



Red Hat OpenStack Platform 15

Fujitsu ETERNUS バックエンドガイド

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

Red Hat OpenStack Platform 15 Fujitsu ETERNUS バックエンドガイド

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

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/Fujitsu_ETERNUS_Back_End_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

本ドキュメントでは、Fujitsu ETERNUS Disk Storage System をバックエンドとして使用するよう Red Hat OpenStack Platform 15 を設定する方法について説明します。

目次

第1章 はじめに	3
第2章 FUJITSU ETERNUS デバイスの設定	4
第3章 FUJITSU ETERNUS HEAT テンプレートの準備	6
3.1. FUJITSU ETERNUS バックエンドごとのドライバー定義の作成	7
3.2. FUJITSU ETERNUS HEAT テンプレートの例	8
第4章 FUJITSU ETERNUS 環境ファイルの作成	10
第5章 設定された FUJITSU ETERNUS バックエンドのデプロイ	13
第6章 FUJITSU ETERNUS 設定のテスト	14

第1章 はじめに

本書では、Block Storage サービスのバックエンドとして Fujitsu ETERNUS Disk Storage System を使用するように Red Hat OpenStack Platform を設定する方法を説明します。本書では、オーバークラウドのデプロイメントで ETERNUS デバイスが提供する Fibre Channel および iSCSI バックエンドを定義する方法について説明します。このプロセスでは、両方のバックエンドを Block Storage サービスの **custom back end** として定義します。デフォルトでは、コントローラーノードには Block Storage サービスが含まれます。

前提条件

- Block Storage バックエンドには、Fujitsu ETERNUS Disk Storage System デバイスおよびドライバーのみを使用する。
- オーバークラウドデプロイメントで作成した **director** インストールユーザーを使用できる。詳しい情報は、『[director のインストールと使用方法](#)』の「[スタックユーザーの作成](#)」を参照してください。
- ETERNUS Web GUI または CLI から ETERNUS デバイスの **Admin** アカウントにアクセスできる。

Red Hat は、Fujitsu ETERNUS デバイスでの Fibre Channel または iSCSI インターフェースの使用と、それぞれのドライバーおよび設定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に関する詳しい情報は、『[カスタム Block Storage バックエンドデプロイメントガイド](#)』を参照してください。

第2章 FUJITSU ETERNUS デバイスの設定

Fujitsu ETERNUS デバイスを Block Storage バックエンドとして定義する前に、デバイスにストレージプールおよびポートを設定します。各手順の詳細については、デバイスのドキュメントを参照してください。

手順

1. Block Storage サービスをホストするコントローラーノードと ETERNUS デバイスの MNT ポート間の LAN 接続を設定します。
2. コンピュータノードと ETERNUS デバイスの CA ポートとの間の SAN 接続を設定します。
3. **Admin** ロールのアカウトを使用して ETERNUS デバイスにログインします。
4. ETERNUS DX の SMI-S を有効にします。
5. SSH キーを設定します。
6. **software** ロールのユーザーアカウントを追加します。
7. **Advanced Copy Feature** ライセンスを登録し、コピーテーブルのサイズを設定します。
8. ボリュームのストレージプールを作成します。このプールは後で、「[Fujitsu ETERNUS バックエンドごとのドライバー定義の作成](#)」の **EternusPool** 設定で使します。
9. オプション：ボリュームのスナップショット用に別のストレージプールを作成します。このプールは、「[Fujitsu ETERNUS バックエンドごとのドライバー定義の作成](#)」の **EternusSnapPool** 設定を表します。
10. オプション：スナップショット機能用に **Snap Data Pool(SDP)**を有効にするために **Snap Data Pool** ボリューム(**SDPV**)を作成 します。

ヒント

EternusSnapPool にシンプロビジョニングプールを使用する場合は、この手順を省略します。

11. Block Storage サービスが使用する **storage ports** を設定します。
12. **storage ports** ポートを **CA** モードに設定します。
13. ストレージポートの **host-affinity** を有効にするには、各ポートに対して ETERNUS CLI から次のコマンドを入力します。

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

- <PROTO> を、**fc** や **iscsi** などのストレージプロトコルに置き換えます。
- <CM#> および <CA#> を、ポートが置かれているコントローラーエンクロージャーの名前に置き換えます。
- <PORT> をポート番号に置き換えます。

