



# Red Hat OpenStack Platform 13

## カスタムブロックストレージバックエンドデプロイメントガイド

Red Hat OpenStack Platform オーバークラウドでのカスタムの Block Storage バックエンドのデプロイガイド



# Red Hat OpenStack Platform 13 カスタムブロックストレージバックエンドデプロイメントガイド

---

Red Hat OpenStack Platform オーバークラウドでのカスタムの Block Storage バックエンドのデプロイガイド

OpenStack Team  
rhos-docs@redhat.com

## 法律上の通知

Copyright © 2023 Red Hat, Inc.

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<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

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

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js<sup>®</sup> 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<sup>®</sup> 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

このドキュメントでは、Red Hat OpenStack Platform 13 オーバークラウドに Block Storage サービス用の統合されていないカスタムバックエンドをデプロイする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 .....	3
第1章 はじめに .....	4
1.1. カスタムバックエンド .....	4
1.2. 要件 .....	5
第2章 プロセスの説明 .....	6
第3章 環境ファイルの作成 .....	7
第4章 設定済みのバックエンドのデプロイ .....	9
第5章 設定したバックエンドのテスト .....	10
付録A 付録 .....	11
A.1. STACK ユーザー .....	11
A.2. サンプル環境ファイルから得られる設定 .....	11



## 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、[Red Hat CTO である Chris Wright のメッセージ](#) をご覧ください。

## 第1章 はじめに

Red Hat OpenStack Platform (RHOSP) director は、完全な RHOSP 環境のインストールおよび管理を行うためのツールセットです。主にアップストリームの TripleO (OpenStack-on-OpenStack) プロジェクトに基づいています。Director の主な目的は、最小限の手動設定で、機能的なエンタープライズグレードの RHOSP デプロイメントを完全に調整することです。OpenStack の個々のコンポーネントを手動で設定する際に生じる問題の多くを解決するのに役立ちます。

Director が提供する最終結果の RHOSP デプロイメントは、**オーバークラウド**と呼ばれます。オーバークラウドには、Block Storage など、エンドユーザーにサービスを提供するすべてのコンポーネントが含まれます。このドキュメントでは、カスタムバックエンドをオーバークラウドの Block Storage サービスにデプロイする方法についてのガイダンスを提供します。

このドキュメントは、Block Storage の手動設定に関連する概念についての知識があることを前提としています。OpenStack のテストデプロイメント (たとえば、Packstack を使用) では、このサービスを設定するには、そのホストノードの `/etc/cinder/cinder.conf` を編集する必要があります。そのファイルの Block Storage 設定のほとんどは、別の場所でより詳細に文書化されています。このドキュメントでは、同じ設定をオーバークラウドに適用してカスタムバックエンドをアタッチする方法について説明します。



### 警告

この手順は、一部のユースケースで正常にテストされています。実稼働以外の環境で計画したデプロイメントを最初にテストするようにしてください。ご質問がございましたら、Red Hat サポートにお問い合わせください。

### 1.1. カスタムバックエンド

このドキュメントでは、カスタムバックエンドは、Red Hat OpenStack Platform の Director にまだ完全に統合されていないストレージサーバー、アプライアンス、または設定として定義されます。サポートされている一部の Block Storage バックエンドは、すでに Director に統合されています。これは、事前設定された Director ファイルがすでに提供されていることを意味します。これらのファイルを使用して、統合バックエンドを設定し、オーバークラウドにデプロイできます。統合されたバックエンドの例には、Red Hat Ceph と、Dell EMC PS シリーズ、Dell Storage Center、および NetApp アプライアンスのシングルバックエンド設定が含まれます。

さらに、すでに Director に統合されている一部のストレージアプライアンスは、単一インスタンスのバックエンドのみをサポートします。たとえば、Dell Storage Center 用の事前設定された Director ファイルは、単一のバックエンドのデプロイメントのみをサポートします。このアプライアンスの複数のバックエンドインスタンスをデプロイするには、このドキュメントで説明されているように、カスタム設定が必要です。

