



Red Hat OpenStack Platform 13

Fujitsu ETERNUS 백엔드 가이드

Red Hat OpenStack Platform 환경에서 Fujitsu ETERNUS 백엔드를 사용하는 가이드

Red Hat OpenStack Platform 13 Fujitsu ETERNUS 백엔드 가이드

Red Hat OpenStack Platform 환경에서 Fujitsu ETERNUS 백엔드를 사용하는 가이드

OpenStack Team
rhos-docs@redhat.com

법적 공지

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[®] 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.

초록

이 문서에서는 Fujitsu ETERNUS Disk Storage System을 백엔드로 사용하도록 Red Hat OpenStack Platform 13을 구성하는 방법에 대해 설명합니다.

차례

| | |
|---|----|
| 보다 포괄적 수용을 위한 오픈 소스 용어 교체 | 3 |
| 1장. 소개 | 4 |
| 2장. FUJITSU ETERNUS 장치 구성 | 5 |
| 3장. FUJITSU ETERNUS HEAT 템플릿 준비 | 7 |
| 3.1. 각 FUJITSU ETERNUS 백엔드에 대한 드라이버 정의 생성 | 7 |
| 3.2. FUJITSU ETERNUS HEAT 템플릿의 예 | 9 |
| 4장. FUJITSU ETERNUS 환경 파일 생성 | 11 |
| 5장. 구성된 FUJITSU ETERNUS 백엔드 배포 | 13 |
| 6장. FUJITSU ETERNUS 설정 테스트 | 14 |

보다 포괄적 수용을 위한 오픈 소스 용어 교체

Red Hat은 코드, 문서, 웹 속성에서 문제가 있는 용어를 교체하기 위해 최선을 다하고 있습니다. 먼저 마스터(master), 슬레이브(slave), 블랙리스트(blacklist), 화이트리스트(whitelist) 등 네 가지 용어를 교체하고 있습니다. 이러한 변경 작업은 작업 범위가 크므로 향후 여러 릴리스에 걸쳐 점차 구현할 예정입니다. 자세한 내용은 [CTO Chris Wright의 메시지](#)를 참조하십시오.

1장. 소개

이 문서에서는 Fujitsu ETERNUS Disk Storage System을 블록 스토리지 서비스의 백엔드로 사용하도록 Red Hat OpenStack Platform을 구성하는 방법에 대해 설명합니다. 이 문서에서는 오버클라우드 배포에서 ETERNUS 장치에서 제공하는 파이버 채널 및 iSCSI 백엔드를 정의하는 방법을 다룹니다. 이 프로세스에는 블록 스토리지 서비스의 **사용자 지정 백엔드**로 두 백엔드를 모두 정의해야 합니다. 기본적으로 컨트롤러 노드에는 Block Storage 서비스가 포함됩니다.

사전 요구 사항

- Fujitsu 플러그인의 컨테이너 registry.connect.redhat.com/fujitsu/rhosp13-fujitsu-cinder-volume-161 을 사용하여 director를 통해 배포된 Red Hat OpenStack Platform 오버클라우드. 자세한 내용은 [공급 업체 플러그인 배포](#)를 참조하십시오.
- 블록 스토리지 백엔드에는 Fujitsu ETERNUS Disk Storage System 장치 및 드라이버만 사용할 수 있습니다.
- 오버클라우드 배포에 생성한 *director* 설치 사용자를 사용할 수 있습니다. 자세한 내용은 [Director 설치 및 사용에서 스택 사용자 생성](#)을 참조하십시오.
- ETERNUS 웹 GUI 또는 CLI를 통해 ETERNUS 장치의 관리자 계정에 액세스할 수 있습니다.

Red Hat은 Fujitsu ETERNUS 장치와 함께 파이버 채널 또는 iSCSI 인터페이스 및 해당 드라이버 및 설정 사용을 지원합니다.



참고

사용자 지정 백엔드 정의에 대한 자세한 내용은 [사용자 정의 블록 스토리지 백엔드 배포 가이드](#)를 참조하십시오.

2장. FUJITSU ETERNUS 장치 구성

Fujitsu ETERNUS 장치를 블록 스토리지 백엔드로 정의하기 전에 장치의 스토리지 풀 및 포트를 구성합니다. 각 단계에 대한 자세한 내용은 장치 설명서를 참조하십시오.

