



# Red Hat OpenStack Platform 13

## 언더클라우드 및 컨트롤 플레인 백업 및 복원

업데이트 및 업그레이드 중 언더클라우드 및 오버클라우드 컨트롤 플레인 백업 및 복원  
절차



# Red Hat OpenStack Platform 13 언더클라우드 및 컨트롤 플레인 백업 및 복원

---

업데이트 및 업그레이드 중 언더클라우드 및 오버클라우드 컨트롤 플레인 백업 및 복원 절차

## 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<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.

## 초록

이 가이드에서는 언더클라우드 및 오버클라우드 컨트롤 플레인 노드에 Relax-and-Recover(ReaR)를 설치하고 구성하는 방법, 업데이트 및 업그레이드 전에 언더클라우드 및 컨트롤 플레인 노드를 백업하는 방법, 업데이트 또는 업그레이드를 수행하는 동안 오류가 발생하면 언더클라우드 및 컨트롤 플레인 노드를 복원하는 방법에 대해 설명합니다.

## 차례

<b>1장. 언더클라우드 및 컨트롤 플레인 소개가 백업 및 복원</b> .....	<b>3</b>
1.1. 배경 정보	3
1.2. 백업 관리 옵션	3
<b>2장. 백업 노드 준비</b> .....	<b>5</b>
2.1. NFS 서버 준비	5
2.2. 백업 디렉터리 생성 및 내보내기(CREATE AND EXPORT THE BACKUP DIRECTORY)	5
<b>3장. RELAX 및 RECOVER (REAR) 설치 및 구성</b> .....	<b>7</b>
3.1. REAR 설치	7
3.2. REAR 구성 파일 생성	7
<b>4장. 백업 프로시저 실행 RUN THE BACK UP PROCEDURE</b> .....	<b>9</b>
4.1. 언더클라우드를 백업하기 전에 사전 요구 사항 작업 수행	9
4.2. 언더클라우드 백업	9
4.3. 컨트롤 플레인 백업	10
<b>5장. 복원 절차 실행</b> .....	<b>14</b>
5.1. 언더클라우드 복원	14
5.2. 컨트롤 플레인 복원	15
5.3. GALERA 클러스터 문제 해결	16
5.4. CEPH 모니터와 함께 언더클라우드 및 컨트롤 플레인 노드 복원	19



## 1장. 언더클라우드 및 컨트롤 플레인 소개가 백업 및 복원

Undercloud 및 컨트롤 플레인 백업 및 복원 절차는 Red Hat OpenStack Platform 13 언더클라우드 및 오버클라우드 컨트롤러 노드의 상태를 업데이트 및 업그레이드하기 전에 컨트롤 플레인 노드라고 하는 단계를 제공합니다. 업데이트 또는 업그레이드 중에 오류가 발생하면 언더클라우드 및 오버클라우드 컨트롤 플레인 노드를 이전 상태로 복원하는 절차를 사용합니다.

### 1.1. 배경 정보

Undercloud 및 컨트롤 플레인 백업 및 복원 절차에서는 Bash로 작성된 오픈 소스 Relax 및 Recover(ReaR) 재해 복구 솔루션을 사용합니다. rear는 언더클라우드 또는 컨트롤 플레인 노드의 최신 상태로 구성된 부팅 가능한 이미지를 생성합니다. Rear 또한 시스템 관리자가 백업 파일을 선택할 수 있습니다.

rear는 다음을 포함하여 다양한 부팅 미디어 형식을 지원합니다.

- ISO
- USB
- eSATA
- PXE

이 문서의 예는 **ISO** 부팅 형식을 사용하여 테스트되었습니다.

rear는 다음을 포함하여 여러 프로토콜을 사용하여 부팅 이미지를 전송할 수 있습니다.

- HTTP/HTTPS
- SSH/SCP
- FTP/SFTP
- NFS
- CIFS(SMB)

Red Hat OpenStack Platform 13 언더클라우드 및 오버클라우드 컨트롤 플레인 노드를 백업하고 복원하는 목적으로 이 문서의 예는 NFS를 사용하여 테스트되었습니다.

### 1.2. 백업 관리 옵션

rear는 내부 및 외부 백업 관리 옵션을 모두 사용할 수 있습니다.

#### 내부 백업 관리

내부 백업 옵션은 다음과 같습니다.

- **tar**
- **rsync**

#### 외부 백업 관리

외부 백업 관리 옵션에는 오픈 소스 및 독점 솔루션이 모두 포함됩니다. 오픈 소스 솔루션은 다음과 같습니다.

- Bacula
- Bareos

전용 솔루션은 다음과 같습니다.

- EMC NetWorker (Legato)
- HP DataProtector
- IBM Tivoli Storage Manager (TSM)
- Symantec NetBackup



## 2장. 백업 노드 준비

언더클라우드 또는 컨트롤 플레인 노드를 백업하기 전에 백업 이미지를 수락하도록 백업 노드를 준비합니다.

### 2.1. NFS 서버 준비

rear는 여러 전송 방법을 사용할 수 있습니다. Red Hat은 NFS를 사용한 ReaR 백업 및 복원을 지원합니다.

1. 백업 노드에 NFS 서버를 설치합니다.

```
[root@backup ~]# yum install -y nfs-utils
```

2. 방화벽에 NFS 서비스를 추가하여 포트 **111** 및 **2049** 가 열려 있는지 확인합니다. 예를 들면 다음과 같습니다.

```
[root@backup ~]# firewall-cmd --add-service=nfs
[root@backup ~]# firewall-cmd --add-service=nfs --permanent
```

3. NFS 서버를 활성화하고 시작합니다.

```
[root@backup ~]# systemctl enable nfs-server
[root@backup ~]# systemctl restart nfs-server
```

### 2.2. 백업 디렉터리 생성 및 내보내기(CREATE AND EXPORT THE BACKUP DIRECTORY)

언더클라우드 또는 컨트롤 플레인 노드에서 백업 ISO 이미지를 백업 노드로 복사하려면 백업 디렉터리를 생성해야 합니다.

