage_domain_path": "/newengine",
  "he_storage_domain_addr": "host1.example.com",
  "he_mount_options": "backup-volfile-servers=host2.example.com:host3.example.com",
  "he_bridge_if": "eth0",
  "he_enable_hc_gluster_service": true,
  "he_mem_size_MB": "16384",
  "he_cluster": "Default",
  "he_restore_from_file": "/root/engine-backup.tar.gz",
  "he_vcpus": "4"
}
```



注記

he_gluster_vars.jsonファイルには2つの重要な値があります。

he_restore_from_file

この値はテンプレートでは指定されていないので、追加する必要があります。この値は、ローカルマシンにコピーされたエンジンのバックアップアーカイブの絶対ファイル名を参照する必要があります。

he_storage_domain_path

この値は、新しく作成されたglusterボリュームを参照する必要があります。

ホストエンジン仮想マシン上で動作していた Red Hat Virtualization の旧バージョンがダウンし、破棄されます。以前のホストエンジン仮想マシンに対応するMACアドレスとFQDNは、新しいエンジンにも再利用できます。

3. ホストエンジンのネットワークを静的に設定する場合は、以下のようにオプションを追加します。

he_vm_ip_addr

エンジン仮想マシンのIPアドレス

he_vm_ip_prefix

エンジン仮想マシンのIPプレフィックス

he_dns_addr

エンジン仮想マシンのDNSサーバー

he_default_gateway

エンジン仮想マシンのデフォルトゲートウェイ



注記

利用可能な特定のDNSがない場合は、**he_vm_etc_hosts: true**と**he_network_test: ping**の2つのオプションを追加します。

4. Playbook を実行して、Hosted Engine をデプロイします。

```
# cd /etc/ansible/roles/gluster.ansible/playbooks/hc-ansible-deployment
# ansible-playbook he.yml --extra-vars=@he_gluster_vars.json'
```



重要

Red Hat Enterprise Linux 8.6(RHEL 8.6)をベースとした Red Hat Virtualization Host(RHVH)4.4 SP1 を使用している場合は、**-e 'ansible_python_interpreter=/usr/bin/python3.6'** パラメーターを追加します。

```
# ansible-playbook -e 'ansible_python_interpreter=/usr/bin/python3.6' he.yml -
-extra-vars=@he_gluster_vars.json'
```

5. ホストエンジンのデプロイメントが完了するのを待ちます。



注記

ホストエンジンのデプロイに失敗した場合は、`/var/log/ovirt-hosted-engine-setup`のログメッセージを見て問題を特定して、解決します。失敗したホストエンジンのデプロイメントを、`ovirt-hosted-engine-cleanup`コマンドを使用してクリーンアップし、デプロイメントを再実行します。

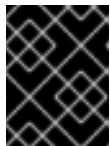
6. 新たにインストールしたRHV managerでRHV 4.4 管理ポータルにログインし、すべてのホストがアップの状態であることを確認します。glusterボリュームの自己修復が完了するのを待ちます。

4.4.10. 次のホストのアップグレード

1. 次のホスト(2台目)に移動し、理想的には順番が次のホストをRHV管理ポータルからメンテナンスモードに切り替え、`gluster`サービスを停止します。
 - a. **Compute** → **Hosts** をクリックし、→ 次のホストを選択します。
 - b. **Management** をクリックして → **Maintenance** を選択すると → **New Maintenance Host(s)** ダイアログボックスが開きます。
 - c. **Stop Gluster**サービスのチェックボックスを選択し、→**OK** を選択します。
2. ホストのコマンドラインからgluster ブリックスをアンマウントします。

```
# umount /gluster_bricks/*
```

3. このホストをRHVH4.4で再インストールします。



重要

ブリックスは、他のディスクに作成されるので、インストール時にこれらのディスクをフォーマットしないようにしてください。

4. gluster設定用tarファイルをバックアップホストから、新規インストールホストにコピーし、コンテンツを展開します。

```
# scp root@backuphost.example.com:/backupdir/rhvh-node-<hostname>-backup.tar.gz /root/
# tar -xvf /root/rhvh-host-<hostname>-backup.tar.gz -C /archive/
```

5. このホストで [Restoring the configuration files on the new installed host](#) で説明した Playbook を実行し、新たにインストールしたホストで gluster設定ファイルを復元します。



注記

`archive_config_inventory.yml` Playbookを編集し、新しくインストールしたホストで実行します。

6. RHV 管理ポータルでホストを再インストールします。
 - a. RHV4.4で最初に導入したホストから認証キーをコピーします。