절차

1. 블록 스토리지 서비스와 ETERNUS 장치의 MNT 포트를 호스팅하는 컨트롤러 노드 간 LAN 연결을 구성합니다.
2. ETERNUS 장치의 컴퓨팅 노드와 CA 포트 간 SAN 연결을 구성합니다.
3. 관리자 역할의 계정을 사용하여 ETERNUS 장치에 로그인합니다.
4. SMI-S of ETERNUS DX를 활성화합니다.
5. SSH 키 설정
6. 소프트웨어 역할이 있는 사용자 계정을 추가합니다.
7. 고급 복사 기능 라이선스를 등록하고 복사 테이블 크기를 구성합니다.
8. 볼륨의 스토리지 풀을 만듭니다. 나중에 3.1절. "각 Fujitsu ETERNUS 백엔드에 대한 드라이버 정의 생성"의 **EternusPool** 설정에서 이 풀을 사용합니다.
9. 선택 사항: 볼륨 스냅샷을 위해 별도의 스토리지 풀을 생성합니다. 이 풀은 3.1절. "각 Fujitsu ETERNUS 백엔드에 대한 드라이버 정의 생성"의 **EternusSnapPool** 설정을 나타냅니다.
10. 선택 사항: **Snap Data Pool Volume(SDPV)** 을 생성하여 스냅샷 생성 기능에 대한 **SDP(SDP)**를 활성화합니다.

작은 정보

EternusSnapPool 에 Thin Provision Pool을 사용하는 경우 이 단계를 건너뜁니다.

11. 블록 스토리지 서비스에서 사용할 스토리지 포트 를 구성합니다.
12. 스토리지 포트 포트를 **CA** 모드로 설정합니다.
13. 스토리지 포트에 대해 **호스트 방지** 를 활성화하려면 각 포트에 대해 ETERNUS CLI에서 다음 명령을 입력합니다.

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

- <PROTO>를 **fc** 또는 **iscsi** 와 같은 스토리지 프로토콜로 바꿉니다.
- <CM#> 및 <CA#>을 포트가 있는 컨트롤러 인클로저 이름으로 바꿉니다.
- <PORT>를 포트 번호로 바꿉니다.

14. 다음 명령을 사용하여 언더클라우드에서 SSH 키를 생성하고 **eternus.ietf** 파일을 ETERNUS 장치에 업로드합니다.

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



참고

나중에 사용할 수 있도록 **eternus** 파일을 저장합니다. 보안상의 이유로 언더클라우드 또는 오버클라우드에서 복사하거나 이동하지 마십시오.

3장. FUJITSU ETERNUS HEAT 템플릿 준비

향후 Red Hat OpenStack Platform 오버클라우드 업데이트 중에 설정이 유지되도록 하려면 director를 통해 배포하는 동안 모든 서비스 구성을 수행합니다.

블록 스토리지 서비스를 호스팅하는 ETERNUS 백엔드의 컨트롤러 노드에 다음 구성을 포함합니다.

- 각 백엔드의 드라이버 설정에 대한 XML 구성 파일을 포함합니다.
- ETERNUS 장치와 통신하기 위한 SSH 서버 키가 포함되어 있습니다.

heat 템플릿을 사용하여 director와 함께 두 작업을 오케스트레이션할 수 있습니다. director heat 템플릿의 구문에 대한 자세한 내용은 *Advanced Overcloud Guide*의 [Heat 템플릿 이해](#)를 참조하십시오.

다음 템플릿 **eternus-temp.yaml**은 필요한 heat 템플릿의 기본 구문을 간략하게 설명합니다.

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** 섹션에는 컨트롤러 노드의 작업을 오케스트레이션하는 리소스가 포함되어 있습니다.
- 2 **config** 섹션에는 컨트롤러 노드에서 실행할 명령이 포함되어 있습니다.
- 3 블록 스토리지 서비스가 호스팅되는 각 컨트롤러 노드에 개인 키 정보를 복사하고, 3.1절. "각 Fujitsu ETERNUS 백엔드에 대한 드라이버 정의 생성"에서 각 백엔드의 드라이버 설정에 대한 XML 구성 파일을 생성하는 명령을 추가합니다.

이 파일을 director 노드 **/home/stack/templates/**의 사용자 지정 heat 템플릿 디렉터리에 저장합니다.

3.1. 각 FUJITSU ETERNUS 백엔드에 대한 드라이버 정의 생성

블록 스토리지 구성 파일 `/etc/cinder/cinder.conf` 가 아닌 별도의 XML 파일에서 각 ETERNUS 백엔드의 드라이버 설정을 정의합니다. 다음 설정을 사용하여 각 백엔드에 XML 파일이 있는지 확인합니다.

EternusIP

ETERNUS 장치의 SMI-S 연결의 IP 주소입니다. 장치의 MNT 포트의 IP 주소를 사용합니다.

EternusPort

ETERNUS 장치의 SMI-S 연결 포트에 대한 포트 번호입니다.

EternusUser

EternusIP 연결에 대한 **소프트웨어** 역할의 사용자 이름입니다.

EternusPassword

EternusIP 에서 **EternusUser** 의 해당 암호입니다.