#### 사전 요구 사항

- NFS 서버를 설치하고 활성화합니다. 자세한 내용은 [NFS 서버 준비](#)를 참조하십시오.

#### 절차

1. 백업 디렉터를 생성합니다.

```
[root@backup ~]# mkdir /ctl_plane_backups
```

2. 디렉터를 내보냅니다. **&lt;IP\_ADDRESS/24&gt;**를 네트워크의 IP 주소 및 서브넷 마스크로 바꿉니다.

```
[root@backup ~]# cat >> /etc/exports << EOF
/ctl_plane_backups <IP_ADDRESS/24>(rw,sync,no_root_squash,no_subtree_check)
EOF
```

**/etc/exports** 파일의 항목은 공백으로 구분된 목록에 있습니다. 언더클라우드 및 오버클라우드 컨트롤 플레인 노드에서 다른 네트워크 또는 서브넷을 사용하는 경우 다음 예와 같이 각 네트워크 또는 서브넷에 대해 이 단계를 반복합니다.

```
cat >> /etc/exports << EOF
```

```
/ctl_plane_backups 192.168.24.0/24(rw,sync,no_root_squash,no_subtree_check) /
ctl_plane_backups 10.0.0.0/24(rw,sync,no_root_squash,no_subtree_check) /
ctl_plane_backups 172.16.0.0/24(rw,sync,no_root_squash,no_subtree_check)
EOF
```

3. NFS 서버를 다시 시작하십시오.

```
[root@backup ~]# systemctl restart nfs-server
```

4. 항목이 NFS 서버에 올바르게 구성되었는지 확인합니다.

```
[root@backup ~]# showmount -e `hostname`
```

## 3장. RELAX 및 RECOVER (REAR) 설치 및 구성

언더클라우드와 오버클라우드 컨트롤 플레인을 백업하려면 먼저 언더클라우드 및 각 컨트롤 플레인 노드에 ReaR을 설치하고 구성해야 합니다.

### 3.1. REAR 설치

언더클라우드 및 컨트롤 플레인을 백업하기 전에 언더클라우드 노드와 각 컨트롤 플레인 노드에 ISO 이미지를 생성하기 위해 Relax 및 Recover(ReaR) 패키지 및 패키지를 설치해야 합니다.

#### 절차

1. 언더클라우드 노드 및 각 컨트롤 플레인 노드에 ISO 이미지를 생성하기 위해 ReaR 패키지 및 패키지를 설치합니다.

```
[root@controller-x ~]# yum install rear genisoimage nfs-utils -y
```

2. 향후 업그레이드 문제를 방지하려면 예약된 ReaR cron 작업을 삭제합니다.

```
[root@controller-x ~]# rm -f /etc/cron.d/rear
```

3. 언더클라우드 및 각 컨트롤 플레인 노드에 백업 디렉터리를 생성합니다.

```
[root@controller-x ~]# mkdir -p /ctl_plane_backups
```

4. 언더클라우드 및 각 컨트롤 플레인 노드에서 NFS를 실행하는 백업 노드에서 **ctl\_plane\_backups** NFS 디렉터리를 마운트합니다.

```
[root@controller-x ~]# mount -t nfs <BACKUP_NODE_IP_ADDRESS>:/ctl_plane_backups /ctl_plane_backups
```

& lt;<BACKUP\_NODE\_IP\_ADDRESS >를 NFS 서버를 실행하는 백업 노드의 IP 주소로 바꿉니다.

### 3.2. REAR 구성 파일 생성

언더클라우드 노드와 각 컨트롤 플레인 노드에 Relax 및 Recovery(ReaR)를 설치한 후 구성 파일을 **root** 사용자로 생성합니다.

#### 절차

1. ReaR 구성 파일을 생성합니다.

```
[root@controller-x ~]# mkdir -p /etc/rear
[root@controller-x ~]# tee -a "/etc/rear/local.conf" > /dev/null <<'EOF'
OUTPUT=ISO
OUTPUT_URL=nfs://<BACKUP_NODE_IP_ADDRESS>/ctl_plane_backups
ISO_PREFIX=<NODE_HOSTNAME>
BACKUP=NETFS
BACKUP_PROG_COMPRESS_OPTIONS=( --gzip )
BACKUP_PROG_COMPRESS_SUFFIX=".gz"
BACKUP_PROG_EXCLUDE=( '/tmp/*' '/data/*' )
BACKUP_URL=nfs://<BACKUP_NODE_IP_ADDRESS>/ctl_plane_backups
```

```
BACKUP_PROG_EXCLUDE=("${BACKUP_PROG_EXCLUDE[@]}" '/media' '/var/tmp'
'/var/crash')
BACKUP_PROG_OPTIONS+=( --anchored --xattrs-include='*.*' --xattrs )
EOF
```

- **< NODE\_HOSTNAME >**을 노드의 호스트 이름으로 바꿉니다. 예를 들어 노드 호스트 이름이 **controller-0** 인 경우 **< NODE\_HOSTNAME >**을 **controller-0** 으로 바꿉니다.
- **< BACKUP\_NODE\_IP\_ADDRESS >**를 NFS 서버를 실행하는 백업 노드의 IP 주소로 바꿉니다. 백업 노드를 준비할 때 구성한 IP 주소입니다. 자세한 내용은 [2장. 백업 노드 준비](#)의 내용을 참조하십시오.



**중요**

언더클라우드 또는 컨트롤 플레인 노드에서 부팅 모드 UEFI를 사용하는 경우 구성 파일에 **USING\_UEFI\_BOOTLOADER=1** 을 추가해야 합니다.