```
# scp root@host1.example.com:/root/.ssh/authorized_keys /root/.ssh/
```

- b. RHV管理ポータルでは、ホストはメンテナンスモードになります。**Compute → Hosts** をクリックし、**→ Installation → Re-install** を選択します。新規ホストのダイアログボックスが表示されますので、**Hosted Engine** タブを選択し、ホストエンジンのデプロイアクションに **deploy** を選択します。
 - c. ホストが **Up** になるのを待ちます。
7. クラスタ内のすべての Red Hat Virtualization Host 4.3 ホストに対して、**Upgrading next host** の手順を繰り返します。

4.4.11. gluster論理ネットワークのアタッチ

(オプション)クラスタ内に別のgluster論理ネットワークが存在する場合には、ホストごとにそのgluster論理ネットワークを必要なインターフェースにアタッチします。

1. **Compute → Hosts → host** を選択してから **→ Network Interfaces** タブを選択します。
2. **ホストネットワークの設定** ボタンをクリックし、gluster 論理ネットワークを適切なネットワークインターフェースにドラッグアンドドロップします。

4.4.12. 古いホストエンジンのストレージドメインの削除

1. **hosted_storage** という名前で、金色の星が付いていない、以前のホストエンジンストレージドメインを特定します。
 - a. **ストレージ → ドメイン → hosted_storage** を選択して **→ データセンタータブ → メンテナンス** の順にクリックします。
 - b. そのストレージドメインがメンテナンスに切り替わるのを待ちます。
 - c. ストレージドメインがメンテナンス状態になったら、**Detach** をクリックすると、ストレージドメインの**アタッチが解除**されます。
 - d. アタッチが解除されたストレージドメインを選択し、**削除** ボタン **→ OK** の順にクリックします。
2. 古いエンジンのボリュームを停止して削除します。
 - a. **Storage → Volumes** をクリックして **→ old engine volume** を選択し、**→ Stop** ボタンをクリックして **→ OK** で確定します。
 - b. 同じボリュームをクリックして **→ Delete** を選び、**→ OK** で確定します。
3. ハイパーコンバージドホストのエンジンブリックを削除します。

```
# rm -r /gluster_bricks/engine/engine
```



注記

古いエンジンブリックを削除すると、新しいエンジンブリックのディレクトリも **/gluster_bricks/engine** と同じマウントパスに作成されるため、注意が必要です。

4.4.13. クラスタの互換性の更新

- **Compute** → **Clusters** → Select the cluster **Default** → **Edit** → update **Compatibility Version** to 4.6 → **OK** の順に選択します。



注記

VMs on the cluster to be restartedと、互換性バージョンの変更に関する警告が表示されたら、**OK** をクリックします。

4.4.14. データセンターの互換性の更新

1. **Compute** → **Data Centers** の順に選択します。
2. 適切なデータセンターを選択します。
3. **編集** をクリックします。
4. **データセンターの編集** ダイアログボックスが開きます。
5. ドロップダウンリストから **Compatibility Version** を **4.6** に更新します。

4.4.15. RHV 4.4で利用可能な新しいglusterボリュームオプションの追加

RHV 4.4で新しいglusterボリュームオプションが利用できるようになり、これらのボリュームオプションをすべてのボリュームに適用します。

クラスター内のいずれかのノードで以下を実行します。

```
# for vol in `gluster volume list`; do gluster volume set $vol group virt; done
```

4.4.16. アーカイブと展開したコンテンツの削除

バックアップ設定ファイルを展開した内容と、アーカイブをすべてのノードから削除します。

```
# rm -rf /root/rhvh-node-<hostname>-backup.tar.gz
# rm -rf /archive/
```



重要

アップグレード後は、すべてのglusterボリュームのオプション**cluster.lookup-optimize**を無効にします。

```
# for volume in `gluster volume list`; do gluster volume set $volume cluster.lookup-optimize off; done
```

4.4.17. トラブルシューティング

1. GFIDの不一致が原因で、HAエージェントが相互に同期しないことがあります。
 - a. 適切な入出力エラーが**/var/log/ovirt-hosted-engine-ha/broker.log**に表示されます。

```
# grep -i error /var/log/ovirt-hosted-engine-ha/broker.log
```

```
MainThread::ERROR::2020-07-13
06:25:16,188::broker::69::ovirt_hosted_engine_ha.broker.broker.Broker::(run) Failed
initializing the broker: [Errno 5] Input/output error: '/rhev/data-center/mnt/glusterSD/rhsqa-
grafon10.lab.eng.blr.redhat.com:._newengine/1d94d115-8ddd-41c9-bd9c-
477347e95ad4/ha_agent/hosted-engine.lockspace'
```

- b. 次のコマンドを実行して、ボリュームにGFIDの不一致がないか確認します。

```
# grep -i 'gfid mismatch' /var/log/glusterfs/rhev*
```

Example:

```
# grep -i 'gfid mismatch' /var/log/glusterfs/rhev*
```

```
/var/log/glusterfs/rhev-data-center-mnt-glusterSD-rhsqa-
grafon10.lab.eng.blr.redhat.com:._newengine.log:[2020-07-13 06:14:12.992345] E
[MSGID: 108008] [afr-self-heal-common.c:392:afr_gfid_split_brain_source] 0-newengine-
replicate-0: Gfid mismatch detected for <gfid:580f8fe2-a42f-4f62-a5b0-
7591c3740885>/hosted-engine.metadata>, d6a1fe1d-fc04-48cc-953f-d195d40749c1 on
newengine-client-1 and c5e89641-e08f-462f-85ab-13518c21b7dc on newengine-client-0.
```

- c. GFIDの不一致として表示されたエントリーがある場合は、GFIDのスプリットブレインを解決してください。