EternusPool

[2장. Fujitsu ETERNUS 장치 구성](#) 볼륨의 스토리지 풀 이름입니다. ETERNUS 장치에서 풀 RAID 그룹 이름 또는 TPP 이름을 사용합니다.

EternusSnapPool

[2장. Fujitsu ETERNUS 장치 구성](#) 에서 볼륨 스냅샷의 스토리지 풀 이름입니다. ETERNUS 장치에서 풀 RAID 그룹 이름을 사용합니다. 스냅샷에 대해 다른 풀을 생성하지 않은 경우 **EternusPool** 과 동일한 값을 사용합니다.

다음 xml 예제를 사용하여 파이버 채널 구성을 정의합니다.

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>
```

다음 xml 예제를 사용하여 ISCSI 구성을 정의합니다.

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>
```

이러한 XML 파일 생성을 오케스트레이션하려면 [3장. Fujitsu ETERNUS heat 템플릿 준비](#) 의 `/home/stack/templates/eternus-temp.yaml` 파일의 **EternusSetup** 리소스의 **config** 섹션에 bash 명령을 포함합니다. 다음 예제 명령을 사용하여 **eternus-fc.xml** 및 **eternus-iscsi.xml** 생성을 오케스트레이션합니다.

```
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 명령을 사용하여 필요한 양의 XML 구성 파일을 만듭니다.

cinder 사용자 및 그룹에 대해 이러한 XML 파일의 소유권 및 권한을 설정합니다.

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

완료된 heat 템플릿의 예는 3.2절. “Fujitsu ETERNUS heat 템플릿의 예” 에서 참조하십시오.

3.2. FUJITSU ETERNUS HEAT 템플릿의 예

다음 `/home/stack/templates/eternus-temp.yaml` 파일에는 `eternus-fc.xml` 및 `eternus-iscsi.xml` 과 같은 XML 구성 파일 예제를 선언하는 데 필요한 매개변수가 포함되어 있습니다.

`/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
  MIIEpAIBAAKCAQEA5yMqonpfniu+1PJ8qdWZpcf0d4UcHj2uyE7ou7vcZUQ1Cq
  s5Q5pjkCgYAxITIpfOYA8jvLgc7vMEa/ZbhUgAPIYlisxbffmRsBWyJSt9gwHpcW
  hvaWo6VD/iUKZ3bOcMK0buUwBdFUt5s9B8mXbYsX6bWovIVkyu8DzQfpDiPnV6C8
  ...
  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 2장. *Fujitsu ETERNUS 장치 구성* 의 언더클라우드에서 생성된 **eternus** 파일의 개인 키 정보입니다.

4장. FUJITSU ETERNUS 환경 파일 생성

환경 파일에는 정의하려는 각 백엔드 및 기타 관련 설정에 대한 설정이 포함되어 있습니다. 환경 파일에 대한 자세한 내용은 *Advanced Overcloud Customization* 가이드의 [환경 파일](#) 을 참조하십시오.

또한 환경 파일은 [3장. Fujitsu ETERNUS heat 템플릿 준비](#) 이전에 생성한 heat 템플릿을 등록합니다. heat 템플릿에 정의된 설치 및 에코 명령은 배포 중에 적절한 노드에서 실행됩니다.

다음 예제 환경 파일에는 ETERNUS 장치를 블록 스토리지 백엔드로 정의하는 데 필요한 섹션이 포함되어 있습니다. 또한 [3.1절. "각 Fujitsu ETERNUS 백엔드에 대한 드라이버 정의 생성"](#) 및 [3.2절. "Fujitsu ETERNUS heat 템플릿의 예"](#) 에서 오케스트레이션된 각 해당 XML 파일에 대한 백엔드 정의를 생성합니다.

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
```

1 NodeExtraConfig 를 사용하여 코어 Puppet 구성 전에 모든 노드에 대한 사용자 정의 설정을 정의합니다. 그러면 블록 스토리지 서비스가 오버클라우드에 배포되면 다음과 같은 설정이 완료됩니다.

- 각 백엔드의 XML 구성 파일이 있습니다.
- 개인 키가 생성됩니다.