2. 네트워크 인터페이스에 OVS 브리지를 사용하는 경우 **/etc/rear/local.conf** 파일에 다음 구성을 추가하여 네트워크를 통해 백업을 검색할 수 있습니다.

```
NETWORKING_PREPARATION_COMMANDS=('<command_1>' '<command_2>' ...
'<command_n>')
```

**< command\_1 >**, 및 모든 명령 **< command\_n >**.을 네트워크 구성을 구성하는 데 필요한 명령으로 바꿉니다. 예를 들면 다음과 같습니다.

- 컨트롤 플레인 브리지 이름을 구성하려면 **ip link add br-ctlplane type bridge** 명령을 추가합니다.
- 인터페이스 이름을 설정하려면 **ip link set eth0 up** 명령을 추가합니다.

3. **rescue.conf** 파일을 생성합니다.

```
[root@controller-x ~]# tee -a "/etc/rear/rescue.conf" > /dev/null <<'EOF'
BACKUP_PROG_OPTIONS+=( --anchored --xattrs-include='*.*' --xattrs )
EOF
```

## 4장. 백업 프로시저 실행 RUN THE BACK UP PROCEDURE

빠른 전달 업그레이드를 수행하기 전에 오류가 발생하면 언더클라우드 및 오버클라우드 컨트롤 플레인 노드를 백업하여 이전 상태로 복원할 수 있습니다.



### 참고

언더클라우드 및 오버클라우드를 백업하기 전에 언더클라우드에서 오버클라우드에서 작업을 수행하지 않는지 확인합니다.

### 4.1. 언더클라우드를 백업하기 전에 사전 요구 사항 작업 수행

언더클라우드를 배포하거나 기존 언더클라우드를 변경할 때 언더클라우드 백업을 수행하지 마십시오.

데이터 손상을 방지하려면 스택 오류와 지속적인 작업이 없는지 확인하고, 언더클라우드 노드를 백업하기 전에 **mariadb** 를 제외한 모든 OpenStack 서비스가 중지되었는지 확인합니다.

#### 절차

1. 스택에 오류가 없는지 확인합니다. **<it>STACKNAME </it>** 을 스택 이름으로 바꿉니다. 배포 및 사용 가능한 모든 스택에 명령을 사용합니다.

```
(undercloud) [stack@undercloud-0 ~]$ openstack stack failures list <STACKNAME>
```

2. 언더클라우드에 진행 중인 작업이 없는지 확인합니다.

```
(undercloud) [stack@undercloud-0 ~]$ openstack stack list --nested | grep -v "_COMPLETE"
```

명령에서 결과를 반환하지 않으면 진행 중인 작업이 없습니다.

3. 언더클라우드에서 모든 OpenStack 서비스를 중지합니다.

```
# systemctl stop openstack-*
# systemctl stop neutron-*
# systemctl stop ironic*
# systemctl stop haproxy
# systemctl stop httpd
```

4. **mariadb** 가 실행 중인지 확인합니다.

```
# sudo systemctl status mariadb
```

### 4.2. 언더클라우드 백업

언더클라우드 노드를 백업하려면 언더클라우드 노드에 root 사용자로 로그인해야 합니다. 예방 조치로 데이터베이스를 백업하여 복원할 수 있는지 확인해야 합니다. As a precaution, you must back up the database to ensure that you can restore it.

#### 사전 요구 사항

- 백업 디렉토리를 생성하고 내보냈습니다. 자세한 내용은 [백업 디렉터리 생성 및 내보내기](#)를 참조하십시오.

- 언더클라우드를 백업하기 전에 사전 요구 사항 작업을 수행했습니다. 자세한 내용은 [언더클라우드를 백업하기 전에 전체 조건 작업 수행](#)을 참조하십시오.
- 각 컨트롤 플레인 노드에 ReaR을 설치하고 구성했습니다. 자세한 내용은 [Relax 및 Recover\(ReaR\) 설치 및 구성](#)을 참조하십시오.

## 절차

1. 데이터베이스 암호를 찾습니다.

```
[root@undercloud stack]# PASSWORD=$(sudo /bin/hiera -c /etc/puppet/hiera.yaml
mysql::server::root_password)
```

2. 데이터베이스를 백업합니다.

```
[root@undercloud stack]# mysql -uroot -p$PASSWORD -s -N -e "select distinct
table_schema from information_schema.tables where engine='innodb' and table_schema !=
'mysql';" | xargs mysqldump -uroot -p$PASSWORD --single-transaction --databases >
openstack-backup-mysql.sql
```

```
[root@undercloud stack]# mysql -uroot -p$PASSWORD -s -N -e "SELECT
CONCAT("\SHOW GRANTS FOR '",user,'"@'",host,'"');" FROM mysql.user where
(length(user) > 0 and user NOT LIKE 'root')" | xargs -n1 mysql -uroot -p$PASSWORD -s -N -
e | sed 's/;/;' > openstack-backup-mysql-grants.sql
```

3. **mariadb** 데이터베이스 서비스를 중지합니다.

```
[root@undercloud stack]# systemctl stop mariadb
```

4. 백업을 생성합니다.

```
[root@undercloud stack]# rear -d -v mkbackup
```

**/ctl\_plane\_backups** 디렉터리의 백업 노드에서 ReaR으로 생성한 백업 ISO 파일을 찾을 수 있습니다.

5. 언더클라우드를 다시 시작합니다.