```
# gluster volume heal <volume> split-brain latest-mtime <relative_path_of_file_in_brick>
```

Example:

```
# gluster volume heal newengine split-brain latest-mtime /1d94d115-8ddd-41c9-bd9c-
477347e95ad4/ha_agent/hosted-engine.lockspace
```

2. RHV 管理ポータルで、gluster ボリュームの状態が degraded と表示され、アップグレードしたノードのブリックの1つが **down** しています。

- a. ハイパーコンバージドホストの gluster コマンドラインから gluster ボリュームのステータスを確認します。アップグレードして再起動されたノードに対応するブリックエントリーが表示され、ブリックのプロセスとポートが **N/A** となっています。次の例では、ホスト **rvh2.example.com** のプロセスIDやポート情報がないことに注意してください。

```
# gluster volume status engine
```

Example:

```
Status of volume: engine
```

```
Gluster process          TCP Port  RDMA Port
```

```
-----
Brick rhvh1.example.com:/gluster_bricks/engine/engine          49158  0
Brick rhvh2.example.com:/gluster_bricks/engine/engine          N/A    N/A
Brick rhvh3.example.com:/gluster_bricks/engine/engine          49152  0
Self-heal Daemon on localhost          N/A    N/A
Self-heal Daemon on rhvh2.example.com  N/A    N/A
Self-heal Daemon on rhvh3.example.com  N/A    N/A
```

```
Online Pid
```

```
-----
Y    94365
Y    11052
Y    31153
Y    128608
Y    11838
Y    9806
```

```
Task Status of Volume engine
-----
```

```
There are no active volume tasks
```

- b. この問題を解決するには、ブリックプロセスを強制終了し、**glusterfsd**サービスを再起動してください。

```
# pkill glusterfsd
# systemctl restart glusterd
```

- c. **gluster volume status** を再度確認し、すべてのブリックエントリにブリックプロセスIDとポート情報が含まれていることを確認します。この情報がRHV管理ポータルに反映されるまで、数分待ちます。

```
# gluster volume status engine
```

4.5. アップグレードの検証

アップグレードが正常に完了したことを確認します。

1. RHV Managerのバージョンを確認してください。

- 管理ポータルにログインし、→ 右上の **Help(? の記号)** → **About** の順に選択します。
 - ソフトウェアのバージョンは、**Software Version:4.4.X.X.X.el8ev**となります。

```
Example: Software Version:4.4.1.8-0.7.el8ev
```

2. ホストのバージョンを確認します。

- すべてのホストで以下のコマンドを実行して、ホストの最新のバージョンを取得します。

```
# nodedctl info | grep default
```

```
Example:
```

```
# nodedctl info | grep default
default: rhvh-4.4.1.1-0.20200707.0 (4.18.0-193.12.1.el8_2.x86_64)
```


第5章 マイナーリリース間の更新

現行バージョンの Red Hat Hyperconverged Infrastructure for Virtualization 1.8 を最新バージョンにアップデートするには、このセクションの手順に従います。

5.1. 更新ワークフロー

Red Hat Hyperconverged Infrastructure for Virtualizationは、複数の異なるコンポーネントで構成されるソフトウェアソリューションです。デプロイメントへの影響を最小限に抑えるため、以下の順序でコンポーネントをアップデートしてください。

1. アップデートするシステムを準備する。
2. ホストエンジン仮想マシンと Red Hat Virtualization Manager 4.4 を更新する。
3. ハイパーコンバージドホストを更新する。

5.2. アップデートするシステムの準備

このセクションでは、アップデート手順のためにシステムを準備する方法を説明します。

5.2.1. サブスクリプションの更新

マシンがどのリポジトリにアクセスできるかを確認するには、ホストエンジン仮想マシン上で root ユーザーとして次のコマンドを実行します。