14. 以下のコマンドを使用してアンダークラウドで SSH キーを生成し、**eternus.ietf** ファイルを ETERNUS デバイスにアップロードします。

■


```
ssh-keygen -t rsa -N "" -f ./eternus  
ssh-keygen -e -f ./eternus.pub > ./eternus.ietf
```



注記

後で使用できるように **eternus** ファイルを保存します。セキュリティ上の理由から、アンダークラウドまたはオーバークラウドからコピーしたり、移動したりしないでください。

第3章 FUJITSU ETERNUS HEAT テンプレートの準備

Red Hat Open Stack Platform オーバークラウドへの今後の更新を通じて設定が維持されるようにするには、director を介したデプロイメント中にすべてのサービス設定を実行します。

Block Storage サービスをホストする ETERNUS バックエンドのコントローラーノードに、以下の設定を追加します。

- 各バックエンドのドライバー設定に XML 設定ファイルを含めます。
- ETERNUS デバイスと通信するための SSH サーバーキーを含めます。

heat テンプレートを使用して、director で両方のタスクをオーケストレーションすることができます。director heat テンプレートの構文の詳細については、[オーバークラウドの詳細ドガイドの Heat テンプレートの概要](#) を参照してください。

以下のテンプレート **eternus-temp.yaml** は、必要な heat テンプレートの基本的な構文を示しています。

eternus-temp.yaml

```
heat_template_version: 2014-10-16

description: >
  Add XML configuration file for the driver settings of each back end

parameters:
  server:
    type: string

resources:
  EternusSetup: # 1
    type: OS::Heat::SoftwareConfig
    properties:
      group: script
      config: | # 2
        #!/bin/bash
        # 3

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

- 1 **EternusSetup** セクションには、コントローラーノードのタスクをオーケストレーションするリソースが含まれます。
- 2 **config** セクションには、コントローラーノードで実行するコマンドが含まれます。
- 3 Block Storage サービスがホストされる各コントローラーノードに秘密鍵の情報をコピーし、「[Fujitsu ETERNUS バックエンドごと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コマンドを追加します。

このファイルを director ノードのカスタム heat テンプレートディレクトリー `/home/stack/templates/` に保存します。

3.1. FUJITSU ETERNUS バックエンドごとのドライバー定義の作成

Block Storage 設定ファイル `/etc/cinder/cinder.conf` ではなく、別の XML ファイルで各 ETERNUS バックエンド用のドライバー設定を定義します。各バックエンドに、以下の設定が含まれる XML ファイルがあることを確認します。

EternusIP

ETERNUS デバイスの SMI-S 接続の IP アドレス。デバイスの MNT ポートの IP アドレスを使用します。

EternusPort

ETERNUS デバイスの SMI-S 接続ポートのポート番号。

EternusUser

接続 **EternusIP** の **software** ロールのユーザー名。

EternusPassword

EternusIP 上の **EternusUser** の対応するパスワード。

EternusPool

[2章 Fujitsu ETERNUS デバイスの設定](#) からのボリュームのストレージプールの名前。ETERNUS デバイスのプール RAID Group 名または TPP 名を使用します。

EternusSnapPool

[2章 Fujitsu ETERNUS デバイスの設定](#) からのボリュームスナップショットのストレージプールの名前。ETERNUS デバイスのプール RAID Group 名を使用します。スナップショットに別のプールを作成していない場合は、**EternusPool** と同じ値を使用します。

以下の xml の例を使用して Fibre Channel 設定を定義します。

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>
```

以下の xml の例で ISCSI 設定を定義します。

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>
</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、[3章 Fujitsu ETERNUS heat テンプレートの準備](#)の `/home/stack/templates/eternus-temp.yaml` ファイルの **EternusSetup** リソースの **config** セクションに `bash` コマンドを含めます。以下のコマンド例を使用して、**eternus-fc.xml** および **eternus-iscsi.xml** の作成をオーケストレーションします。

```
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>
</FUJITSU>
EOF
```

sudo cat コマンドを使用して、必要な数の XML 設定ファイルを作成します。

cinder ユーザーおよびグループに対するこれらの XML ファイルの所有権および権限を設定します。

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

完了した heat テンプレートの例については、「[Fujitsu ETERNUS heat テンプレートの例](#)」を参照してください。

3.2. FUJITSU ETERNUS HEAT テンプレートの例

以下の `/home/stack/templates/eternus-temp.yaml` ファイルには、XML 設定ファイルのサンプルを宣言するために必要なパラメーター (**eternus-fc.xml** や **eternus-iscsi.xml** など) が含まれます。