ノードの `/etc/cinder/cinder.conf` ファイルを直接編集して Block Storage サービスを手動で設定することもできますが、**openstack overcloud deploy** コマンドを実行すると、director によって設定が上書きされます。このため、Block Storage バックエンドをデプロイするための推奨される方法は、Director を使用することです。これにより、設定がオーバークラウドのデプロイメントおよび更新を通じて保持されることが保証されます。

バックエンド設定がすでに完全に統合されている場合は、パッケージ化された環境ファイルを編集して呼び出すことができます。ただし、カスタムバックエンドでは、独自の環境ファイルを作成する必要があります。詳しい情報は、**Director Installation and Usage** の [Including Environment Files in Overcloud](#)



[Creation](#) を参照してください。このドキュメントには、独自のデプロイ用に編集できるアノテーション付きのサンプル [/home/stack/templates/custom-env.yaml](#) が含まれています。このサンプルファイルは、2つの NetApp バックエンドを使用するように Block Storage サービスを設定するのに適しています。

## 1.2. 要件

### 前提条件

- Block Storage とデプロイするバックエンドを手動で設定するための予備知識も説明します。
- サードパーティーのバックエンドアプライアンスを使用している場合は、ストレージリポジトリとして適切に設定されている必要があります。
- オーバークラウドは、Director を使用してすでにデプロイされています。[Director のインストールと使用](#) ガイドを参照してください。
- 昇格した特権を持つアカウントのユーザー名およびパスワードを所有している。オーバークラウドをデプロイするために作成したものと同一アカウントを使用できます。[Director のインストールと使用](#) ガイドの [スタックユーザーの作成](#) を参照してください。**stack** ユーザーは、この目的のために作成されます。
- `/etc/cinder/cinder.conf` で、Block Storage バックエンドに必要な結果の設定をすでにマッピングしています。

## 第2章 プロセスの説明

Block Storage サービスの設定は `/etc/cinder/cinder.conf` に保存されます。これらの設定には、**バックエンド定義**が含まれます。Block Storage サービスで使用できる (またはサポートされている) サードパーティーのバックエンドのほとんどは、`/etc/cinder/cinder.conf` 設定の編集を含むセットアップ手順を提供します。[1章はじめに](#)で述べたように、これにより、Block Storage サービスが設定されます。ただし、これらの設定は今後のオーバークラウドの更新で上書きされます。

いずれにせよ、`/etc/cinder/cinder.conf` による手動設定に関するドキュメントは、オーバークラウドのデプロイメントに役立ちます。結局、Director は Heat を介して同じ設定を `/etc/cinder/cinder.conf` に適用します。そのため、バックエンド設定を計画するには、次のことが必要です。

- 必要な Block Storage バックエンド設定を綿密に計画し、
- この設定の結果の `/etc/cinder/cinder.conf` ファイルをマッピングします。

結果の `/etc/cinder/cinder.conf` ファイルをマッピングしたら、バックエンド設定を調整する環境ファイルを作成します。[環境ファイル](#)では、サンプルファイル `/home/stack/templates/custom-env.yaml` を使用して、この手順について詳しく説明しています。環境ファイルを手元に用意しておく、今後のオーバークラウドの更新後もバックエンド設定が維持されるようになります。

## 第3章 環境ファイルの作成

環境ファイルには、定義する各バックエンドの設定と、その他の関連する設定が含まれています。環境ファイルについての詳細は、[オーバークラウドの高度なカスタマイズ](#)の [環境ファイル](#) を参照してください。