```
# subscription-manager repos --list-enabled
```

ホストエンジン仮想マシンが以下のリポジトリにサブスクライブされていることを確認します。

- rhel-8-for-x86_64-baseos-rpms
- rhel-8-for-x86_64-appstream-rpms
- rhv-4.4-manager-for-rhel-8-x86_64-rpms
- fast-datapath-for-rhel-8-x86_64-rpms
- jb-eap-7.4-for-rhel-8-x86_64-rpms
- openstack-16.2-cinderlib-for-rhel-8-x86_64-rpms
- rhceph-4-tools-for-rhel-8-x86_64-rpms

ハイパーコンバージドホスト (Red Hat Virtualization ノード) が以下のリポジトリにサブスクライブしていることを確認します。

- rhvh-4-for-rhel-8-x86_64-rpms

上記のリポジトリへのサブスクライブの詳細は、「[Red Hat Virtualization Manager リポジトリの有効化](#)」を参照してください。

5.2.2. ジオレプリケーションによるデータの同期が行われていないことの確認

以下の手順で、ジオレプリケーションが行われているかどうかを確認します。

1. Managerの右下にある**Tasks**タブをクリックします。データ同期に関連する継続的なタスクがないことを確認します。データ同期タスクがある場合は、それらが完了するまで待ってから更新処理を開始してください。
2. スケジュールされたジオレプリケーションセッションをすべて削除し、更新中に同期が行われないようにします。
 - a. **ストレージ→ドメイン**の順にクリックして、→ドメインを選択し、ドメイン名をクリックします。
 - b. **Remote Data Sync Setup**タブ → **Setup**ボタンをクリックします。
 - c. ジオレプリケーションスケジュール設定用の新しいダイアログウィンドウが表示されるので、再帰性を**None**に設定します。

5.3. ホストエンジン仮想マシンと RED HAT VIRTUALIZATION MANAGER 4.4 の更新

このセクションでは、ハイパーコンバージドホストを更新するため、ホストエンジン仮想マシンと Red Hat Virtualization Manager 4.4 を更新する手順について説明します。

5.3.1. ホストエンジン仮想マシンの更新

1. クラスタをグローバルメンテナンスモードに切り替えます。
 - a. いずれかのハイパーコンバージドノードのWebコンソールにログインします。
 - b. **Virtualization → Hosted Engine** をクリックします。
 - c. **Put this cluster into global maintenance**をクリックします。
2. Manager マシンで、更新されたパッケージが利用可能かどうかを確認します。ホストエンジンの仮想マシンにログインし、以下のコマンドを実行します。

```
# engine-upgrade-check
```

5.3.2. Red Hat Virtualization Manager の更新

1. ホストエンジンの仮想マシンにログインします。
2. 以下のコマンドでセットアップパッケージをアップグレードします。

```
# yum update ovirt-engine\*setup\* rh\*vm-setup-plugins
```

3. engine-setup スクリプトで Red Hat Virtualization Manager を更新します。engine-setup スクリプトは以下の作業を行います。
 - 設定に関する質問がプロンプトとして表示されます。
 - ovirt-engine サービスを停止します。
 - 更新したパッケージをダウンロードしてインストールします。

- データベースのバックアップとアップデートを行います。
- インストール後の設定を実行します。
- ovirt-engine サービスを開始します。
 - a. **engine-setup**スクリプトを実行し、画面の指示に従ってManagerをアップグレードします。このプロセスには時間がかかり、中断できないため、Red Hatでは**tmux**セッション内で実行することをお勧めします。

```
# engine-setup
```

スクリプトが正常に完了すると、以下のメッセージが表示されます。

セットアップの実行が正常に完了しました。

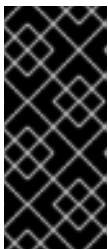


重要

更新プロセスに時間がかかる場合があります。完了する前にプロセスを停止しないでください。

4. その他のパッケージをすべてアップグレードします。

```
# yum update
```



重要

カーネルパッケージが更新された場合は、以下を実行します。

1. グローバルメンテナンスモードを無効にする
2. マシンを再起動して更新を完了する

5. グローバルメンテナンスモードからクラスタを削除します。
 - a. ハイパーコンバージドノードの1つの Webコンソールにログインします。
 - b. **Virtualization → Hosted Engine** をクリックします。
 - c. **Remove this cluster from maintenance**をクリックします。

5.4. ハイパーコンバージドホストのアップグレード

アップグレードプロセスは、ノードが Red Hat Virtualization バージョン 4.4.1 またはバージョン 4.4.2 のどちらを使用しているかによって異なります。

次のコマンドで、使用しているバージョンを確認してください。

```
# cat /etc/os-release | grep "PRETTY_NAME"
```

お使いのバージョンに適したプロセスを実行します。

- [Red Hat Virtualization 4.4.2 以降からのアップグレード](#)

- [Red Hat Virtualization 4.4.1 以前からのアップグレード](#)

5.4.1. Red Hat Virtualization 4.4.2 以降からのアップグレード

1. クラスタ内のハイパーコンバージドホストを1台ずつアップグレードします。
クラスタ内の各ハイパーコンバージドホストでは以下を実行します:
 - a. ハイパーコンバージドホストをアップグレードします。
 - i. Managerで、**Compute→Hosts**をクリックし、ノードを選択します。
 - ii. **Installation → Upgrade** をクリックします。
 - iii. **OK** をクリックしてアップグレードを確定します。
ノードがアップグレードされ、再起動されます。
 - b. 自己修復が完了したことを確認します。
 - i. ホスト名をクリックします。
 - ii. **Bricks** タブをクリックします。
 - iii. **Self-Heal Info** の欄の全ブリックの横に「**OK**」と表示されていることを確認します。
2. クラスタの互換性設定を更新し、新機能を使用できるようにします。
 - a. 管理ポータルにログインします。
 - b. **Cluster** をクリックし、クラスタ名 (**Default**) を選択します。
 - c. **編集** をクリックします。
 - d. クラスタの **互換バージョン** を **4.6** に変更します。



重要

クラスタの互換性は、仮想マシンが再起動されるまで完全には更新されません。メンテナンス期間をスケジュールし、アプリケーション仮想マシンをメンテナンスモードに移行してから、各ノードの仮想マシンをすべて再起動します。

- e. **Compute → Data Centers** をクリックします。
 - f. **編集** をクリックします。
 - g. **Compatibility version** を **4.6** に変更します。
3. データセンターの互換性設定を更新して、新機能を使用できるようにします。
 - a. **Compute → Data Centers** の順に選択します。
 - b. 適切なデータセンターを選択します。
 - c. **編集** をクリックします。
 - d. **データセンターの編集** ダイアログボックスが開きます。

- e. ドロップダウンリストから **Compatibility Version** を **4.6** に更新します。

5.4.2. Red Hat Virtualization 4.4.1 以前からのアップグレード

1. Managerで、**Compute**→**Hosts**をクリックし、ノードを選択します。
2. **Installation** → **Check for Upgrade** をクリックします。これにより、そのホストにホストアップデートが存在するかどうか、バックグラウンドでチェックされます。
3. アップデートが利用可能になると、ホストの横にホストのアップデートが利用可能である旨の通知が表示されます。
4. ホストをメンテナンスモードに切り替えます。
 - a. RHV 管理ポータルで、**Hosts** に移動して → ホストを選択します。
 - b. **Management** → **Maintenance** の順にクリックすると、→ **Maintenance Host** ダイアログボックスが開きます。
 - c. **Maintenance Host** ダイアログボックスで、**Stop Gluster service** ボックスにチェックを入れて、→ **OK** をクリックします。
5. ホストが **maintenance mode** に変わったら、**Installation** → **Upgrade** をクリックします。**Upgrade Host dialog box** が表示されるので、**Reboot host after upgrade**のチェックを外しておいてください。
6. **OK** をクリックしてアップグレードを確定します。
7. アップグレードが完了するまで待機します。
8. 以下のコマンドを使用して、アップグレードしたホストで既存のLVMフィルターを削除してから再起動します。