- 2 다른 백엔드 유형을 비활성화하려면 다음 매개변수를 **false** 로 설정합니다.
 - **cinderEnableiscsiBackend**: 기타 iSCSI 백엔드.
 - **CinderEnableRbdBackend**: Red Hat Ceph Storage.
 - **CinderEnableNfsBackend**: NFS.
 - **NovaEnableRbdBackend**: 임시 Red Hat Ceph Storage.
- 3 **GlanceBackend** 매개 변수를 사용하여 이미지 서비스 이미지 스토리지 설정을 정의합니다. 지원되는 값은 다음과 같습니다.
 - **파일** 은 각 컨트롤러 노드의 **/var/lib/glance/images** 에 이미지를 저장합니다.
 - **Swift** 는 이미지 스토리지에 Object Storage 서비스를 사용합니다.
 - **Cinder** 는 이미지 스토리지에 블록 스토리지 서비스를 사용합니다.
- 4 **controllerExtraConfig** 를 사용하여 모든 컨트롤러 노드에 대한 사용자 정의 설정을 정의합니다. **cinder::config::cinder_config** 클래스는 Block Storage 서비스용입니다. **director**는 이러한 백엔드 설정을 각 노드의 **/etc/cinder/cinder.conf** 파일에 저장합니다.
- 5 **FJFC/ string**을 사용하여 **FJFC** 라는 백엔드 정의를 구성하고 해당 백엔드 정의에 **volume_driver** 매개 변수를 선언합니다. **volume_driver** 매개 변수를 사용하여 백엔드의 파이버 채널 ETERNUS 드라이버를 설정합니다(예: **cinder.volume.drivers.fujitsu.eternus_dx.eternus_dx_fc.FJDXFCDriver**).
- 6 드라이버가 **cinder_eternus_config_file** 을 사용하여 백엔드에 사용하는 XML 구성 파일의 경로를 설정합니다. **heat** 템플릿(예: **/home/stack/templates/eternus-temp.yaml**)을 통해 **/etc/cinder/eternus-fc.xml** 생성을 오케스트레이션합니다.
- 7 **volume_backend_name** 은 블록 스토리지 서비스에서 백엔드를 활성화하는 데 사용하는 이름입니다.
- 8 **FJISCSI/ 문자열**을 사용하여 새 백엔드 정의를 구성합니다. **volume_driver** 매개 변수를 사용하여 백엔드의 iSCSI ETERNUS 드라이버를 설정합니다(예: **cinder.volume.drivers.fujitsu.eternus_dx.eternus_dx_iscsi.FJDXISCSIDriver**).
- 9 **cinder_user_enabled_backends** 클래스를 사용하여 사용자 지정 백엔드를 설정하고 활성화합니다. **cinder::config::cinder_config** 클래스에 정의된 것과 같이 사용자 지정 백엔드에만 이 클래스를 사용합니다.
- 10 호스트의 사용자 지정 구성 파일을 **CinderVolumeOptVolumes** 가 있는 컨테이너에서 실행되는 **cinder-volume** 서비스에서 사용할 수 있도록 합니다.

환경 파일을 생성한 후 구성을 배포할 수 있습니다. 환경 파일 **/home/stack/templates/eternusbackend-env.yaml** 에 대한 자세한 내용은 [5장. 구성된 Fujitsu ETERNUS 백엔드 배포](#) 을 참조하십시오.

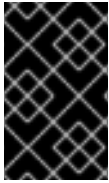
5장. 구성된 FUJITSU ETERNUS 백엔드 배포

`/home/stack/templates/` 에서 `eternusbackend-env.yaml` 파일을 생성한 후 다음 단계를 완료합니다.

절차

1. **stack** 사용자로 로그인합니다.
2. 다음 명령을 사용하여 백엔드 구성을 배포합니다.

```
$ openstack overcloud deploy --templates \
-e [your environment files] \
-e /home/stack/templates/eternusbackend-env.yaml
```



중요

오버클라우드를 생성할 때 추가 환경 파일을 전달한 경우 **-e** 옵션을 사용하여 오버클라우드를 원하지 않는 변경을 수행하지 않도록 여기에서 다시 전달합니다. 자세한 내용은 *Director 설치 및 사용 가이드* 의 [Overcloud 환경 수정](#) 을 참조하십시오.

director 오케스트레이션이 완료된 후 백엔드를 테스트합니다. [6장. Fujitsu ETERNUS 설정 테스트](#) 을 참조하십시오.

6장. FUJITSU ETERNUS 설정 테스트

새로운 ETERNUS 백엔드를 사용하도록 Block Storage 서비스를 구성한 후 각 백엔드에 대한 **볼륨 유형**을 선언합니다. 볼륨 유형을 사용하여 새 볼륨을 만들 때 사용할 백엔드를 지정합니다.

- 파이버 채널 백엔드를 생성하고 다음 명령으로 해당 백엔드에 매핑합니다.

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

- iSCSI 백엔드를 만들고 다음 명령을 사용하여 각 백엔드에 매핑합니다.

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

볼륨 유형에 대한 자세한 내용은 [4장. Fujitsu ETERNUS 환경 파일 생성](#)을 참조하십시오.

- **test_iscsi** 이라는 1GB iSCSI 볼륨을 생성하여 구성을 확인합니다.

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

- 파이버 채널 백엔드를 테스트합니다.

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