



# Red Hat OpenStack Platform

## 11

# Fujitsu ETERNUS バックエンドガイド

---

Red Hat OpenStack Platform 11 環境で Fujitsu ETERNUS バックエンドを使用するためのガイド

OpenStack Team



## Red Hat OpenStack Platform 11 環境で Fujitsu ETERNUS バックエンドを使用するためのガイド

OpenStack Team  
rhos-docs@redhat.com

## 法律上の通知

Copyright © 2017 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, 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 Software Collections 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

本ガイドは、Red Hat OpenStack Platform 11 で Fujitsu ETERNUS ディスクストレージシステムをバックエンドとして使用するための設定方法について説明します。

目次

1. はじめに ..... 2

2. FUJITSU ETERNUS デバイスの設定 ..... 2

3. HEAT テンプレートの準備 ..... 3

    3.1. 各バックエンドのドライバー定義の作成 ..... 4

    3.2. 完全な Heat テンプレート例 ..... 6

4. 環境ファイルの作成 ..... 7

5. 設定したバックエンドのデプロイ ..... 10

6. 設定のテスト ..... 10

## 1. はじめに

本ガイドでは、OpenStack で Fujitsu ETERNUS ディスクストレージシステムを Block Storage サービスのバックエンドとして使用するための設定方法について説明します。具体的には、オーバークラウドのデプロイメント上の ETERNUS デバイスによって提供されるファイバーチャネルおよび iSCSI バックエンドを定義する方法を記載します。これには、Block Storage サービスの **カスタムバックエンド** として両バックエンドを定義する必要があります。

以下のセクションは、下記を前提としています。

- ※ Red Hat OpenStack Platform オーバークラウドがすでに **director** を使用してデプロイ済みであること。
- ※ Fujitsu ETERNUS ディスクストレージシステムのデバイスおよびドライバーのみを Block Storage バックエンドに使用する予定であること。
- ※ オーバークラウドのデプロイメントの一部として作成される **director** のインストールユーザーを使用すること。詳しい情報は、『[Director のインストールと使用方法](#)』の「[director のインストールユーザーの作成](#)」を参照してください。
- ※ ETERNUS デバイスの **Admin** アカウントへのアクセス権があること。つまり、ETERNUS の Web GUI または CLI から **Admin** ロールのあるアカウントを使用してログインできること。
- ※ Block Storage は、コントローラーノードにインストールされること (デフォルト)。

Fujitsu ETERNUS では、ファイバーチャネルまたは iSCSI インターフェースのいずれかを使用することができます。各インターフェースには、独自の設定とドライバーがあります。Red Hat は、OpenStack で両方のインターフェース (およびそれぞれのドライバー) の使用をサポートしています。



### 注記

カスタムバックエンドの定義方法に関連する情報は『[カスタムの Block Storage バックエンドのデプロイメントガイド](#)』を参照してください。

## 2. FUJITSU ETERNUS デバイスの設定

先に Fujitsu ETERNUS デバイス上のストレージプールとポートを設定をしてからでないと、Fujitsu ETERNUS デバイスをブロックデバイスとして定義できません。それぞれの手順に関する情報は、デバイスのドキュメントを参照してください。

1. Block Storage サービスがホストされているコントローラーノードと ETERNUS デバイスの MNT ポートとの間の LAN 接続を設定します。
2. コンピュートノードと ETERNUS デバイスの CA ポート間の SAN 接続を設定します。
3. **Admin** ロールの付いたアカウントを使用して、ETERNUS デバイスにログインします。
4. ETERNUS DX の SMI-S を有効化します。
5. **Advanced Copy Feature** ライセンスを登録して、コピーテーブルサイズを設定します。
6. ボリュームのストレージプールを作成します。このプールは、後ほど「[各バックエンドのドライバー定義の作成](#)」で **EternusPool** の設定を行う際に使用します。



## 注記

異なるストレージプールでボリュームのスナップショットを作成する場合は、そのストレージプールも作成します。このプールは、「[各バックエンドのドライバー定義の作成](#)」で **EternusSnapPool** の設定を行う際に使用します。