```
# sed -i /^filter/d /etc/lvm/lvm.conf
```

9. ホストを再起動します。
10. ホストが再起動したら、LVMフィルタを再生成します。

```
# vdsm-tool config-lvm-filter -y
```

11. 自己修復が完了したことを確認してから次のホストをアップグレードしてください。
 - a. ホスト名をクリックします。
 - b. **Bricks** タブをクリックします。
 - c. 次のホストをアップグレードする前に、すべてのブリックの**Self-Heal information column**がOKと表示されていることを確認します。
12. 他のハイパーコンバージドホストでも上記の手順を繰り返します。
13. クラスタの互換性設定を更新し、新機能を使用できるようにします。
 - a. 管理ポータルにログインします。
 - b. **Cluster** をクリックし、クラスタ名 (**Default**) を選択します。

- c. **編集** をクリックします。
- d. クラスターの **互換バージョン** を **4.6** に変更します。



重要

クラスターの互換性は、仮想マシンが再起動されるまで完全には更新されません。メンテナンス期間をスケジュールし、アプリケーション仮想マシンをメンテナンスモードに移行してから、各ノードの仮想マシンをすべて再起動します。

- e. **Compute → Data Centers** をクリックします。
 - f. **編集** をクリックします。
 - g. **Compatibility version** を **4.6** に変更します。
14. データセンターの互換性設定を更新して、新機能を使用できるようにします。
- a. **Compute → Data Centers**の順に選択します。
 - b. 適切なデータセンターを選択します。
 - c. **編集** をクリックします。
 - d. **データセンターの編集** ダイアログボックスが開きます。
 - e. ドロップダウンリストから **Compatibility Version** を **4.6** に更新します。



重要

更新後に、すべてのglusterボリュームのオプション**cluster.lookup-optimize**を無効にします。

```
# for volume in `gluster volume list`; do gluster volume set $volume cluster.lookup-optimize off; done
```

トラブルシューティング

1. 自己修復プロセスは、各ハイパーコンバージドホストが再起動後に自動的に開始されます。コマンドで自己修復の状態を確認します。

```
# gluster volume heal <volname> info summary
```

自己修復のエントリーが長期間保留されている場合は、以下を確認してください。

- a. Glusterネットワークが接続されている。

```
# ip addr show <ethernet-interface>
```

- b. ボリューム内のすべてのブリックプロセスがアップしている。

```
# gluster volume status <vol>
```

ブリックのプロセスが停止していると報告されている場合は、ブリックが停止していると報告されているノードで**glusterd**サービスを再起動します。

```
# systemctl restart glusterd
```

2. Red Hat Virtualization ノードが起動できずにメンテナンスシェルで起動される場合には、原因の1つとして、不安定な LVM フィルターが一部の物理ボリューム (PV) を拒否していることが挙げられます。

- a. **root** のパスワードでメンテナンスシェルにログインします。
- b. 既存のLVMフィルターの設定を削除します。

```
# sed -i /^filter/d /etc/lvm/lvm.conf
```

- c. ホストを再起動します。
- d. ノードが起動したら、LVMフィルタを再生成します。

