Red Hat OpenStack Platform 16.2

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
Red Hat OpenStack Platform 16.2 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
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

MAKING OPEN SOURCE MORE INCLUSIVE ................................................................. 3

PROVIDING FEEDBACK ON RED HAT DOCUMENTATION ........................................ 4

CHAPTER 1. INTRODUCTION TO UNDERCLOUD AND CONTROL PLANE BACK UP AND RESTORE ........ 5
  1.1. ABOUT THE REAR DISASTER RECOVERY SOLUTION ........................................ 5
  1.2. REAR BACKUP MANAGEMENT OPTIONS .................................................. 5

CHAPTER 2. CONFIGURING THE BACKUP NODE .................................................... 7

CHAPTER 3. INSTALLING REAR ON THE UNDERCLOUD AND CONTROL PLANE NODES ............ 8
  3.1. INSTALLING REAR ON THE UNDERCLOUD NODE .......................................... 8
  3.2. INSTALLING REAR ON THE CONTROL PLANE NODES .................................... 9

CHAPTER 4. CREATING A BACKUP OF THE UNDERCLOUD AND CONTROL PLANE NODES ........ 10
  4.1. CREATING A DATABASE BACKUP OF THE UNDERCLOUD NODE DURING AN UPGRADE FROM 13 TO 16.2 .............................................................. 10
  4.2. CREATING A BACKUP OF THE UNDERCLOUD NODE .................................... 11
  4.3. CREATING A BACKUP OF THE CONTROL PLANE NODES ................................ 12

CHAPTER 5. RESTORING THE UNDERCLOUD AND CONTROL PLANE NODES .................... 14
  5.1. RESTORING THE UNDERCLOUD NODE ..................................................... 14
  5.2. RESTORING THE CONTROL PLANE NODES ................................................ 15
  5.3. RESTORING THE UNDERCLOUD AND CONTROL PLANE NODES WITH COLOCATED CEPH MONITORS ........................................................................ 17
  5.4. RESTORING THE GALERA CLUSTER MANUALLY ........................................... 19

CHAPTER 6. SCHEDULING CONTROL PLANE NODE BACKUPS WITH CRON ...................... 24
MAKING OPEN SOURCE MORE INCLUSIVE

Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, see our CTO Chris Wright’s message.
PROVIDING FEEDBACK ON RED HAT DOCUMENTATION

We appreciate your input on our documentation. Tell us how we can make it better.

Using the Direct Documentation Feedback (DDF) function

Use the Add Feedback DDF function for direct comments on specific sentences, paragraphs, or code blocks.

1. View the documentation in the Multi-page HTML format.
2. Ensure that you see the Feedback button in the upper right corner of the document.
3. Highlight the part of text that you want to comment on.
4. Click Add Feedback.
5. Complete the Add Feedback field with your comments.
6. Optional: Add your email address so that the documentation team can contact you for clarification on your issue.
7. Click Submit.
CHAPTER 1. INTRODUCTION TO UNDERCLOUD AND CONTROL PLANE BACK UP AND RESTORE

You must back up the state of the Red Hat OpenStack Platform 16.2 undercloud and overcloud controller nodes, also known as control plane nodes, before updates and upgrades. You can use the created backups to restore the undercloud and overcloud control plane nodes to their previous state if an error occurs during an update or upgrade.

1.1. ABOUT THE REAR DISASTER RECOVERY SOLUTION

The tasks described in the Undercloud and Control Plane Back Up and Restore guide use the open source Relax and Recover (ReaR) disaster recovery solution that is written in Bash. You can use ReaR to create bootable images of the latest state of the undercloud or control plane nodes, or to back up specific files.

ReaR supports the following boot media formats:

- ISO
- USB
- eSATA
- PXE

The examples in this document were tested using the ISO bootable files format.

ReaR can use the following protocols to transport files:

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

For the purposes of backing up and restoring the Red Hat OpenStack Platform 16.2 undercloud and overcloud control plane nodes, the examples in this document were tested using NFS.