- a. stack 사용자로 언더클라우드에 로그인합니다.
- b. 언더클라우드를 다시 시작합니다.

```
[stack@undercloud]$ sudo reboot
```

## 4.3. 컨트롤 플레인 백업

컨트롤 플레인을 백업하려면 먼저 **pacemaker** 클러스터 및 컨트롤 플레인 노드에서 작동하는 모든 컨테이너를 중지해야 합니다. 상태 일관성을 보장하기 위해 스택을 작동하지 마십시오. 백업 절차를 완료한 후 **pacemaker** 클러스터와 컨테이너를 시작합니다.

precaution으로 **pacemaker** 클러스터 및 컨테이너를 다시 시작한 후 데이터베이스를 복원할 수 있도록 데이터베이스를 백업해야 합니다.

컨트롤 플레인 노드를 동시에 백업합니다.

### 사전 요구 사항

- 백업 디렉토리를 생성하고 내보냈습니다. 자세한 내용은 [백업 디렉터리 생성 및 내보내기를](#) 참조하십시오.
- 각 컨트롤 플레인 노드에 ReaR을 설치하고 구성했습니다. 자세한 내용은 [Relax 및 Recover\(ReaR\) 설치 및 구성](#)을 참조하십시오.

### 절차

1. 데이터베이스 암호를 찾습니다.

```
[heat-admin@overcloud-controller-x ~]# PASSWORD=$(sudo /bin/hiera -c /etc/puppet/hiera.yaml mysql::server::root_password)
```

2. 데이터베이스를 백업합니다.

```
[heat-admin@overcloud-controller-x ~]# mysql -uroot -p$PASSWORD -s -N -e "select distinct table_schema from information_schema.tables where engine='innodb' and table_schema != 'mysql';" | xargs mysqldump -uroot -p$PASSWORD --single-transaction --databases > openstack-backup-mysql.sql
```

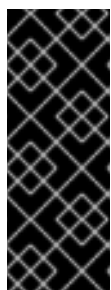
```
[heat-admin@overcloud-controller-x ~]# mysql -uroot -p$PASSWORD -s -N -e "SELECT CONCAT("\SHOW GRANTS FOR '",user,'"@",host,'"');" FROM mysql.user where (length(user) > 0 and user NOT LIKE 'root')" | xargs -n1 mysql -uroot -p$PASSWORD -s -N -e | sed 's/$;/ ' > openstack-backup-mysql-grants.sql
```



### 참고

데이터베이스 백업은 예방 조치입니다. Backing up the databases is a precautionary measure. 이 단계를 수행하면 복원 절차의 일부로 자동으로 복원되지 않는 경우 Galera 클러스터를 수동으로 복원할 수 있습니다. Galera 클러스터를 복원하는 방법에 대한 자세한 내용은 [Galera 클러스터 문제 해결](#)을 참조하십시오.

3. 컨트롤 플레인 노드 중 하나에서 pacemaker 클러스터를 중지합니다.



### 중요

스택을 작동시키지 마십시오. pacemaker 클러스터와 컨테이너를 중지하면 컴퓨팅 노드로 컨트롤 플레인 서비스가 일시적으로 중단됩니다. 또한 네트워크 연결, Ceph 및 NFS 데이터 플레인 서비스가 중단되었습니다. 인스턴스를 생성하거나, 인스턴스를 마이그레이션하거나, 요청을 인증하거나, pacemaker 클러스터 및 컨테이너가 이 절차의 마지막 단계에 따라 컨테이너가 서비스로 돌아갈 때까지 클러스터 상태를 모니터링할 수 없습니다.

```
[heat-admin@overcloud-controller-x ~]# sudo pcs cluster stop --all
```

4. 각 컨트롤 플레인 노드에서 컨테이너를 중지합니다.

- a. 컨테이너를 중지합니다.

-

```
[heat-admin@overcloud-controller-x ~]# sudo docker stop $(sudo docker ps -a -q)
```

- b. `ceph-mon@controller.service` 컨테이너를 중지합니다.

```
[heat-admin@overcloud-controller-x ~]# sudo systemctl stop ceph-mon@$(hostname -s)
```

- c. `ceph-mgr@controller.service` 컨테이너를 중지합니다.

```
[heat-admin@overcloud-controller-x ~]# sudo systemctl stop ceph-mgr@$(hostname -s)
```

5. 선택 사항: **ganesha-nfs** 를 사용하는 경우 한 컨트롤러에서 파일 서버를 비활성화합니다.

```
[heat-admin@overcloud-controller-x ~]# sudo pcs resource disable ceph-nfs
```

6. 선택 사항: ceph 서비스 **ceph-mds** 및 **ceph-rgw** 를 사용하는 경우 다음 서비스를 중지합니다.

```
[heat-admin@overcloud-controller-x ~]# sudo systemctl stop ceph-mds@$(hostname -s)
```

```
[heat-admin@overcloud-controller-x ~]# sudo systemctl stop ceph-rgw@$(hostname -s)
```

7. 컨트롤 플레인을 백업하려면 각 컨트롤 플레인 노드에서 컨트롤 플레인 백업을 실행합니다.

```
[heat-admin@overcloud-controller-x ~]# sudo rear -d -v mkbackup
```

`/ctl_plane_backups` 디렉터리의 백업 노드에서 ReaR으로 생성한 백업 ISO 파일을 찾을 수 있습니다.



### 참고

backup 명령을 실행하면 다음 경고와 유사하게 tar 프로세스 중에 무시된 **tar** 명령 및 소켓에 관한 경고 메시지가 표시될 수 있습니다.

```
WARNING: tar ended with return code 1 and below output:
```

```
---snip---
```

```
tar: /var/spool/postfix/public/qmgr: socket ignored
```

```
...
```

```
...
```

This message indicates that files have been modified during the archiving process and the backup might be inconsistent. Relax-and-Recover continues to operate, however, it is important that you verify the backup to ensure that you can use this backup to recover your system.