```
# vdsm-tool config-lvm-filter -y
```

パート I. 参考資料

付録A ANSIBLE VAULT で暗号化されたファイルの使用

Red Hat は、パスワードやその他の機密情報を含むデプロイメントおよび管理ファイルの内容を暗号化することを推奨します。Ansible Vault は、これらのファイルを暗号化する1つの方法です。Ansible Vault の詳細は、[Ansible documentation](#) を参照してください。

A.1. ファイルの暗号化

ansible-vault create コマンドで暗号化ファイルを作成したり、**ansible-vault encrypt** コマンドで既存のファイルを暗号化することができます。

暗号化ファイルを作成したり、既存のファイルを暗号化したりすると、パスワードを指定するように求められます。このパスワードは、暗号化後にファイルを復号化するために使用されます。このファイル内の情報を直接使用する場合や、ファイルの内容に依存する Playbook を実行する場合は、このパスワードを指定する必要があります。

暗号化されたファイルの作成

```
$ ansible-vault create variables.yml
New Vault password:
Confirm New Vault password:
```

ansible-vault create コマンドは新しいファイルのパスワードを要求し、保存する前にファイルを生成できるようにデフォルトのテキストエディター（シェル環境で **\$EDITOR** として定義）で新規ファイルを開きます。

すでにファイルを作成していて、それを暗号化したい場合は、**ansible-vault encrypt** コマンドを使用します。

既存のファイルの暗号化

```
$ ansible-vault encrypt existing-variables.yml
New Vault password:
Confirm New Vault password:
Encryption successful
```

A.2. 暗号化されたファイルの編集

暗号化されたファイルを編集するには、**ansible-vault edit** コマンドを使用し、そのファイルのVaultパスワードを指定します。

暗号化ファイルの編集

```
$ ansible-vault edit variables.yml
New Vault password:
Confirm New Vault password:
```

ansible-vault edit コマンドは、ファイルのパスワードの入力を求め、デフォルトのテキストエディタ（シェル環境で **\$EDITOR** と定義されている）でファイルを開き、ファイルの内容を編集および保存ができるようにします。

A.3. 暗号化されたファイルの新規パスワードへのキーの指定

ansible-vault rekey コマンドを使用して、ファイルの復号化に使用するパスワードを変更できます。

```
$ ansible-vault rekey variables.yml  
Vault password:  
New Vault password:  
Confirm New Vault password:  
Rekey successful
```

ansible-vault rekey コマンドは、現在の Vault パスワードの入力を要求し、新しい Vault パスワードを設定して確認するプロンプトを表示します。

付録B GLUSTER_VOLUME_INVENTORY.YML ファイルについて

`gluster_volume_inventory.yml` インベントリファイルは、空き容量がある場合に、既存のボリュームグループから論理ボリュームを作成時に使用できるサンプルファイルです。

`/etc/ansible/roles/gluster.ansible/playbooks/hc-ansible-deployment/gluster_volume_inventory.yml` に、このファイルを作成してください。

B.1. 論理ボリュームを作成するための設定パラメータ

hosts

ハイパーコンバージドホストのバックエンドネットワークFQDN。 `cluster_nodes` セクションに同じホスト名のセットを記載します。

```
hosts:
  <host1-backend-FQDN>:
  <host2-backend-FQDN>:
  <host3-backend-FQDN>:
```

vgname

ホスト上の既存のボリュームグループ (VG) の名前。

gluster_infra_disktype

ディスクアグリゲーションタイプ。対応できる値:

- RAID6
- RAID5
- JBOD

gluster_infra_diskcount

RAID セット内のデータディスクの数。JBODの場合、値は1です。

gluster_infra_stripe_unit_size

RAID ストライプサイズJBODではこのパラメータは無視してください。

gluster_volume_inventoryファイルの例

```
hc_nodes:
  hosts:
    host1-backend.example.com:
    host2-backend.example.com:
    host3-backend.example.com:

# Common configurations
vars:
  gluster_infra_volume_groups:
    - vgname: gluster_vg_sdb
      pvname: /dev/sdb

  gluster_infra_mount_devices:
    - path: /gluster_bricks/newengine
      lvname: gluster_lv_newengine
```

```
vgname: gluster_vg_sdb
```

```
gluster_infra_thick_lvs:
```

- vgname: gluster_vg_sdb
- lvname: gluster_lv_newengine
- size: 100G

```
gluster_infra_disktype: RAID6
```

```
gluster_infra_diskcount: 10
```

```
gluster_infra_stripe_unit_size: 256
```

```
gluster_features_force_varlogsizecheck: false
```

```
gluster_set_selinux_labels: true
```

```
cluster_nodes:
```

- host1-backend.example.com
- host2-backend.example.com
- host3-backend.example.com

```
gluster_features_hci_cluster: "{{ cluster_nodes }}"
```

```
gluster_features_hci_volumes:
```

- volname: newengine
- brick: /gluster_bricks/newengine/newengine
- arbiter: 0

付録C ARCHIVE_CONFIG_INVENTORY.YML ファイルについて

archive_config_inventory.yml ファイルは、Ansible インベントリーファイルのサンプルで、このファイルを使用して Red Hat Hyperconverged Infrastructure for Virtualization クラスターの設定をバックアップおよび復元できます。

このファイルは、いずれかのハイパーコンバージドホストの `/etc/ansible/roles/gluster.ansible/playbooks/hc-ansible-deployment/archive_config_inventory.yml` にあります。

archive_config.yml と **backup.yml** の2つの Playbook があります。**archive_config.yml** はラッパーの Playbook で、**tasks/backup.yml** をインポートします。

C.1. ARCHIVE_CONFIG_INVENTORY.YML のバックアップとリストアの設定パラメータ

hosts

バックアップするクラスター内の各ホストのバックエンドFQDN。

backup_dir

バックアップファイルを保存するディレクトリー。

nbde_setup

アップグレードではNBDEの設定をサポートしていないため、`false`に設定しています。

upgrade

`true` に設定します。

以下に例を示します。