7. スナップショットの作成 機能の Snap Data Pool (SDP) を有効化するには、**Snap Data Pool Volume (SDPV)** を作成します。
8. Block Storage サービスに使用する **ストレージポート** を設定してから、以下を実行します。
  - a. これらのポートを **CA** モードに設定します。
  - b. これらのストレージポートの **ホストアフィニティ** 設定を有効化します。ホストアフィニティを有効化するには、ETERNUS CLI から各ポートに対して以下のコマンドを実行します。

```
CLI> set PROTO-parameters -host-affinity enable -port CM# CA#
PORT
```

\* **PROTO** は、使用中のストレージプロトコルを **fc** (ファイバーチャネル) または **iscsi** のいずれかに設定します。**CM# CA#** は、ポートがあるコントローラーエンクロージャーに、**PORT** はポート番号に置き換えてください。

## 3. HEAT テンプレートの準備

Red Hat OpenStack Platform オーバークラウドのデプロイ時には、**director** を使用してサービス設定すべてを実行することを推奨します。**director** で設定すると、今後オーバークラウドの更新の際に設定の一貫性を保たれるようになります。

ETERNUS バックエンドでは、(Block Storage サービスがホストされる) コントローラーノードで以下が要件となっています。

- ✦ **pywbem** パッケージがインストールされていること
- ✦ 各バックエンドのドライバー設定に対する XML 設定ファイル

いずれのタスクも **heat** テンプレートを使用して、**director** でオーケストレーションすることができます。**director** 用に使用する **heat** テンプレートの構文に関する情報は、「[Heat テンプレートの理解](#)」を参照してください。

以下のテンプレート (**eternus-temp.yaml**) には、**heat** テンプレートに必要な基本構文が含まれています。また、必須の **pywbem** パッケージのインストール方法も含まれています。

### eternus-temp.yaml

```
heat_template_version: 2014-10-16

description: >
  Installs the pywbem package on all Controller nodes

parameters:
  server:
```

```

type: string

resources:
  EternusSetup: # 1
    type: OS::Heat::SoftwareConfig
    properties:
      group: script
      config: | # 2
        #!/bin/bash
        sudo yum install pywbem -y
        # 3

    ExtraPreDeployment:
      type: OS::Heat::SoftwareDeployment
      properties:
        config: {get_resource: EternusSetup}
        server: {get_param: server}
        actions: [CREATE, UPDATE]

```

## 1

**EternusSetup** では、コントローラーノードで実行する必要があるタスクをオーケストレーションするためのリソースを定義します。

## 2

**config** セクションには、コントローラーノードで実行する必要がある全コマンドを挿入します。現在、**pywbem** のインストールコマンドのみが記載されています。

## 3

「[各バックエンドのドライバー定義の作成](#)」の各バックエンドのドライバー設定用に XML 設定ファイルを作成するためのコマンドは、ここに追加します。

director ノードの **/home/stack/templates/** にこのファイルを保存します。一般的な Red Hat OpenStack Platform director デプロイメントでは、カスタムの heat テンプレートはここに保存されます。

### 3.1. 各バックエンドのドライバー定義の作成

各 ETERNUS バックエンドのドライバー設定は、Block Storage 設定ファイル (**/etc/cinder/cinder.conf**) ではなく別の XML ファイルで定義されます。バックエンドごとに、独自の XML ファイルが必要です。各 XML ファイルには、以下の設定を含める必要があります。

#### EternusIP

ETERNUS デバイスの SMI-S 接続に使用する IP アドレス。具体的にはデバイスの MNT ポートの IP アドレスのことです。

#### EternusPort

ETERNUS デバイスの SMI-S 接続ポートのポート番号



**EternusUser**

SMI-S 接続 (**EternusIP**) に使用するユーザー名

**EternusPassword**

**EternusIP** での **EternusUser** に対応するパスワード

**EternusPool**

「[Fujitsu ETERNUS デバイスの設定](#)」でボリューム用に作成したストレージプール名。具体的には、ETERUNUS デバイスのプールの RAID グループ名または TPP 名を使用します。

**EternusSnapPool**

「[Fujitsu ETERNUS デバイスの設定](#)」でボリュームのスナップショット用に作成したストレージプール名。具体的には、ETERUNUS デバイスのプールの RAID グループ名または TPP 名を使用します。スナップショット用に別のプールを作成しなかった場合は、**EternusPool** と同じ値を使用します。