8. 백업 절차가 각 컨트롤 플레인 노드에 대한 ISO 이미지를 생성하면 pacemaker 클러스터를 다시 시작합니다. 컨트롤 플레인 노드 중 하나에서 다음 명령을 입력합니다.

```
[heat-admin@overcloud-controller-x ~]# sudo pcs cluster start --all
```

9. 각 컨트롤 플레인 노드에서 컨테이너를 시작합니다.

- a. 컨테이너를 시작합니다.

```
[heat-admin@overcloud-controller-x ~]# sudo systemctl restart docker
```



b. **ceph-mon@controller.service** 컨테이너를 시작합니다.

```
[heat-admin@overcloud-controller-x ~]# sudo systemctl start ceph-mon@$(hostname -s)
```

c. **ceph-mgr@controller.service** 컨테이너를 시작합니다.

```
[heat-admin@overcloud-controller-x ~]# sudo systemctl start ceph-mgr@$(hostname -s)
```

10. 선택 사항: **ceph-mds** 및 **ceph-rgw** 를 사용하는 경우 다음 서비스를 시작합니다.

```
[heat-admin@overcloud-controller-x ~]# sudo systemctl start ceph-rgw@$(hostname -s)
[heat-admin@overcloud-controller-x ~]# sudo systemctl start ceph-mds@$(hostname -s)
```

11. 선택 사항: **ganasha-nfs** 를 사용하는 경우 한 컨트롤러에서 파일 서버를 활성화합니다.

```
[heat-admin@overcloud-controller-x ~]# sudo pcs resource enable ceph-nfs
```

## 5장. 복원 절차 실행

업데이트 또는 업그레이드하는 동안 오류가 발생하면 언더클라우드 또는 오버클라우드 컨트롤 플레인 노드 또는 둘 다를 복원하여 이전 상태를 가정할 수 있습니다. Galera 클러스터가 복원 절차의 일부로 자동으로 복원되지 않으면 클러스터를 수동으로 복원해야 합니다.

공동 배치된 ceph 모니터로 언더클라우드 또는 오버클라우드 컨트롤 플레인 노드를 복원할 수도 있습니다.



### 참고

ISO 파일에서 부팅하는 경우 언더클라우드와 오버클라우드에서 NFS 서버에 연결할 수 있는지 확인합니다.

다음 일반 단계를 사용합니다.

1. 부팅 가능한 ISO 이미지를 DVD로 굽거나 ILO 원격 액세스를 통해 로드하십시오.
2. 복구 미디어에서 복원이 필요한 노드를 부팅합니다.
3. **Recover <HOSTNAME>**을 선택합니다. **&lt;HOSTNAME >**을 복원할 노드 이름으로 바꿉니다.
4. 사용자 **root** 로 에 로그인합니다.
5. 백업을 복구합니다.

### 5.1. 언더클라우드 복원

fast-forward 업그레이드 중에 오류가 발생하면 4.2절. "언더클라우드 백업" 절차를 사용하여 생성한 ISO 이미지를 사용하여 언더클라우드 노드를 이전에 저장된 상태로 복원할 수 있습니다. 백업 프로시저는 2.2 절. "백업 디렉터리 생성 및 내보내기(Create and export the backup directory)" 단계에서 생성한 폴더의 백업 노드에 ISO 이미지를 저장합니다.

#### 절차

1. 언더클라우드 노드를 종료합니다. 계속하기 전에 언더클라우드 노드가 완전히 종료되었는지 확인합니다.
2. 백업 프로세스 중에 생성된 ISO 이미지로 부팅하여 언더클라우드 노드를 복원합니다. ISO 이미지는 Backup 노드의 **/ctl\_plane\_backups** 디렉터리에 있습니다.
3. **Relax-and-Recover** 부팅 메뉴가 나타나면 **Recover <UNDERCLOUD\_NODE >**를 선택합니다. 여기서 **<UNDERCLOUD\_NODE>**는 언더클라우드 노드의 이름입니다.
4. **root** 사용자로 로그인합니다.  
다음 메시지가 표시됩니다.

```
Welcome to Relax-and-Recover. Run "rear recover" to restore your system!
RESCUE <UNDERCLOUD_NODE>:~ # rear recover
```

이미지 복원이 빠르게 진행됩니다. 완료되면 콘솔에서 다음 메시지를 에코합니다.

```
Finished recovering your system
Exiting rear recover
Running exit tasks
```

- 명령줄 인터페이스를 사용할 수 있으면 이미지가 복원됩니다. 노드를 해제합니다.

```
RESCUE <UNDERCLOUD_NODE>:~ # poweroff
```

부팅 시 노드가 이전 상태로 재개됩니다.

## 5.2. 컨트롤 플레인 복원

fast-forward 업그레이드 중에 오류가 발생하면 4.3절. "컨트롤 플레인 백업" 절차를 사용하여 생성된 ISO 이미지를 사용하여 컨트롤 플레인 노드를 이전에 저장된 상태로 복원할 수 있습니다. 컨트롤 플레인을 복원하려면 모든 컨트롤 플레인 노드를 이전 상태로 복원하여 상태 일관성을 유지해야 합니다.



### 참고

Red Hat은 OVS(Open vSwitch) 및 기본 OVN(Open Virtual Network)과 같은 기본 SDN을 사용하는 Red Hat OpenStack Platform의 백업을 지원합니다. 타사 SDN에 대한 자세한 내용은 타사 SDN 설명서를 참조하십시오.