```
all:
  hosts:
    host1:
    host2:
    host3:
  vars:
    backup_dir: /archive
    nbde_setup: false
    upgrade: true
```

C.2. PLAYBOOK ファイルARCHIVE_CONFIG.YMLの作成

archive_config.yml Playbook ファイルが `/etc/ansible/roles/gluster.ansible/playbooks/hc-ansible-deployment` に存在しない場合のみ作成します。

以下の内容を **archive_config.yml** に追加します。

```
---
- import_playbook: tasks/backup.yml
  tags: backupfiles
```

C.3. PLAYBOOK ファイルTASKS/BACKUP.YMLの作成

tasks/backup.yml Playbook ファイルが `/etc/ansible/roles/gluster.ansible/playbooks/hc-ansible-deployment` に存在しない場合のみ作成します。

backup.yml ファイルに以下の内容を追加します。

```
---
- hosts: all
  tasks:
  - name: Check if backup dir is already available
    stat:
      path: "{{ backup_dir }}"
    register: result

  - fail:
      msg: Backup directory "{{backup_dir}}" exists, remove it and retry
    when: result.stat.isdir is defined

  - name: Create temporary backup directory
    file:
      path: "{{ backup_dir }}"
      state: directory

  - name: Get the hostname
    shell: uname -n
    register: hostname

  - name: Add hostname details to archive
    shell: echo {{ hostname.stdout }} > {{ backup_dir }}/hostname

  - name: Dump the IP configuration details
    shell: ip addr show > {{ backup_dir }}/ipconfig

  - name: Dump the IPv4 routing information
    shell: ip route > {{ backup_dir }}/ip4route

  - name: Dump the IPv6 routing information
    shell: ip -6 route > {{ backup_dir }}/ip6route

  - name: Get the disk layout information
    shell: lsblk > {{ backup_dir }}/lsblk

  - name: Get the mount information for reference
    shell: df -Th > {{ backup_dir }}/mount

  - name: Check for VDO configuration
    stat:
      path: /etc/vdoconf.yml
    register: vdoconfstat

  - name: Copy VDO configuration, if available
    shell: cp -a /etc/vdoconf.yml "{{backup_dir}}"
    when: vdoconfstat.stat.isreg is defined

  - name: Backup fstab
    shell: cp -a /etc/fstab "{{backup_dir}}"
```

- name: Backup glusterd config directory
shell: cp -a /var/lib/glusterd "{{backup_dir}}"

- name: Backup /etc/crypttab, if NBDE is enabled
shell: cp -a /etc/crypttab "{{ backup_dir }}"
when: nbde_setup is defined and nbde_setup

- name: Backup keyfiles used for LUKS decryption
shell: cp -a /etc/sd*keyfile "{{ backup_dir }}"
when: nbde_setup is defined and nbde_setup

- name: Check for the inventory file generated from cockpit
stat:
 path: /etc/ansible/hc_wizard_inventory.yml
register: inventory

- name: Copy the host inventory file generated from cockpit
shell: cp /etc/ansible/hc_wizard_inventory.yml {{ backup_dir }}
when: inventory.stat.isreg is defined

- name: Create a tar.gz with all the contents
archive:
 path: "{{ backup_dir }}/*"
 dest: /root/rhvh-node-{{ hostname.stdout }}-backup.tar.gz

付録D HE_GLUSTER_VARS.JSON ファイルについて

he_gluster_vars.json ファイルは、Ansible 変数ファイルの例です。Red Hat Hyperconverged Infrastructure for Virtualization をデプロイするには、このファイルの変数を定義する必要があります。

例となるファイルは、任意のハイパーコンバージドホストの `/etc/ansible/roles/gluster ansible/playbooks/hc-ansible-deployment/he_gluster_vars.json` にあります。

he_gluster_vars.json ファイルの例