**EternusISCSIIP**

**(iSCSI のみ)**: ETERNUS デバイスへの iSCSI 接続用の IP アドレス。それぞれにエントリーを作成することで、複数の IP を指定できます。

たとえば、ファイバーチャネルの設定を定義する場合は以下のようになります。

**eternus-fc.xml**

```
<?xml version='1.0' encoding='UTF-8'?>
<FUJITSU>
<EternusIP>0.0.0.0</EternusIP>
<EternusPort>5988</EternusPort>
<EternusUser>smisuser</EternusUser>
<EternusPassword>smispassword</EternusPassword>
<EternusPool>raid5_0001</EternusPool>
<EternusSnapPool>raid5_0001</EternusSnapPool>
</FUJITSU>
```

iSCSI の 4 つの接続に関する iSCSI の設定:

**eternus-iscsi.xml**

```
<?xml version='1.0' encoding='UTF-8'?>
<FUJITSU>
<EternusIP>0.0.0.0</EternusIP>
<EternusPort>5988</EternusPort>
<EternusUser>smisuser</EternusUser>
<EternusPassword>smispassword</EternusPassword>
<EternusPool>raid5_0001</EternusPool>
<EternusSnapPool>raid5_0001</EternusSnapPool>
<EternusISCSIIP>1.1.1.1</EternusISCSIIP>
<EternusISCSIIP>1.1.1.2</EternusISCSIIP>
<EternusISCSIIP>1.1.1.3</EternusISCSIIP>
<EternusISCSIIP>1.1.1.4</EternusISCSIIP>
</FUJITSU>
```

これらの XML ファイルの作成をオーケストレーションするには、bash コマンドを使用します。これらのコマンドは、`/home/stack/templates/eternus-temp.yaml` の **EternusSetup** リソースの **config** セクションに追加する必要があります (「[Heat テンプレートの準備](#)」参照)。たとえば、`eternus-fc.xml` および `eternus-iscsi.xml` の作成をオーケストレーションするには、以下を実行します。

```
sudo cat > /etc/cinder/eternus-fc.xml <<EOF
<?xml version=1.0 encoding=UTF-8?>
<FUJITSU>
<EternusIP>0.0.0.0</EternusIP>
<EternusPort>5988</EternusPort>
<EternusUser>smisuser</EternusUser>
<EternusPassword>smispassword</EternusPassword>
<EternusPool>raid5_0001</EternusPool>
<EternusSnapPool>raid5_0001</EternusSnapPool>
</FUJITSU>
EOF

sudo cat > /etc/cinder/eternus-iscsi.xml <<EOF
<?xml version=1.0 encoding=UTF-8?>
<FUJITSU>
<EternusIP>0.0.0.0</EternusIP>
<EternusPort>5988</EternusPort>
<EternusUser>smisuser</EternusUser>
<EternusPassword>smispassword</EternusPassword>
<EternusPool>raid5_0001</EternusPool>
<EternusSnapPool>raid5_0001</EternusSnapPool>
<EternusISCSIIP>1.1.1.1</EternusISCSIIP>
<EternusISCSIIP>1.1.1.2</EternusISCSIIP>
<EternusISCSIIP>1.1.1.3</EternusISCSIIP>
<EternusISCSIIP>1.1.1.4</EternusISCSIIP>
</FUJITSU>
EOF
```

XML 設定ファイルを必要な数だけ作成するようにオーケストレーションするには、作成するファイルに合わせて **sudo cat** を使用します。

XML ファイルは、**cinder** ユーザーとグループが所有し、所有者のみが利用できるようにする必要があります。そのため、これらのファイルの所有者およびパーミッションもオーケストレーションする必要があります。

```
sudo chown cinder:cinder /etc/cinder/eternus-*.xml
sudo chmod 0600 /etc/cinder/eternus-*.xml
```

完全な heat テンプレート例は「[完全な Heat テンプレート例](#)」を参照してください。

## 3.2. 完全な Heat テンプレート例

以下の `/home/stack/templates/eternus-temp.yaml` ファイルには、必要な **pywbem** パッケージのインストールおよび「[各バックエンドのドライバ定義の作成](#)」のサンプル XML 設定ファイル (`eternus-fc.xml` および `eternus-iscsi.xml`) の宣言に必要なコンポーネントがすべて含まれています。