### 절차

- 각 컨트롤 플레인 노드를 종료합니다. 진행하기 전에 컨트롤 플레인 노드가 완전히 종료되었는지 확인합니다.
- 백업 프로세스 중에 생성한 ISO 이미지로 부팅하여 컨트롤 플레인 노드를 복원합니다. ISO 이미지는 Backup 노드의 **/ctl\_plane\_backups** 디렉터리에 있습니다.
- Relax-and-Recover** 부팅 메뉴가 나타나면 **Recover <CONTROL\_PLANE\_NODE>**를 선택합니다. <CONTROL\_PLANE\_NODE>를 컨트롤 플레인 노드의 이름으로 바꿉니다. 다음 메시지가 표시됩니다.

```
Welcome to Relax-and-Recover. Run "rear recover" to restore your system!
RESCUE <CONTROL_PLANE_NODE>:~ # rear recover
```

이미지 복원이 빠르게 진행됩니다. 복원이 완료되면 콘솔에서 다음 메시지를 에코합니다.

```
Finished recovering your system
Exiting rear recover
Running exit tasks
```

명령줄 인터페이스를 사용할 수 있으면 이미지가 복원됩니다. 노드를 해제합니다.

```
RESCUE <CONTROL_PLANE_NODE>:~ # poweroff
```

부팅 시퀀스를 일반 부팅 장치로 설정합니다. 부팅 시 노드가 이전 상태로 재개됩니다.

- 서비스가 올바르게 실행되고 있는지 확인하려면 pacemaker의 상태를 확인합니다. 컨트롤러에 **root** 사용자로 로그인하고 다음 명령을 실행합니다.

```
# pcs status
```

5. 오버클라우드 상태를 보려면 Tempest를 사용합니다. Tempest에 대한 자세한 내용은 [OpenStack Integration Test Suite Guide](#)의 4 장을 참조하십시오.

### 5.3. GALERA 클러스터 문제 해결

Galera 클러스터가 복원 절차의 일부로 복원되지 않으면 Galera를 수동으로 복원해야 합니다.



#### 참고

이 절차에서는 하나의 컨트롤러 노드에서 일부 단계를 수행해야 합니다. 절차를 진행하는 것과 동일한 컨트롤러 노드에서 다음 단계를 수행해야 합니다.

#### 절차

1. Controller-0에서 Galera 클러스터 가상 IP를 검색합니다.

```
$ sudo hiera -c /etc/puppet/hiera.yaml mysql_vip
```

2. 모든 컨트롤러 노드에서 가상 IP를 통해 데이터베이스 연결을 비활성화합니다.

```
$ sudo iptables -I INPUT -p tcp --destination-port 3306 -d $MYSQL_VIP -j DROP
```

3. Controller-0에서 MySQL 루트 암호를 검색합니다.

```
$ sudo hiera -c /etc/puppet/hiera.yaml mysql::server::root_password
```

4. Controller-0에서 Galera 리소스를 **Unmanaged** 모드로 설정합니다.

```
$ sudo pcs resource unmanage galera-bundle
```

5. 모든 컨트롤러 노드에서 MySQL 컨테이너를 중지합니다.

```
$ sudo docker container stop $(sudo docker container ls --all --format "{{.Names}}") --filter=name=galera-bundle)
```

6. 모든 컨트롤러 노드에서 현재 디렉토리를 이동합니다.

```
$ sudo mv /var/lib/mysql /var/lib/mysql-save
```

7. 모든 컨트롤러 노드에서 새 디렉토리 **/var/lib/mysq** 를 생성합니다.

```
$ sudo mkdir /var/lib/mysql
$ sudo chown 42434:42434 /var/lib/mysql
$ sudo chcon -t container_file_t /var/lib/mysql
$ sudo chmod 0755 /var/lib/mysql
$ sudo chcon -r object_r /var/lib/mysql
$ sudo chcon -u system_u /var/lib/mysql
```

8. 모든 컨트롤러 노드에서 MySQL 컨테이너를 시작합니다.

```
$ sudo docker container start $(sudo docker container ls --all --format "{{.Names}}") --filter=name=galera-bundle)
```

9. 모든 컨트롤러 노드에서 MySQL 데이터베이스를 생성합니다.

```
$ sudo docker exec -i $(sudo docker container ls --all --format "{{ .Names }}" \
  --filter=name=galera-bundle) bash -c "mysql_install_db --datadir=/var/lib/mysql --
  user=mysql"
```

10. 모든 컨트롤러 노드에서 데이터베이스를 시작합니다.

```
$ sudo docker exec $(sudo docker container ls --all --format "{{ .Names }}" \
  --filter=name=galera-bundle) bash -c "mysqld_safe --skip-networking --wsrep-on=OFF" &
```

11. 모든 컨트롤러 노드에서 **.my.cnf** Galera 구성 파일을 이동합니다.

```
$ sudo docker exec $(sudo docker container ls --all --format "{{ .Names }}" \
  --filter=name=galera-bundle) bash -c "mv /root/.my.cnf /root/.my.cnf.bck"
```

12. 모든 컨트롤러 노드에서 Galera 루트 암호를 재설정합니다.

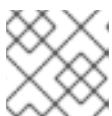
```
$ sudo docker exec $(sudo docker container ls --all --format "{{ .Names }}" \
  --filter=name=galera-bundle) bash -c "mysql -uroot -e'use mysql;update user set
  password=PASSWORD(\"$ROOTPASSWORD\")where User='root';flush privileges;"
```

13. 모든 컨트롤러 노드의 Galera 컨테이너 내부에 **.my.cnf** Galera 구성 파일을 복원합니다.

```
$ sudo docker exec $(sudo docker container ls --all --format "{{ .Names }}" \
  --filter=name=galera-bundle) bash -c "mv /root/.my.cnf.bck /root/.my.cnf"
```

14. Controller-0에서 백업 데이터베이스 파일을 **/var/lib/MySQL** 에 복사합니다.

```
$ sudo cp $BACKUP_FILE /var/lib/mysql
$ sudo cp $BACKUP_GRANT_FILE /var/lib/mysql
```



### 참고

이러한 파일의 경로는 /home/heat-admin/입니다.