```
{
  "he_appliance_password": "encrypt-password-using-ansible-vault",
  "he_admin_password": "UI-password-for-login",
  "he_domain_type": "glusterfs",
  "he_fqdn": "FQDN-for-Hosted-Engine",
  "he_vm_mac_addr": "Valid MAC address",
  "he_default_gateway": "Valid Gateway",
  "he_mgmt_network": "ovirtmgmt",
  "he_storage_domain_name": "HostedEngine",
  "he_storage_domain_path": "/engine",
  "he_storage_domain_addr": "host1-backend-network-FQDN",
  "he_mount_options": "backup-volfile-servers=host2-backend-network-FQDN:host3-backend-network-FQDN",
  "he_bridge_if": "interface name for bridge creation",
  "he_enable_hc_gluster_service": true,
  "he_mem_size_MB": "16384",
  "he_cluster": "Default",
  "he_vcpus": "4"
}
```

Red Hat は、このファイルの暗号化を推奨します。詳細は、[Working with files encrypted using Ansible Vault](#) を参照してください。

D.1. 必須の変数

he_appliance_password

ホストされるエンジンのパスワード。実稼働クラスターの場合、Ansible Vault で作成された暗号化された値を使用します。

he_admin_password

ホストされたエンジンの **admin** アカウントのパスワードです。実稼働クラスターの場合、Ansible Vault で作成された暗号化された値を使用します。

he_domain_type

ストレージドメインのタイプ。**glusterfs** に設定します。

he_fqdn

ホスト型エンジン仮想マシンの FQDN。

he_vm_mac_addr

ホストされるエンジン仮想マシンの適切なネットワークデバイスの MAC アドレス。このオプションは、静的 IP 設定で、Hosted Engine の MAC アドレスが自動的に生成されます。

he_default_gateway

使用するゲートウェイの FQDN。

he_mgmt_network

管理ネットワークの名前。 **ovirtmgmt** に設定します。

he_storage_domain_name

ホストエンジン用に作成するストレージドメインの名前。 **HostedEngine** に設定します。

he_storage_domain_path

ストレージドメインを提供する Gluster ボリュームのパス。 **/engine** に設定します。

he_storage_domain_addr

engine ドメインを提供する最初のホストのバックエンド FQDN。

he_mount_options

追加のマウントオプションを指定します。

For a three node deployment with IPv4 configurations, set:

```
"he_mount_options": "backup-volfile-servers=host2-backend-network-FQDN:host3-backend-network-FQDN"
```

he_mount_optionは、Red Hat Hyperconverged Infrastructure for Virtualization の IPv4 ベースのシングルノード展開には必要ありません。

3つのノードでIPv6の設定を行う場合は、次のように設定します。

```
"he_mount_options": "backup-volfile-servers=host2-backend-network-FQDN:host3-backend-network-FQDN,xlator-option='transport.address-family=inet6'"
```

IPv6 設定を使用する単一ノードのデプロイメントでは、以下を設定します。

```
"he_mount_options": "xlator-option='transport.address-family=inet6'"
```

he_bridge_if

ブリッジの作成に使用するインターフェースの名前。

he_enable_hc_gluster_service

Gluster サービスを有効にします。 **true** に設定します。

he_mem_size_MB

ホストエンジン仮想マシンに割り当てられるメモリー量（メガバイト単位）。

he_cluster

ハイパーコンバージドホストが配置されるクラスタの名前。

he_vcpus

エンジン仮想マシンで使用される CPU の量。デフォルトでは、ホストエンジンの仮想マシンには 4 つの VCPU が割り当てられます。

D.2. 静的ネットワーク設定に必要な変数

DHCP 設定は、デフォルトでホスト型エンジン仮想マシンで使用されます。ただし、静的 IP または FQDN を使用する場合は、以下の変数を定義します。

he_vm_ip_addr

ホスト型エンジン仮想マシン (IPv4 または IPv6) の静的 IP アドレス。

he_vm_ip_prefix

ホスト型エンジン仮想マシン (IPv4 または IPv6) の IP プレフィックス

he_dns_addr

ホスト型エンジン仮想マシン (IPv4 または IPv6) 用の DNS サーバー。

he_default_gateway

ホスト型エンジン仮想マシン (IPv4 または IPv6) のデフォルトゲートウェイ。

he_vm_etc_hosts

ホストエンジンの仮想マシンの IP アドレスと FQDN をホストの `/etc/hosts` に指定します (ブール値)。

静的ホスト型エンジン設定を含む `he_gluster_vars.json` ファイルの例

```
{
  "he_appliance_password": "mybadappliancepassword",
  "he_admin_password": "mybadadminpassword",
  "he_domain_type": "glusterfs",
  "he_fqdn": "engine.example.com",
  "he_vm_mac_addr": "00:01:02:03:04:05",
  "he_default_gateway": "gateway.example.com",
  "he_mgmt_network": "ovirtmgmt",
  "he_storage_domain_name": "HostedEngine",
  "he_storage_domain_path": "/engine",
  "he_storage_domain_addr": "host1-backend.example.com",
  "he_mount_options": "backup-volfile-servers=host2-backend.example.com:host3-backend.example.com",
  "he_bridge_if": "interface name for bridge creation",
  "he_enable_hc_gluster_service": true,
  "he_mem_size_MB": "16384",
  "he_cluster": "Default",
  "he_vm_ip_addr": "10.70.34.43",
  "he_vm_ip_prefix": "24",
  "he_dns_addr": "10.70.34.6",
  "he_default_gateway": "10.70.34.255",
  "he_vm_etc_hosts": "false",
  "he_network_test": "ping"
}
```



注記

DNS が利用できない場合は、`dns` の代わりに `he_network_test` の `ping` を使用してください。

Example: `"he_network_test": "ping"`