以下のサンプル環境ファイルは、`netapp1` および `netapp2` の2つの NetApp バックエンドを定義します。

`/home/stack/templates/custom-env.yaml`

```
parameter_defaults: # 1
  CinderEnableScsiBackend: false
  CinderEnableRbdBackend: false
  CinderEnableNfsBackend: false
  NovaEnableRbdBackend: false
  GlanceBackend: file # 2
  ControllerExtraConfig: # 3
    cinder::config::cinder_config:
      netapp1/volume_driver: # 4
        value: cinder.volume.drivers.netapp.common.NetAppDriver
      netapp1/netapp_storage_family:
        value: ontap_7mode
      netapp1/netapp_storage_protocol:
        value: iscsi
      netapp1/netapp_server_hostname:
        value: 10.35.64.11
      netapp1/netapp_server_port:
        value: 80
      netapp1/netapp_login:
        value: root
      netapp1/netapp_password:
        value: p@$w0rd
      netapp1/volume_backend_name:
        value: netapp1
      netapp2/volume_driver: # 5
        value: cinder.volume.drivers.netapp.common.NetAppDriver # 6
      netapp2/netapp_storage_family:
        value: ontap_7mode
      netapp2/netapp_storage_protocol:
        value: iscsi
      netapp2/netapp_server_hostname:
        value: 10.35.64.11
      netapp2/netapp_server_port:
        value: 80
      netapp2/netapp_login:
        value: root
      netapp2/netapp_password:
        value: p@$w0rd
      netapp2/volume_backend_name:
        value: netapp2
    cinder_user_enabled_backends: ['netapp1','netapp2'] # 7
```

- 1 次のパラメーターは **false** に設定されているため、他のバックエンドタイプが無効になっています。
  - **CinderEnableIscsiBackend**: 他の iSCSI バックエンド。
  - **CinderEnableRbdBackend**: Red Hat Ceph。
  - **CinderEnableNfsBackend**: NFS。
  - **NovaEnableRbdBackend**: 一時 Red Hat Ceph ストレージ。
- 2 **GlanceBackend** パラメーターは、Image サービスがイメージを保存するために使用するものを設定します。以下の値がサポートされます。
  - **ファイル**: 各コントローラーノードの **/var/lib/glance/images** にイメージを保存します。
  - **Swift**: イメージストレージに Object Storage サービスを使用します。
  - **Cinder**: イメージストレージに Block Storage サービスを使用します。
- 3 **ControllerExtraConfig** は、すべてのコントローラーノードに適用されるカスタム設定を定義します。**cinder::config::cinder\_config** クラスは、設定を Block Storage (**cinder**) サービスに適用する必要があることを意味します。これは、バックエンド設定が最終的に各コントローラーノードの **/etc/cinder/cinder.conf** ファイルで終了することを意味します。
- 4 **netapp1/volume\_driver** および **netapp2/volume\_driver** 設定は、**section/setting** の構文に準拠します。Block Storage サービスでは、各バックエンドは **/etc/cinder/cinder.conf** の独自のセクションで定義されます。**netapp1** 接頭辞を使用する各設定は、新しい **netapp1** バックエンドセクションで定義されます。
- 5 同様に、**netapp2** 設定は別の **netapp2** セクションで定義されます。
- 6 値の接頭辞は前述の設定を行います。
- 7 **cinder\_user\_enabled\_backends** クラスは、カスタムバックエンドを設定して有効にします。名前が示すように、このクラスはユーザー対応のバックエンドにのみ使用する必要があります。具体的には、**cinder::config::cinder\_config** クラスで定義されているものです。

**cinder\_user\_enabled\_backends** を使用して、Director を介してネイティブに有効にできるバックエンドをリストしないでください。サポートされている NetApp アプライアンスまたは Dell アプライアンスには、Red Hat Ceph、NFS、および単一のバックエンドが含まれます。たとえば、Red Hat Ceph バックエンドも有効にする場合は、それを **cinder\_user\_enabled\_backends** にリストしないでください。**CinderEnableRbdBackend: true** を使用して有効にします。



#### 注記

OpenStack Block Storage の Red Hat Ceph バックエンドの定義に関する詳細は、[コンテナ化された Red Hat Ceph を使用したオーバークラウドのデプロイ](#) を参照してください。

設定済みバックエンドのデプロイ では、環境ファイル **/home/stack/templates/custom-env.yaml** を使用してカスタムバックエンドのデプロイを調整する方法について説明しています。**/home/stack/templates/custom-env.yaml** から得られる **/etc/cinder/cinder.conf** 設定を確認するには、「[サンプル環境ファイルから得られる設定](#)」を参照してください。