15. Controller-0에서 MySQL 데이터베이스를 복원합니다.

```
$ sudo docker exec $(docker container ls --all --format "{{ .Names }}" \
  --filter=name=galera-bundle) bash -c "mysql -u root -p$ROOT_PASSWORD <
  \" /var/lib/mysql/$BACKUP_FILE \" "
```

```
$ sudo docker exec $(sudo docker container ls --all --format "{{ .Names }}" \
  --filter=name=galera-bundle) bash -c "mysql -u root -p$ROOT_PASSWORD <
  \" /var/lib/mysql/$BACKUP_GRANT_FILE \" "
```

16. 모든 컨트롤러 노드에서 데이터베이스를 종료합니다.

```
$ sudo docker exec $(sudo docker container ls --all --format "{{ .Names }}" \
  --filter=name=galera-bundle) bash -c "mysqladmin shutdown"
```

17. Controller-0에서 부트스트랩 노드를 시작합니다.

```
$ sudo docker exec $(sudo docker container ls --all --format "{{ .Names }}" --
filter=name=galera-bundle) \
  /usr/bin/mysqld_safe --pid-file=/var/run/mysql/mysqld.pid --
socket=/var/lib/mysql/mysql.sock --datadir=/var/lib/mysql \
  --log-error=/var/log/mysql_cluster.log --user=mysql --open-files-limit=16384 \
  --wsrep-cluster-address=gcomm:// &
```

18. 검증: Controller-0에서 클러스터 상태를 확인합니다.

```
$ sudo docker exec $(sudo docker container ls --all --format "{{ .Names }}" \
--filter=name=galera-bundle) bash -c "clustercheck"
```

"Galera 클러스터 노드가 동기화됨"이라는 메시지가 표시되는지 확인합니다. 그러지 않으면 노드를 다시 생성해야 합니다.

19. Controller-0에서 구성에서 클러스터 주소를 검색합니다.

```
$ sudo docker exec $(sudo docker container ls --all --format "{{ .Names }}" \
--filter=name=galera-bundle) bash -c "grep wsrep_cluster_address /etc/my.cnf.d/galera.cnf" |
awk '{print $3}'
```

20. 나머지 각 컨트롤러 노드에서 데이터베이스를 시작하고 클러스터를 검증합니다.

- a. 데이터베이스를 시작합니다.

```
$ sudo docker exec $(sudo docker container ls --all --format "{{ .Names }}" \
--filter=name=galera-bundle) /usr/bin/mysqld_safe --pid-
file=/var/run/mysql/mysqld.pid --socket=/var/lib/mysql/mysql.sock \
--datadir=/var/lib/mysql --log-error=/var/log/mysql_cluster.log --user=mysql --open-
files-limit=16384 \
--wsrep-cluster-address=$CLUSTER_ADDRESS &
```

- b. MySQL 클러스터의 상태를 확인합니다.

```
$ sudo docker exec $(sudo docker container ls --all --format "{{ .Names }}" \
--filter=name=galera-bundle) bash -c "clustercheck"
```

"Galera 클러스터 노드가 동기화됨"이라는 메시지가 표시되는지 확인합니다. 그러지 않으면 노드를 다시 생성해야 합니다.

21. 모든 컨트롤러 노드에서 MySQL 컨테이너를 중지합니다.

```
$ sudo docker exec $(sudo docker container ls --all --format "{{ .Names }}" --
filter=name=galera-bundle) \
  /usr/bin/mysqladmin -u root shutdown
```

22. 모든 컨트롤러 노드에서 가상 IP 주소를 통한 데이터베이스 연결을 허용하도록 다음 방화벽 규칙을 제거합니다.

```
$ sudo iptables -D INPUT -p tcp --destination-port 3306 -d $MYSQL_VIP -j DROP
```

23. 모든 컨트롤러 노드에서 MySQL 컨테이너를 다시 시작합니다.

```
$ sudo docker container restart $(sudo docker container ls --all --format "{{.Names }}" --filter=name=galera-bundle)
```

24. 모든 컨트롤러 노드에서 **clustercheck** 컨테이너를 다시 시작합니다.

```
$ sudo docker container restart $(sudo docker container ls --all --format "{{.Names }}" --filter=name=clustercheck)
```

25. Controller-0에서 Galera 리소스를 **관리** 모드로 설정합니다.

```
$ sudo pcs resource manage galera-bundle
```

## 5.4. CEPH 모니터와 함께 언더클라우드 및 컨트롤 플레인 노드 복원

업데이트 또는 업그레이드하는 동안 오류가 발생하면 ReaR 백업을 사용하여 언더클라우드 또는 오버클라우드 컨트롤 플레인 노드 또는 둘 다 이전 상태로 복원할 수 있습니다.

### 사전 요구 사항

- ReaR 설치 및 구성. 자세한 내용은 [ReaR 설치 및 구성](#) 을 참조하십시오.
- 백업 노드를 준비합니다. 자세한 내용은 [백업 노드 준비](#) 를 참조하십시오.
- 백업 프로시저를 실행합니다. Execute the backup procedure. 자세한 내용은 [백업 절차 실행](#) 을 참조하십시오.