`/home/stack/templates/eternus-temp.yaml`

```
heat_template_version: 2014-10-16
```

```

description: >
    Installs the pywbem package on all Controller nodes

parameters:
  server:
    type: string

resources:
  EternusSetup:
    type: OS::Heat::SoftwareConfig
    properties:
      group: script
      config: |
        #!/bin/bash
        sudo yum install pywbem -y
        sudo cat > /etc/cinder/eternus-fc.xml <<EOF
        <?xml version=1.0 encoding=UTF-8?>
        <FUJITSU>
        <EternusIP>0.0.0.0</EternusIP>
        <EternusPort>5988</EternusPort>
        <EternusUser>smisuser</EternusUser>
        <EternusPassword>smispassword</EternusPassword>
        <EternusPool>raid5_0001</EternusPool>
        <EternusSnapPool>raid5_0001</EternusSnapPool>
        </FUJITSU>
        EOF
        sudo cat > /etc/cinder/eternus-iscsi.xml <<EOF
        <?xml version=1.0 encoding=UTF-8?>
        <FUJITSU>
        <EternusIP>0.0.0.0</EternusIP>
        <EternusPort>5988</EternusPort>
        <EternusUser>smisuser</EternusUser>
        <EternusPassword>smispassword</EternusPassword>
        <EternusPool>raid5_0001</EternusPool>
        <EternusSnapPool>raid5_0001</EternusSnapPool>
        <EternusISCSIIP>1.1.1.1</EternusISCSIIP>
        <EternusISCSIIP>1.1.1.2</EternusISCSIIP>
        <EternusISCSIIP>1.1.1.3</EternusISCSIIP>
        <EternusISCSIIP>1.1.1.4</EternusISCSIIP>
        </FUJITSU>
        EOF
        sudo chown cinder:cinder /etc/cinder/eternus-*.xml
        sudo chmod 0600 /etc/cinder/eternus-*.xml

  ExtraPreDeployment:
    type: OS::Heat::SoftwareDeployment
    properties:
      config: {get_resource: EternusSetup}
      server: {get_param: server}
      actions: [CREATE, UPDATE]

```

## 4. 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が記載されます。また、カスタムのバックエンドのデプロイメントに関連した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、『Advanced Overcloud Customization』の「[Environment Files](#)」のセクションを参照してください。

さらに、環境ファイルは、以前に「[Heat テンプレートの準備](#)」で作成した heat テンプレートも登録します。これにより、テンプレートで定義したインストールコマンドおよび echo コマンドが適切なノードで実行されます。

以下の環境ファイルの例には、ETERNUS デバイスを Block Storage バックエンドとして定義するのに必要なセクションがすべて含まれています。「[各バックエンドのドライバー定義の作成](#)」および「[完全な Heat テンプレート例](#)」でオーケストレーションした、各 XML ファイルに対するバックエンドの定義も作成します。

### eternusbackend-env.yaml

```
parameters: # 1
  CinderEnableIscsiBackend: false
  CinderEnableRbdBackend: false
  CinderEnableNfsBackend: false
  NovaEnableRbdBackend: false
  GlanceBackend: file # 2

resource_registry:
  OS::TripleO::NodeExtraConfig: /home/stack/templates/eternus-temp.yaml # 3

parameter_defaults:
  controllerExtraConfig: # 4
    cinder::config::cinder_config:
      FJFC/volume_driver: # 5
        value: cinder.volume.drivers.fujitsu.eternus_dx_fc.FJDXFCDriver
      FJFC/cinder_eternus_config_file: # 6
        value: /etc/cinder/eternus-fc.xml
      FJFC/volume_backend_name: # 7
        value: FJFC
      FJISCSI/volume_driver: # 8
        value:
cinder.volume.drivers.fujitsu.eternus_dx_iscsi.FJDXISCSIDriver
      FJISCSI/cinder_eternus_config_file:
        value: /etc/cinder/eternus-iscsi.xml
      FJISCSI/volume_backend_name:
        value: FJISCSI
    cinder_user_enabled_backends: [FJFC, FJISCSI] # 9
```

## 1

以下のパラメーターは、**false** に設定されているので、必要のない他のバックエンド種別は無効になります。

- ✳ **CinderEnableIscsiBackend**: その他の iSCSI バックエンド
- ✳ **CinderEnableRbdBackend**: Red Hat Ceph