1.2. REAR BACKUP MANAGEMENT OPTIONS

You can use ReaR with internal and external backup management options.

**Internal backup management**

You can use ReaR with the following internal backup options:

- tar
- rsync

**External backup management**
External backup management options include open source and proprietary solutions. You can use ReaR with the following open source solutions:

- Bacula
- Bareos

You can use ReaR with the following proprietary solutions:

- EMC NetWorker (Legato)
- HP DataProtector
- IBM Tivoli Storage Manager (TSM)
- Symantec NetBackup
CHAPTER 2. CONFIGURING THE BACKUP NODE

Before you can create a backup of the undercloud or control plane nodes, you must configure the backup node. To store backup files, you can use your existing NFS server or you can use the Command Line Interface (CLI) tool to provision and configure an NFS server. To install and configure an NFS server on the backup node, use the `backup-and-restore` Ansible role.

NOTE

If you previously installed and configured your NFS server, you do not need to complete this procedure. By default, the Rest and Recover (ReaR) configuration assumes that the IP address of the NFS server is 192.168.24.1. If your NFS server has a different IP address, you must add the parameter `tripleo_backup_and_restore_nfs_server` to the setup ReaR command. For more information, see Installing ReaR on the control plane nodes.

Procedure

1. On the undercloud node, source the undercloud credentials:

```
[stack@undercloud-0 ~]$ source stackrc
(undercloud) [stack@undercloud ~]$
```

2. On the undercloud node, create an inventory file for the backup node:

   ```
   (undercloud) [stack@undercloud ~]$ cat <<'EOF'> ~/nfs-inventory.yaml
   [BackupNode]
   serverX ansible_host=<ip_address> ansible_user=<user>
   EOF
   ```

   Replace `<ip_address>` and `<user>` with the values that apply to your environment.

3. Copy the public SSH key from the undercloud node to the backup node:

   ```
   (undercloud) [stack@undercloud ~]$ ssh-copy-id -i ~/.ssh/id_rsa.pub <BackupNode>
   ```

4. Copy the public SSH key from the undercloud node to the backup node.

   ```
   (undercloud) [stack@undercloud ~]$ ssh-copy-id -i ~/.ssh/id_rsa.pub <BackupNode>
   ```

5. Configure the backup node:

   ```
   (undercloud) [stack@undercloud ~]$ openstack undercloud backup --setup-nfs --extra-vars /home/stack/bar-vars.yaml --inventory /home/stack/nfs-inventory.yaml
   ```

- By default, the `tripleo_backup_and_restore_nfs_server` parameter is set to `192.168.24.1` in the `bar-vars.yaml` environment file. If you installed and configured your own NFS server, add the `tripleo_backup_and_restore_nfs_server` parameter to `bar-vars.yaml` and set the value to the IP address of your NFS server.

- By default, the `tripleo_backup_and_restore_nfs_clients_nets` parameter is set to `['192.168.24.0/24', '10.0.0.0/24', '172.16.0.0/24']` in the `bar-vars.yaml` environment file. Ensure that you add the network ranges where your undercloud and overcloud are deployed so that the NFS backup directory is exported to them.
CHAPTER 3. INSTALLING REAR ON THE UNDERCLOUD AND CONTROL PLANE NODES

Before you create a backup of the undercloud and control plane nodes, you must install the Relax and Recover (ReaR) packages on the nodes that you want to back up.

To install ReaR, complete the following procedures:

1. Section 3.1, “Installing ReaR on the undercloud node”
2. Section 3.2, “Installing ReaR on the control plane nodes”

3.1. INSTALLING REAR ON THE UNDERCLOUD NODE

To create a backup of the undercloud node, you must install and configure Relax and Recover (ReaR) on the undercloud.

Prerequisites

- You have configured the backup node. For more information, see Configuring the backup node.