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

```
heat_template_version: 2014-10-16

description: >
  Add XML configuration file for the driver settings of each back end
```

```

parameters:
  server:
    type: string

resources:
  EternusSetup:
    type: OS::Heat::SoftwareConfig
    properties:
      group: script
      config: |
        #!/bin/bash
        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>
        </FUJITSU>
        EOF
        sudo cat > /etc/cinder/eternus <<EOF
        -----BEGIN RSA PRIVATE KEY----- # 1
        MIIEpAIBAACAQEA5yMqonpfniu+11PJ8qdWZpcf0d4UcHj2uyE7ou7vcZUQ1Cq
        s5Q5pjkCgYAXITlpfOYA8jvLgc7vMEa/ZbhUgAPIYlisxbffmRsBWYJSt9gwHpcW
        hvaWo6VD/iUKZ3bOcmK0buUwBdFUt5s9B8mXbYsX6bWovlVkyu8DzQfpDiPnV6C8
        ...
        IB+46ldmCUO0DaciuEz5/KQd4AXBNdTOss2od6OzihDJXKjBwPyP1g==
        -----END RSA PRIVATE KEY-----
        EOF

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

```

1 2章 [Fujitsu ETERNUS デバイスの設定](#) にてアンダークラウドで生成された **eternus** ファイルの秘密鍵情報。

第4章 FUJITSU ETERNUS 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、およびその他の関連する設定が含まれます。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_openstack_platform/15/html-single/advanced_overcloud_customization/#sect-Environment_Files

また、環境ファイルは、[3章 Fujitsu ETERNUS heat テンプレートの準備](#) で作成した heat テンプレートに登録します。heat テンプレートで定義した installation コマンドと echo コマンドは、デプロイメント中に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、Block Storage バックエンドとして ETERNUS デバイスを定義するのに必要なセクションが含まれています。また、「[Fujitsu ETERNUS バックエンドごとのドライバー定義の作成](#)」および「[Fujitsu ETERNUS heat テンプレートの例](#)」でオーケストレーションされる対応する各 XML ファイルのバックエンド定義も作成します。

eternusbackend-env.yaml

```
resource_registry:
  OS::TripleO::NodeExtraConfig: /home/stack/templates/eternus-temp.yaml # 1

parameter_defaults: # 2
  CinderEnableScsiBackend: false
  CinderEnableRbdBackend: false
  CinderEnableNfsBackend: false
  NovaEnableRbdBackend: false
  GlanceBackend: file # 3
  controllerExtraConfig: # 4
    cinder::config::cinder_config:
      FJFC/volume_driver: # 5
        value: cinder.volume.drivers.fujitsu.eternus_dx.eternus_dx_fc.FJDXFCDriver
      FJFC/cinder_eternus_config_file: # 6
        value: /etc/cinder/eternus-fc.xml
      FJFC/volume_backend_name: # 7
        value: FJFC
      FJFC/fujitsu_private_key_path:
        value: /etc/cinder/eternus
      FJISCSI/volume_driver: # 8
        value: cinder.volume.drivers.fujitsu.eternus_dx.eternus_dx_iscsi.FJDXISCSIDriver
      FJISCSI/cinder_eternus_config_file:
        value: /etc/cinder/eternus-iscsi.xml
      FJISCSI/volume_backend_name:
        value: FJISCSI
      FJISCSI/fujitsu_private_key_path:
        value: /etc/cinder/eternus
    cinder_user_enabled_backends: ['FJFC','FJISCSI'] # 9
  CinderVolumeOptVolumes: # 10
    - /etc/cinder/eternus-iscsi.xml:/etc/cinder/eternus-iscsi.xml:ro
    - /etc/cinder/eternus-fc.xml:/etc/cinder/eternus-fc.xml:ro
    - /etc/cinder/eternus:/etc/cinder/eternus:ro
  DockerCinderVolumeImage: registry.connect.redhat.com/fujitsu/rhosp15-fujitsu-cinder-volume-161
  ContainerImageRegistryLogin: True
  ContainerImageRegistryCredentials:
```

```
registry.connect.redhat.com:
  my-username: my-password
registry.redhat.io:
  my-username: my-password
```

