



Red Hat OpenStack Platform 16.0

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 16.0 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

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/Custom_Block_Storage_Back_End_Deployment_Guide.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

目次

第1章 BLOCK STORAGE サービスのカスタムバックエンドのデプロイの概要	3
1.1. カスタムバックエンドについて	3
1.2. 要件	4
1.3. 設定プロセスについて	4
第2章 環境ファイルの作成	5
第3章 設定済みのバックエンドのデプロイ	7
第4章 設定したバックエンドのテスト	8
付録A 付録	9

第1章 BLOCK STORAGE サービスのカスタムバックエンドのデプロイの概要

Red Hat OpenStack Platform director は、手動の設定を最小限に抑え、完全なエンタープライズレベルの OpenStack デプロイメントをインストールし、管理します。director の詳細は、『[director のインストールと使用方法](#)』を参照してください。

director が作成する Openstack 環境は、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。本項では、オーバークラウド上の Block Storage サービス(cinder)にカスタムのバックエンドをデプロイする方法について説明します。デフォルトでは、Block Storage サービスはコントローラーノードにインストールされます。

前提条件

- director を使用してオーバークラウドがインストール済みである。
- オーバークラウドに、機能する Block Storage サービスが設定されている。
- Block Storage の概念および設定を理解している。Block Storage の詳細は、『[Storage Guide](#)』の「[Block Storage and Volumes](#)」を参照してください。



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について

カスタムバックエンドは、Red Hat OpenStack Platform director に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、または設定です。サポート対象の Block Storage バックエンドは、すでにビルトインの director ファイルで統合および事前設定されています。たとえば、Red Hat Ceph および Dell EMC PS Series、Dell Storage Center、および NetApp アプライアンスの設定などです。

統合ストレージアプライアンスは、シングル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 向けに事前に設定された director ファイルでは、1つのバックエンドしかデプロイできません。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場合は、カスタム設定が必要です。

Block Storage サービスが配置されているノード上の `/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できますが、`openstack overcloud deploy` コマンドを実行すると、設定が director により上書きされます。詳細は、「[3章 設定済みのバックエンドのデプロイ](#)」を参照してください。director を使用して Block Storage バックエンドをデプロイし、オーバークラウドのデプロイメントと更新を行っても、設定が永続されるようになります。

バックエンド設定が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の場合は、独自の環境ファイルを作成する必要があります。本ガイドには、デプロイメント用に編集可能な `/home/stack/templates/custom-env.yaml` ファイルが含まれます。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。環境ファイルの詳細は、『[director のインストールと使用方法](#)』の「[オーバークラウドデプロイメントへの環境ファイルの追加](#)」を参照してください。

1.2. 要件

カスタムの Block Storage バックエンドを設定するには、以下の追加の前提条件を環境に適用する必要があります。

- 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には、それらをストレージリポジトリとして設定している。
- 『[director のインストールと使用方法](#)』に記載の手順に従って、director を使用してオーバークラウドをデプロイしている。
- 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。作成した同じ **stack** ユーザーアカウントを使用して、オーバークラウドをデプロイできる。
- `/etc/cinder/cinder.conf` に Block Storage バックエンドに指定する設定がすでに計画されている。

1.3. 設定プロセスについて

カスタム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には、以下の手順が必要です。

- 環境ファイルの作成。詳細は、「[2章 環境ファイルの作成](#)」を参照してください。
- 設定済みのバックエンドのデプロイ。詳細は、「[3章 設定済みのバックエンドのデプロイ](#)」を参照してください。
- 設定したバックエンドのテスト。詳細はを参照してください。[4章 設定したバックエンドのテスト](#)

第2章 環境ファイルの作成

カスタムバックエンド設定用に作成する環境ファイルには、定義する各バックエンドの設定が含まれます。また、カスタムバックエンドのデプロイメントに関連するその他の設定も含まれます。環境ファイルに関する詳しい情報は、『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。https://access.redhat.com/documentation/ja-jp/red_hat_openshift_platform/16.0/html-single/advanced_overcloud_customization/#sect-Environment_Files

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 以下のパラメーターは **false** に設定され、他のバックエンド種別が無効になります。
 - **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** パラメーターで、Image サービスがイメージ保管に使用するものを設定します。以下の値がサポートされています。
 - **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、全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。
- 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** プレフィックス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 **netapp2** 設定は、別の **[netapp2]** セクションで定義します。
- 6 値のプレフィックス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。このクラスは、ユーザー対応バックエンド（具体的には **cinder::config::cinder_config** クラスで定義されるもの）にのみ使用します。

cinder_user_enabled_backends を使用して、director でネイティブに有効化できるバックエンドを一覧表示しないでください。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 バックエンドを有効にする場合には、**cinder_user_enabled_backends** に一覧を表示しないでください。**CinderEnableRbdBackend** を **true** に設定して有効にします。



注記

OpenStack Block Storage 用の Red Hat Ceph バックエンドの定義の詳細は、『[コンテナ化された Red Hat Ceph を持つオーバークラウドのデプロイ](#)』を参照してください。

/home/stack/templates/custom-env.yaml から作成された **/etc/cinder/cinder.conf** の設定を確認するには、**???**を参照してください。

第3章 設定済みのバックエンドのデプロイ

設定したバックエンドをデプロイするには、以下の手順を実施します。

1. **stack** ユーザとしてログインします。
2. 以下のコマンドを実行して、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。詳しい情報は、『[directorのインストールと使用方法](#)』の「オーバークラウド環境の変更」を参照してください。

第4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後に、そこにボリュームを正常に作成できることをテストします。

手順

1. **stack** ユーザーとして以下のコマンドを実行し、**home/stack/overcloudrc** で定義した環境変数を読み込みます。

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

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種別を作成します。**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
```

これらのコマンドにより、作成した環境ファイルの **cinder::config::cinder_config** クラスで定義されるバックエンドごとに、ボリューム種別の **backend1** と **backend2** が作成されます。

2. 作成した環境ファイルの **cinder_user_enabled_backends** クラスで有効化されたバックエンドの **volume_backend_name** に各ボリューム種別をマップします。以下のコマンドは、ボリューム種別の **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
```

3. 以下のコマンドを実行して、**backend1** のボリューム種別を呼び出して、**netapp1** にバックエンドを作成できることを確認します。

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

4. **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

サンプル環境ファイルの設定

2章 [環境ファイルの作成](#) で作成した環境ファイルは、Block Storage サービスが2つのNetAppバックエンドを使用するように設定します。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
```