# Red Hat OpenStack Platform 13

## Undercloud and Control Plane Back Up and Restore

Procedures for backing up and restoring the undercloud and the overcloud control plane during updates and upgrades

Last Updated: 2021-09-29

# Red Hat OpenStack Platform 13 Undercloud and Control Plane Back Up and Restore

Procedures for backing up and restoring the undercloud and the overcloud control plane during updates and upgrades

## Legal Notice



## Abstract

This guide explains how to install and configure Relax-and-Recover (ReaR) on the undercloud and overcloud control plane nodes; how to back up the undercloud and Control Plane nodes before updates and upgrades; and, how to restore the undercloud and Control Plane nodes if an error occurs while performing updates or upgrades.

# Table of Contents

# CHAPTER 1. INTRODUCTION TO UNDERCLOUD AND CONTROL PLANE BACK UP AND RESTORE

The Undercloud and Control Plane Back Up and Restore procedure provides steps for backing up the state of the Red Hat OpenStack Platform 13 undercloud and overcloud Controller nodes, hereinafter referred to as Control Plane nodes, before updates and upgrades. Use the procedure to restore the undercloud and the overcloud Control Plane nodes to their previous state if an error occurs during an update or upgrade.

## 1.1. BACKGROUND

The Undercloud and Control Plane Back Up and Restore procedure uses the open source Relax and Recover (ReaR) disaster recovery solution, written in Bash. ReaR creates a bootable image consisting of the latest state of an undercloud or a Control Plane node. ReaR also enables a system administrator to select files for backup.

ReaR supports numerous boot media formats, including:

- ISO
- USB
- eSATA
- PXE

The examples in this document were tested using the **ISO** boot format.

ReaR can transport the boot images using multiple protocols, including:

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

For the purposes of backing up and restoring the Red Hat OpenStack Platform 13 undercloud and overcloud Control Plane nodes, the examples in this document were tested using NFS.

## 1.2. BACK UP MANAGEMENT OPTIONS

ReaR can use both internal and external back up management options.

### Internal Back Up Management

Internal back up options include:

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

### External Back Up Management

External back up management options include both open source and proprietary solutions. Open source solutions include:

- Bacula
- Bareos

Proprietary solutions include:

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

# CHAPTER 2. PREPARE THE BACKUP NODE

Before you back up the undercloud or control plane nodes, prepare the backup node to accept the backup images.

## 2.1. PREPARE THE NFS SERVER

ReaR can use multiple transport methods. Red Hat supports back up and restore with ReaR using NFS.

1. Install the NFS server on the backup node.

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

2. Add the NFS service to the firewall to ensure ports **111** and **2049** are open. For example:

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

3. Enable the NFS server and start it.

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

## 2.2. CREATING AND EXPORTING THE BACKUP DIRECTORY

To copy backup ISO images from the undercloud or Control Plane nodes to the backup node, you must create a backup directory.

### Prerequisites

- You installed and enabled the NFS server. For more information, see Preparing the NFS server.

### Procedure

1. Create the backup directory:

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

2. Export the directory. Replace **<IP_ADDRESS/24>** with the IP address and subnet mask of the network:

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

The entries in the **/etc/exports** file are in a space-delimited list. If the undercloud and the overcloud Control Plane nodes use different networks or subnets, repeat this step for each network or subnet, as shown in this example:

```
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. Restart the NFS server:

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

4. Verify that the entries are correctly configured in the NFS server:

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

# CHAPTER 3. INSTALLING AND CONFIGURING RELAX AND RECOVER (REAR)

To back up the undercloud and the overcloud control plane, you must first install and configure ReaR on the undercloud and on each control plane node.

## 3.1. INSTALLING REAR

Before you back up the undercloud and control plane, you must install the Relax and Recover (ReaR) packages and packages for generating ISO images on the undercloud node and on each control plane node.

**Procedure**

1. Install the ReaR packages and packages for generating ISO images on the undercloud node and on each control plane node:

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

2. To avoid upgrade issues in the future, delete the scheduled ReaR cron task:

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

3. Create a backup directory on the undercloud and on each control plane node:

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

4. Mount the **ctl_plane_backups** NFS directory from the backup node that runs NFS on the undercloud and on each control plane node:

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

Replace **<BACKUP_NODE_IP_ADDRESS>** with the IP address of the backup node running the NFS server.

## 3.2. CREATING THE REAR CONFIGURATION FILES

After you install Relax and Recovery (ReaR) on the undercloud node and on each control plane node, create the configuration files as the **root** user.

**Procedure**

1. Create the ReaR configuration file:

```
[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
```

- Replace **<NODE_HOSTNAME>** with the hostname of the node. For example, if the node hostname is **controller-0**, replace **<NODE_HOSTNAME>** with **controller-0**.
- Replace **<BACKUP_NODE_IP_ADDRESS>** with the IP address of the backup node that runs the NFS server. This is the IP address that you configured when you prepared the backup node. For more information, see Chapter 2, *Prepare the Backup Node*.

**IMPORTANT**

If the undercloud or control plane nodes use boot mode UEFI, you must add **USING_UEFI_BOOTLOADER=1** to the configuration file.

2. Create the **rescue.conf** file:

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

# CHAPTER 4. EXECUTING THE BACK UP PROCEDURE

Before you perform a fast forward upgrade, back up the undercloud and the overcloud control plane nodes so that you can restore them to their previous state if an error occurs.

NOTE

Before you backup the undercloud and overcloud, ensure that you do not perform any operations on the overcloud from the undercloud.

## 4.1. PERFORMING PREREQUISITE TASKS BEFORE BACKING UP THE UNDERCLOUD

Do not perform an undercloud backup when you deploy the undercloud or when you make changes to an existing undercloud.

To prevent data corruptions, confirm that there are no stack failures and ongoing tasks, and that all OpenStack services except for **mariadb** are stopped before you back up the undercloud node.

**Procedure**

1. Confirm that there are no failures on the stack. Replace **<STACKNAME>** with the name of the stack. Use the command for every stack that is deployed and available:

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

2. Verify that there are no ongoing tasks on the undercloud:

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

   If the command returns no results, there are no ongoing tasks.

3. Stop all OpenStack services on the undercloud:

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

4. Verify that **mariadb** is running:

```
# sudo systemctl status mariadb
```

## 4.2. BACKING UP THE UNDERCLOUD

To back up the undercloud node, you must log in as the root user on the undercloud node. As a precaution, you must back up the database to ensure that you can restore it.

**Prerequisites**

- You have created and exported the backup directory. For more information, see Creating and exporting the backup directory.
- You have performed prerequisite tasks before backing up the undercloud. For more information, see Performing prerequisite tasks before backing up the undercloud .
- You have installed and configured ReaR on the undercloud node. For more information, see Install and Configure ReaR.
- If you use an OVS bridge for your network interfaces, manually configure the OVS interfaces by adding the **NETWORKING_PREPARATION_COMMANDS** parameter to the **/etc/rear/local.conf** file in the following format:

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

Replace **<command_1>** and **<command_2>** with the commands that configure the network interface names or IP addresses. For example, you can add the **ip link add br-ctlplane type bridge** command to configure the control plane bridge name, or add the **ip link set eth0 up** command to set the name of the interface. You can add more commands to the parameter based on your network configuration.

**Procedure**

1. Locate the database password:

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

2. Back up the databases:

```
[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. Stop the **mariadb** database service:

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

4. Create the backup:

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

You can find the backup ISO file that you create with ReaR on the backup node under the **/ctl_plane_backups** directory.

5. Restart the undercloud:

   a. Log in to the undercloud as the stack user.

b. Restart the undercloud:

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

## 4.3. BACKING UP THE CONTROL PLANE

To back up the control plane, you must first stop the pacemaker cluster and all containers operating on the control plane nodes. Do not operate the stack to ensure state consistency. After you complete the backup procedure, start the pacemaker cluster and the containers.

As a precaution, you must back up the database to ensure that you can restore the database after you restart the pacemaker cluster and containers.

Back up the control plane nodes simultaneously.

### Prerequisites

- You have created and exported the backup directory. For more information, see Creating and exporting the backup directory.
- You have installed and configured ReaR on each control plane node. For more information, see Install and Configure ReaR.
- If you use an OVS bridge for your network interfaces, manually configure the OVS interfaces by adding the **NETWORKING_PREPARATION_COMMANDS** parameter to the **/etc/rear/local.conf** file in the following format:

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

  Replace **<command_1>** and **<command_2>** with the commands that configure the network interface names or IP addresses. For example, you can add the **ip link add br-ctlplane type bridge** command to configure the control plane bridge name, or add the **ip link set eth0 up** command to set the name of the interface. You can add more commands to the parameter based on your network configuration.

### Procedure

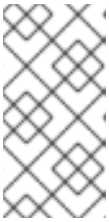
1. Locate the database password:

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

2. Back up the databases:

   ```
   [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
   ```



NOTE

Backing up the databases is a precautionary measure. This step ensures that you can manually restore the Galera cluster if it does not restore automatically as part of the restoration procedure. For more information about restoring the Galera cluster, see Troubleshooting the Galera cluster.