Procedure

1. On the undercloud node, source the undercloud credentials:

   ```bash
   [stack@undercloud-0 ~]$ source stackrc
   
   If you use a custom stack name, add the `--stack <stack_name>` option to the `tripleo-ansible-inventory` command.
   ```

2. If you have not done so before, create an inventory file and use the `tripleo-ansible-inventory` command to generate a static inventory file that contains hosts and variables for all the overcloud nodes:

   ```bash
   (undercloud) [stack@undercloud ~]$ tripleo-ansible-inventory \ 
   --ansible_ssh_user heat-admin \ 
   --static-yaml-inventory /home/stack/tripleo-inventory.yaml
   
   (undercloud) [stack@undercloud ~]$ openstack undercloud backup --setup-rear --extra-vars /home/stack/bar-vars.yaml --inventory /home/stack/tripleo-inventory.yaml
   ```

3. Install ReaR on the undercloud:

   ```bash
   (undercloud) [stack@undercloud ~]$ openstack undercloud backup --setup-rear --extra-vars /home/stack/bar-vars.yaml --inventory /home/stack/tripleo-inventory.yaml
   ```

4. If your system uses the UEFI boot loader, perform the following steps on the undercloud node:
   a. Install the following tools:

   ```bash
   ```
3.2. INSTALLING REAR ON THE CONTROL PLANE NODES

To create a backup of the overcloud control plane, you must install and configure Relax and Recover (ReaR) on each of the control plane nodes.

Prerequisites

- You have configured the backup node. For more information, see Configuring the backup node.

Procedure

1. On the undercloud node, source the undercloud credentials:

   ```bash
   [stack@undercloud-0 ~]$ source stackrc
   ```

2. If you have not done so before, create an inventory file and use the `tripleo-ansible-inventory` command to generate a static inventory file that contains hosts and variables for all the overcloud nodes:

   ```bash
   (undercloud) [stack@undercloud ~]$ tripleo-ansible-inventory --ansible_ssh_user heat-admin --static-yaml-inventory /home/stack/tripleo-inventory.yaml
   ```

3. Install ReaR on the control plane nodes:

   ```bash
   (undercloud) [stack@undercloud ~]$ openstack overcloud backup --setup-rear --extra-vars /home/stack/bar-vars.yaml --inventory /home/stack/tripleo-inventory.yaml
   ```

4. If your system uses the UEFI boot loader, perform the following steps on the control plane nodes:

   a. Install the following tools:

      ```bash
      $ sudo dnf install dosfstools efibootmgr
      ```

   b. Enable UEFI backup in the ReaR configuration file located in `/etc/rear/local.conf` by replacing the `USING_UEFI_BOOTLOADER` parameter value 0 with the value 1.
CHAPTER 4. CREATING A BACKUP OF THE UNDERCLOUD AND CONTROL PLANE NODES

To create a backup of the undercloud and control plane nodes using the `backup-and-restore` Ansible role, complete the following procedures:

1. Section 4.2, “Creating a backup of the undercloud node”
2. Section 4.3, “Creating a backup of the control plane nodes”

To create a backup of the database that runs on the undercloud node and use that backup to recover the state of the database in the event that it becomes corrupted, complete the following procedure:

- Section 4.1, “Creating a database backup of the undercloud node during an upgrade from 13 to 16.2”

4.1. CREATING A DATABASE BACKUP OF THE UNDERCLOUD NODE DURING AN UPGRADE FROM 13 TO 16.2

If you are upgrading your Red Hat OpenStack Platform environment from 13 to 16.2, you must create a separate database backup after you perform the undercloud upgrade and before you perform the Leapp upgrade process on the overcloud nodes.

**NOTE**

When you create a full backup of the undercloud node, the overcloud backup includes a database backup of the undercloud node. Therefore, you do not need to create a separate database backup.