- ※ **CinderEnableNfsBackend**: NFS
- ※ **NovaEnableRbdBackend**: Red Hat Ceph の一時ストレージ

## 2

**GlanceBackend** パラメーターは、Image サービスがイメージの保管に使用すべきバックエンドを設定します。以下の値がサポートされています。

- ※ **file**: 各コントローラーノード上の **/var/lib/glance/images** にイメージを保管します。
- ※ **swift**: イメージの保管に Object Storage サービスを使用します。
- ※ **cinder**: イメージの保存に Block Storage サービスを使用します。

## 3

**NodeExtraConfig** は、コアの Puppet 設定を行う **前に**、すべてのノードに適用するカスタムを設定を定義します。これにより、Block Storage サービスがオーバークラウドにデプロイされるまでに、以下が確実に行われるようにします。

- ※ コントローラーノードにすでに **pywbem** パッケージがインストールされていること
- ※ 各バックエンドの XML 設定ファイルがすでに作成されていること

## 4

**controllerExtraConfig** は、コントローラーノードすべてに適用されるカスタムを設定を定義します。**cinder::config::cinder\_config** クラスは、Block Storage (**cinder**) サービスに適用すべき設定を意味します。同様に、バックエンドの設定は最終的に、各ノードの **/etc/cinder/cinder.conf** ファイルに指定されることになります。

## 5

**FJFC/** 文字列は **FJFC** という名前のバックエンド定義を作成し、このバックエンド定義で以下のパラメーターが宣言されます。**volume\_driver** パラメーターは、バックエンドに対して固有の ETERNUS ドライバーを設定します。今回の例では、**cinder.volume.drivers.fujitsu.eternus\_dx\_fc.FJDXFCDriver** はファイバーチャネルドライバを設定します。

## 6

**cinder\_eternus\_config\_file** は、バックエンドにドライバーが使用すべき XML 設定ファイルへのパスを設定します。**/etc/cinder/eternus-fc.xml** の作成は、**/home/stack/templates/eternus-temp.yaml** という heat テンプレートでオーケストレーションされます。

## 7

**volume\_backend\_name** は、バックエンドを有効化するのに使用すべき Block Storage サービスの名前です。

**FJISCSI** 文字列は、前述の **FJFC** と同様に、新しいバックエンドの定義を作成します。

**cinder\_user\_enabled\_backends** クラスは、カスタムのバックエンドを設定して有効にします。各前が示すように、このクラスはユーザーが有効化したバックエンドにのみ使用する必要があります。対象となるバックエンドは、具体的には **cinder::config::cinder\_config** クラスで定義されています。

環境ファイルの作成後には、設定をデプロイすることができます。この目的で、**/home/stack/templates/eternusbackend-env.yaml** の環境ファイルを使用する方法についての情報は、「[設定したバックエンドのデプロイ](#)」を参照してください。

## 5. 設定した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、以下のコマンドを実行してバックエンド設定をデプロイします。

```
$ openstack overcloud deploy --templates -e
/home/stack/templates/eternusbackend-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。

**director** がオーケストレーションを完了したら、バックエンドをテストします。手順は、「[設定のテスト](#)」を参照してください。

## 6. 設定のテスト

Block Storage サービスが新しい ETERNUS バックエンドを使用するように設定した後は、それぞれの **ボリューム種別** を宣言します。ボリューム種別の設定により、新規ボリュームの作成時に使用するバックエンドを指定することができます。以下のコマンドを実行すると、**FJFC** (ファイバーチャネルバックエンド) と **FJISCSI** (iSCSI バックエンド) の 2 つのボリュームが作成されます。

```
# cinder type-create FJFC
# cinder type-create FJISCSI
```

次に、ボリューム種別をそれぞれのバックエンドにマッピングします（「[環境ファイルの作成](#)」の定義に従います）。

```
# cinder type-key FJFC set volume_backend_name=FJFC
# cinder type-key FJISCSI set volume_backend_name=FJISCSI
```

**test\_iscsi** という名前の 1 GB の iSCSI ボリュームを作成して、設定を検証します。

```
# cinder create --volume_type FJISCSI --display_name test_iscsi 1
```

ファイバーチャネルバックエンドをテストするには、以下のコマンドを実行します。

```
# cinder create --volume_type FJFC --display_name test_fc 1
```