### 절차

1. 백업 노드에서 NFS 디렉토리를 내보내 Ceph 백업을 호스팅합니다. &lt;IP\_ADDRESS/24> 를 네트워크의 IP 주소 및 서브넷 마스크로 바꿉니다.

```
[root@backup ~]# cat >> /etc/exports << EOF
/ceph_backups <IP_ADDRESS/24>(rw, sync, no_root_squash, no_subtree_check)
EOF
```

2. 언더클라우드 노드에서 언더클라우드 자격 증명을 가져오고 다음 스크립트를 실행합니다.

```
# source stackrc
```

```
#!/bin/bash
for i in `openstack server list -c Name -c Networks -f value | grep controller | awk -F='{' '{print $2}' | awk -F'}' '{print $1}'`; do ssh -q heat-admin@$i 'sudo systemctl stop ceph-mon@$(hostname -s) ceph-mgr@$(hostname -s)'; done
```

**ceph-mgr@controller.service** 컨테이너가 중지되었는지 확인하려면 다음 명령을 입력합니다.

```
[heat-admin@overcloud-controller-x ~]# sudo docker ps | grep ceph
```

3. 언더클라우드 노드에서 언더클라우드 자격 증명을 가져오고 다음 스크립트를 실행합니다.

```
# source stackrc
```

```
#!/bin/bash
for i in `openstack server list -c Name -c Networks -f value | grep controller | awk -F=' '{print $2}' | awk -F' '{print $1}'`; do ssh -q heat-admin@$i 'sudo mkdir /ceph_backups'; done

#!/bin/bash
for i in `openstack server list -c Name -c Networks -f value | grep controller | awk -F=' '{print $2}' | awk -F' '{print $1}'`; do ssh -q heat-admin@$i 'sudo mount -t nfs
<BACKUP_NODE_IP_ADDRESS>:/ceph_backups /ceph_backups'; done

#!/bin/bash
for i in `openstack server list -c Name -c Networks -f value | grep controller | awk -F=' '{print $2}' | awk -F' '{print $1}'`; do ssh -q heat-admin@$i 'sudo mkdir /ceph_backups/${hostname
-s)'; done

#!/bin/bash
for i in `openstack server list -c Name -c Networks -f value | grep controller | awk -F=' '{print $2}' | awk -F' '{print $1}'`; do ssh -q heat-admin@$i 'sudo tar -zcv --xattrs-include=*. * --xattrs
--xattrs-include=security.capability --xattrs-include=security.selinux --acls -f
/ceph_backups/${hostname -s}/${hostname -s}.tar.gz /var/lib/ceph'; done
```

4. 복원할 노드에서 다음 작업을 완료합니다.

- 계속하기 전에 노드의 전원을 끕니다.
- 백업 프로세스 중에 생성한 ReaR 백업 파일을 사용하여 노드를 복원합니다. 파일은 백업 노드의 **/ceph\_backups** 디렉터리에 있습니다.
- Relax-and-Recover** 부팅 메뉴에서 **Recover <CONTROL\_PLANE\_NODE >**를 선택합니다. 여기서 **<CONTROL\_PLANE\_NODE >**는 컨트롤 플레인 노드의 이름입니다.
- 프롬프트에 다음 명령을 입력합니다.

```
RESCUE <CONTROL_PLANE_NODE> :~ # rear recover
```

이미지 복원 프로세스가 완료되면 콘솔에 다음 메시지가 표시됩니다.

```
Finished recovering your system
Exiting rear recover
Running exit tasks
```

5. 복원할 노드의 경우 **/ceph\_backups** 디렉토리의 Ceph 백업을 **/var/lib/ceph** 디렉터리에 복사합니다.

- 시스템 마운트 지점을 확인합니다.

```
RESCUE <CONTROL_PLANE_NODE>:~# df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        16G   0  16G   0% /dev
tmpfs           16G   0  16G   0% /dev/shm
tmpfs           16G  8.4M  16G   1% /run
tmpfs           16G   0  16G   0% /sys/fs/cgroup
/dev/vda2       30G  13G  18G  41% /mnt/local
```

**/dev/vda2** 파일 시스템은 **/mnt/local**에 마운트됩니다.



- b. 임시 디렉터리를 생성합니다.

```
RESCUE <CONTROL_PLANE_NODE>:~ # mkdir /tmp/restore
RESCUE <CONTROL_PLANE_NODE>:~ # mount -v -t nfs -o rw,noatime
<BACKUP_NODE_IP_ADDRESS>:/ceph_backups /tmp/restore/
```

- c. 컨트롤 플레인 노드에서 기존 **/var/lib/ceph** 디렉터리를 삭제합니다.

```
RESCUE <CONTROL_PLANE_NODE>:~ # rm -rf /mnt/local/var/lib/ceph/*
```

- d. 이전 Ceph 맵을 복원합니다. **<CONTROL\_PLANE\_NODE>**를 컨트롤 플레인 노드의 이름으로 바꿉니다.

```
RESCUE <CONTROL_PLANE_NODE>:~ # tar -xvC /mnt/local/ -f
/tmp/restore/<CONTROL_PLANE_NODE>/<CONTROL_PLANE_NODE>.tar.gz --xattrs -
-xattrs-include='*.*' var/lib/ceph
```

- e. 파일이 복원되었는지 확인합니다.

```
RESCUE <CONTROL_PLANE_NODE>:~ # ls -l
total 0
drwxr-xr-x 2 root 107 26 Jun 18 18:52 bootstrap-mds
drwxr-xr-x 2 root 107 26 Jun 18 18:52 bootstrap-osd
drwxr-xr-x 2 root 107 26 Jun 18 18:52 bootstrap-rbd
drwxr-xr-x 2 root 107 26 Jun 18 18:52 bootstrap-rgw
drwxr-xr-x 3 root 107 31 Jun 18 18:52 mds
drwxr-xr-x 3 root 107 31 Jun 18 18:52 mgr
drwxr-xr-x 3 root 107 31 Jun 18 18:52 mon
drwxr-xr-x 2 root 107 6 Jun 18 18:52 osd
drwxr-xr-x 3 root 107 35 Jun 18 18:52 radosgw
drwxr-xr-x 2 root 107 6 Jun 18 18:52 tmp
```

6. 노드의 전원을 끕니다.

```
RESCUE <CONTROL_PLANE_NODE> :~ # poweroff
```

7. 노드의 전원을 켭니다. 노드가 이전 상태를 재개합니다.