3. On one of the control plane nodes, stop the pacemaker cluster:

IMPORTANT

Do not operate the stack. When you stop the pacemaker cluster and the containers, this results in the temporary interruption of control plane services to Compute nodes. There is also disruption to network connectivity, Ceph, and the NFS data plane service. You cannot create instances, migrate instances, authenticate requests, or monitor the health of the cluster until the pacemaker cluster and the containers return to service following the final step of this procedure.

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

4. On each control plane node, stop the containers:

   a. Stop the containers:

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

   b. Stop the ceph-mon@controller.service container:

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

   c. Stop the ceph-mgr@controller.service container:

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

5. Optional: If you use **ganesha-nfs**, disable the file server on one controller:

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

6. Optional: If you use the ceph services **ceph-mds** and **ceph-rgw**, stop these services:

```
[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. To back up the control plane, run the control plane backup on each control plane node:

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

You can find the backup ISO file that you create with ReaR on the backup node under the **/ctl_plane_backups** directory.

NOTE

When you execute the backup command, you might see warning messages regarding the **tar** command and sockets that are ignored during the tar process, similar to the following warning:

```
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. When the backup procedure generates ISO images for each of the control plane nodes, restart the pacemaker cluster. On one of the control plane nodes, enter the following command:

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

9. On each control plane node, start the containers:

   a. Start the containers:

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

   b. Start the **ceph-mon@controller.service** container:

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

   c. Start the **ceph-mgr@controller.service** container:

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

10. Optional: If you use **ceph-mds** and **ceph-rgw**, start these services:

```
[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. Optional: If you use **ganesha-nfs**, enable the file server on one controller:

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

# CHAPTER 5. EXECUTING THE RESTORE PROCEDURE

If an error occurs during an update or upgrade, you can restore either the undercloud or overcloud control plane nodes or both so that they assume their previous state. If the Galera cluster does not restore automatically as part of the restoration procedure, you must restore the cluster manually.

You can also restore the undercloud or overcloud control plane nodes with colocated ceph monitors.

**NOTE**

When you boot from an ISO file, ensure that the NFS server is reachable by the undercloud and overcloud.

Use the following general steps:

1. Burn the bootable ISO image to a DVD or load it through ILO remote access.
2. Boot the node that requires restoration from the recovery medium.
3. Select **Recover <HOSTNAME>**. Replace **<HOSTNAME>** with the name of the node to restore.
4. Log on as user **root**.
5. Recover the backup.

## 5.1. RESTORING THE UNDERCLOUD

If an error occurs during a fast-forward upgrade, you can restore the undercloud node to its previously saved state by using the ISO image that you created using the Section 4.2, “Backing up the undercloud” procedure. The backup procedure stores the ISO images on the backup node in the folders that you created during the Section 2.2, “Creating and exporting the backup directory” step.

**Procedure**

1. Shut down the undercloud node. Ensure that the undercloud node is shutdown completely before you proceed.
2. Restore the undercloud node by booting it with the ISO image created during the backup process. The ISO image is located under the **/ctl_plane_backups** directory of the Backup node.
3. When the **Relax-and-Recover** boot menu appears, select **Recover <UNDERCLOUD_NODE>** where <UNDERCLOUD_NODE> is the name of the undercloud node.
4. Log in as user **root**.
   The following message displays:

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

   The image restore progresses quickly. When it is complete, the console echoes the following message:

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

5. When the command line interface is available, the image is restored. Switch the node off.

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

On boot up, the node resumes with its previous state.

## 5.2. RESTORING THE CONTROL PLANE

If an error occurs during a fast-forward upgrade, you can use the ISO images created using the Section 4.3, “Backing up the control plane” procedure to restore the control plane nodes to their previously saved state. To restore the control plane, you must restore all control plane nodes to the previous state to ensure state consistency.

**NOTE**

Red Hat supports backups of Red Hat OpenStack Platform with native SDNs, such as Open vSwitch (OVS) and the default Open Virtual Network (OVN). For information about third-party SDNs, refer to the third-party SDN documentation.

**Procedure**

1. Shut down each control plane node. Ensure that the control plane nodes are shut down completely before you proceed.

2. Restore the control plane nodes by booting them with the ISO image that you created during the backup process. The ISO images are located under the **/ctl_plane_backups** directory of the Backup node.

3. When the **Relax-and-Recover** boot menu appears, select **Recover <CONTROL_PLANE_NODE>**. Replace <CONTROL_PLANE_NODE> with the name of the control plane node.
The following message displays:

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

The image restore progresses quickly. When the restore completes, the console echoes the following message:

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

When the command line interface is available, the image is restored. Switch the node off.

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

Set the boot sequence to the normal boot device. On boot up, the node resumes with its previous state.

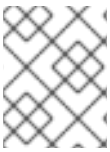
4. To ensure that the services are running correctly, check the status of pacemaker. Log in to a controller as **root** user and run the following command:

```
# pcs status
```

5. To view the status of the overcloud, use Tempest. For more information about Tempest, see Chapter 4 of the OpenStack Integration Test Suite Guide .

## 5.3. TROUBLESHOOTING THE GALERA CLUSTER

If the Galera cluster does not restore as part of the restoration procedure, you must restore Galera manually.



NOTE

In this procedure, you must perform some steps on one Controller node. Ensure that you perform these steps on the same Controller node as you go through the procedure.

**Procedure**

1. On Controller-0, retrieve the Galera cluster virtual IP:

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

2. Disable the database connections through the virtual IP on all Controller nodes:

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

3. On Controller-0, retrieve the MySQL root password:

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

4. On Controller-0, set the Galera resource to **unmanaged** mode:

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

5. Stop the MySQL containers on all Controller nodes:

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

6. Move the current directory on all Controller nodes:

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

7. Create the new directory **/var/lib/mysq** on all Controller nodes:

```
$ 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. Start the MySQL containers on all Controller nodes:

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

9. Create the MySQL database on all Controller nodes:

```
$ 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. Start the database on all Controller nodes:

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

11. Move the **.my.cnf** Galera configuration file on all Controller nodes:

```
$ 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. Reset the Galera root password on all Controller nodes:

```
$ 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. Restore the **.my.cnf** Galera configuration file inside the Galera container on all Controller nodes:

```
$ 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. On Controller-0, copy the backup database files to **/var/lib/MySQL**:

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

NOTE

The path to these files is /home/heat-admin/.

15. On Controller-0, restore the MySQL database:

```
$ 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. Shut down the databases on all Controller nodes:

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

17. On Controller-0, start the bootstrap node:

```
$ 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. Verification: On Controller-0, check the status of the cluster:

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

Ensure that the following message is displayed: “Galera cluster node is synced”, otherwise you must recreate the node.

19. On Controller-0, retrieve the cluster address from the configuration:

```
$ 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. On each of the remaining Controller nodes, start the database and validate the cluster:

    a. Start the database:

```
$ 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. Check the status of the MYSQL cluster:

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

Ensure that the following message is displayed: “Galera cluster node is synced”, otherwise you must recreate the node.

21. Stop the MySQL container on all Controller nodes:

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

22. On all Controller nodes, remove the following firewall rule to allow database connections through the virtual IP address:

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

23. Restart the MySQL container on all Controller nodes:

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

24. Restart the **clustercheck** container on all Controller nodes:

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

25. On Controller-0, set the Galera resource to **managed** mode:

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

## 5.4. RESTORING THE UNDERCLOUD AND CONTROL PLANE NODES WITH COLOCATED CEPH MONITORS

If an error occurs during an update or upgrade, you can use ReaR backups to restore either the undercloud or overcloud control plane nodes, or both, to their previous state.

### Prerequisites

- Install and configure ReaR. For more information, see Install and configure ReaR.
- Prepare the backup node. For more information, see Prepare the backup node.
- Execute the backup procedure. For more information, see Execute the backup procedure.

### Procedure

1. On the backup node, export the NFS directory to host the Ceph backups. Replace **<IP_ADDRESS/24>** with the IP address and subnet mask of the network:

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

2. On the undercloud node, source the undercloud credentials and run the following script:

```
# 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
```

To verify that the **ceph-mgr@controller.service** container has stopped, enter the following command:

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

3. On the undercloud node, source the undercloud credentials and run the following script:

```
# 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. On the node that you want to restore, complete the following tasks:

   a. Power off the node before you proceed.

   b. Restore the node with the ReaR backup file that you have created during the backup process. The file is located in the **/ceph_backups** directory of the backup node.

   c. From the **Relax-and-Recover** boot menu, select **Recover <CONTROL_PLANE_NODE>**, where **<CONTROL_PLANE_NODE>** is the name of the control plane node.

   d. At the prompt, enter the following command:

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

      When the image restoration process completes, the console displays the following message:

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

5. For the node that you want to restore, copy the Ceph backup from the **/ceph_backups** directory into the **/var/lib/ceph** directory:

   a. Identify the system mount points:

   ```
   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
   ```

   The **/dev/vda2** file system is mounted on **/mnt/local**.

   b. Create a temporary directory:

   ```
   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. On the control plane node, remove the existing **/var/lib/ceph** directory:

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

   d. Restore the previous Ceph maps. Replace **<CONTROL_PLANE_NODE>** with the name of your control plane node:

   ```
   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. Verify that the files are restored:

   ```
   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. Power off the node:

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

7. Power on the node. The node resumes its previous state.