- 1 **Node Extra Config** を使用して、コア Puppet 設定の前にすべてのノードのカスタム設定を定義します。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際に、以下の設定が可能になります。
 - 各バックエンドの XML 設定ファイルが存在する。
 - 秘密鍵が生成される。
- 2 他のバックエンド種別を無効にするには、以下のパラメーターを **false** に設定します。
 - **CinderEnableIscsiBackend**: 他の iSCSI バックエンド。
 - **CinderEnableRbdBackend**: Red Hat Ceph Storage.
 - **CinderEnableNfsBackend**: NFS
 - **NovaEnableRbdBackend**: ephemeral Red Hat Ceph Storage.
- 3 **GlanceBackend** パラメーターを使用して、Image サービスのイメージストレージ設定を定義します。以下の値がサポートされています。
 - このファイルは、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** は、イメージストレージに Object Storage サービスを使用します。
 - **cinder** は、イメージストレージに Block Storage サービスを使用します。
- 4 **controllerExtraConfig** を使用して、全コントローラーノードのカスタム設定を定義します。**cinder::config::cinder_config** クラスは Block Storage サービス用です。director は、これらのバックエンド設定を各ノードの **/etc/cinder/cinder.conf** ファイルに保存します。
- 5 **FJFC** という名前のバックエンド定義を **FJFC/** 文字列で設定し、そのバックエンド定義で **volume_driver** パラメーターを宣言します。**volume_driver** パラメーターを使用して、バックエンドの Fibre Channel ETERNUS ドライバーを設定します (例: **cinder.volume.drivers.fujitsu.eternus_dx.eternus_dx_fc.FJDXFCDriver**)。
- 6 **cinder_eternus_config_file** を使用して、ドライバーがバックエンドに使用する XML 設定ファイルへのパスを設定します。**/home/stack/templates/eternus-temp.yaml** などの heat テンプレートを使用して、**/etc/cinder/eternus-fc.xml** の作成をオーケストレーションします。
- 7 **volume_backend_name** は、Block Storage サービスがバックエンドを有効にするために使用する名前です。
- 8 **FJISCSI/** 文字列で新しいバックエンド定義を設定します。**volume_driver** パラメーターを使用して、バックエンドの iSCSI ETERNUS ドライバーを設定します (例: **cinder.volume.drivers.fujitsu.eternus_dx.eternus_dx_iscsi.FJDXISCSIDriver**)。
- 9 **cinder_user_enabled_backends** クラスで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 (**cinder::config::cinder_config** クラスで定義されるものなど) にのみ使用します。

- 10 **CinderVolumeOptVolumes** を使用して、コンテナで実行中の cinder-volume サービスがホストのカスタム設定ファイルを利用できるようにします。

環境ファイルの作成後に、設定をデプロイすることができます。環境ファイル `/home/stack/templates/eternusbackend-env.yaml` の詳細については、[5章設定された Fujitsu ETERNUS バックエンドのデプロイ](#) を参照してください。

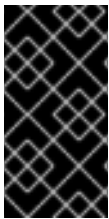
第5章 設定された FUJITSU ETERNUS バックエンドのデプロイ

`/home/stack/templates/` に `eternusbackend-env.yaml` ファイルを作成したら、以下の手順を実行します。

手順

1. **stack** ユーザとしてログインします。
2. 以下のコマンドを使用して、バックエンド設定をデプロイします。

```
$ openstack overcloud deploy --templates \  
-e [your environment files] \  
-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` 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。[6章 Fujitsu ETERNUS 設定のテスト](#) を参照してください。

第6章 FUJITSU ETERNUS 設定のテスト

新しい ETERNUS バックエンドを使用するように Block Storage サービスを設定した後、各バックエンドの **volume type** を宣言します。ボリュームタイプを使用して、新しいボリュームを作成するときに使用するバックエンドを指定します。

- Fibre Channel バックエンドを作成し、以下のコマンドで該当するバックエンドにマッピングします。

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

- iSCSI バックエンドを作成し、以下のコマンドで該当するバックエンドにマッピングします。

```
# cinder type-create FJISCSI
# cinder type-key FJISCSI volume_backend_name=FJISCSI
```

ボリュームタイプの詳しい情報は、[4章 Fujitsu ETERNUS 環境ファイルの作成](#) を参照してください。

- **test_iscsi** という名前の 1GB の iSCSI ボリュームを作成し、設定を確認します。

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

- Fibre Channel バックエンドをテストします。

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