## 第4章 設定済みのバックエンドのデプロイ

`/home/stack/templates/` に `custom-env.yaml` ファイルを作成したら、`stack` ユーザーとしてログインします。次に、次を実行してカスタムバックエンド設定をデプロイします。

```
$ openstack overcloud deploy --templates -e /home/stack/templates/custom-env.yaml
```



### 重要

オーバークラウドの作成時に追加の環境ファイルを渡した場合には、予定外の変更がオーバークラウドに加えられないように、ここで `-e` オプションを使用して環境ファイルを再度渡します。詳細は、`director` のインストールと使用方法ガイドの [オーバークラウド環境の変更](#) を参照してください。

`director` のオーケストレーションが完了したら、バックエンドをテストします。[5章 設定したバックエンドのテスト](#) を参照してください。

## 第5章 設定したバックエンドのテスト

バックエンドをオーバークラウドにデプロイしたら、それらにボリュームを正常に作成できるかどうかをテストします。最初に必要な環境変数をロードする必要があります。変数は、デフォルトで `/home/stack/overcloudrc` に定義されています。

1. 変数をロードするには、**stack** ユーザーとして次のコマンドを実行します。

```
$ source /home/stack/overcloudrc
```



### 注記

詳細は、**director** のインストールと使用方法の [オーバークラウドへのアクセス](#) を参照してください。

1. バックエンドごとに **ボリュームタイプ** を作成します。**stack** ユーザーとしてオーバークラウドのコントローラーノードにログインし、以下のコマンドを実行します。

```
$ cinder type-create backend1
$ cinder type-create backend2
```

これらのコマンドは、**envfile** の **cinder::config::cinder\_config** クラスで定義されたバックエンドごとに1つずつ、ボリュームタイプ **backend1** および **backend2** を作成します。

1. **envfile** の **cinder\_user\_enabled\_backends** クラスで有効化されたバックエンドの **volume\_backend\_name** に各ボリュームタイプをマッピングします。次のコマンドは、ボリュームタイプ **backend1** を **netapp1** に、**backend2** を **netapp2** にマッピングします。

```
$ cinder type-key backend1 set volume_backend_name=netapp1
$ cinder type-key backend2 set volume_backend_name=netapp2
```

1. これで、各バックエンドをテストできます。**backend1** ボリュームタイプを呼び出して、**netapp1** バックエンドに **netapp\_volume\_1** という名前の1GB ボリュームを作成します。

```
$ cinder create --volume-type backend1 --display_name netappvolume_1 1
```

1. **backend2** のボリューム種別を呼び出して、**netapp2** バックエンド上に同様のボリュームを作成します。

```
$ cinder create --volume-type backend2 --display_name netappvolume_2 1
```

## 付録A 付録

### A.1. STACK ユーザー

**stack** ユーザーアカウントを使用して、バックエンドのデプロイやオーバークラウドにアクセスするための環境変数の読み込みなど、昇格された権限を必要とするコマンドを実行できます。**stack** ユーザーの詳細については、**Director のインストールと使用** ガイドの [stack ユーザーの作成](#) を参照してください。

### A.2. サンプル環境ファイルから得られる設定

[3章環境ファイルの作成](#) の環境ファイルは、2つの NetApp バックエンドを使用するように Block Storage サービスを設定します。以下のスニペットには、関連する設定が表示されます。

```
enabled_backends = netapp1,netapp2

[netapp1]
volume_backend_name=netapp_1
volume_driver=cinder.volume.drivers.netapp.common.NetAppDriver
netapp_login=root
netapp_storage_protocol=iscsi
netapp_password=p@$$$w0rd
netapp_storage_family=ontap_7mode
netapp_server_port=80
netapp_server_hostname=10.35.64.11

[netapp2]
volume_backend_name=netapp_2
volume_driver=cinder.volume.drivers.netapp.common.NetAppDriver
netapp_login=root
netapp_storage_protocol=iscsi
netapp_password=p@$$$w0rd
netapp_storage_family=ontap_7mode
netapp_server_port=80
netapp_server_hostname=10.35.64.11
```