**Procedure**

1. Log in to the undercloud as the `stack` user.

2. Source the `stackrc` file.

   ```bash
   (undercloud) [stack@undercloud ~]$ source ~/stackrc
   ```

3. On the undercloud node, create the following Ansible playbook:

   ```yaml
   (undercloud) [stack@undercloud ~]$ cat <<'EOF' >
   ~/bar_create_undercloud_db_backup.yaml
   # Playbook
   # DB backup on the undercloud node.
   - become: true
     hosts: undercloud
     tasks:
       - name: Create DB Backup for the undercloud
         include_role:
           name: backup-and-restore
           tasks_from: ../backup/tasks/db_backup.yml
           vars:
             tripleo_backup_and_restore_service_manager: false
   EOF
   ```
4. Create a database backup of the undercloud node:

```
(undercloud) [stack@undercloud ~]$ ansible-playbook \
-v -i ~/tripleo-inventory.yaml \
--extra="ansible_ssh_common_args=-o StrictHostKeyChecking=no" \
--become \
--become-user root \
~/bar_create_undercloud_db_backup.yaml
```

Additional resources

- Framework for Upgrades (13 to 16.2)
- Section 4.2, “Creating a backup of the undercloud node”

### 4.2. CREATING A BACKUP OF THE UNDERCLOUD NODE

To create a backup of the undercloud node, use the `openstack undercloud backup` command. The backup of the undercloud node includes the backup of the database that runs on the undercloud node. You can use the database backup to recover the state of the database in the event that it becomes corrupted and does not restore automatically as a part of the restoration procedure.

**NOTE**

- If you are upgrading your Red Hat OpenStack Platform environment from 13 to 16.2, you must create a separate database backup after you perform the undercloud upgrade and before you perform the Leapp upgrade process on the overcloud nodes. For more information, see Creating a database backup of the undercloud node.

- The backup process does not mount the EFI partition by default. To ensure that you can back up and restore EFI partition, you must manually mount it on the undercloud node. For more information, see the Knowledge Base article How to backup/restore an openstack system installed with UEFI - error Empty string passed to get_device_name during recovery.

**Prerequisites**

- You have configured the backup node. For more information, see Configuring the backup node.

- You have installed ReaR on the undercloud node. For more information, see Installing ReaR on the undercloud node.

- 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>` with the commands to 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. Add more commands to the parameter based on your network configuration.
Procedure

1. On the undercloud node, source the undercloud credentials:

```
[stack@undercloud-0 ~]$ source stackrc
```

2. If your system uses the UEFI boot loader, mount the EFI partition:

```
(undercloud) [stack@undercloud ~]$ mkdir /boot/efi_stack
(undercloud) [stack@undercloud ~]$ mount <EFI_partition> /boot/efi_stack
(undercloud) [stack@undercloud ~]$ dd if=<openstack_partition> of=/boot/<partition_name>
```

3. If you have not done so before, create an inventory file and use the `tripleo-ansible-inventory` command to generate a static inventory file that contains hosts and variables for all the overcloud nodes:

```
(undercloud) [stack@undercloud ~]$ tripleo-ansible-inventory \
--ansible_ssh_user heat-admin \ 
--static-yaml-inventory /home/stack/tripleo-inventory.yaml
```

4. Create a backup of the undercloud node:

```
(undercloud) [stack@undercloud ~]$ openstack undercloud backup --inventory 
/home/stack/tripleo-inventory.yaml
```

4.3. CREATING A BACKUP OF THE CONTROL PLANE NODES

To create a backup of the undercloud node, use the `openstack undercloud backup` command. The backup of the undercloud node includes the backup of the database that runs on the undercloud node. You can use the database backup to recover the state of the database in the event that it becomes corrupted and does not restore automatically as a part of the restoration procedure.

**NOTE**

The backup process does not mount the EFI partition by default. To ensure that you can back up and restore EFI partition, you must manually mount it on each of the control plane nodes. For more information, see the Knowledge Base article [How to backup/restore an openstack system installed with UEFI - error Empty string passed to get_device_name during recovery](https://access.redhat.com/kb/docs/article/382).

Prerequisites

- You have configured the backup node. For more information, see Configuring the backup node.

- You have installed ReaR on the control plane nodes. For more information, see Installing ReaR on the control plane nodes.

- 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>' '...')
  ```
Procedure

1. On the undercloud node, source the undercloud credentials:

   [stack@undercloud-0 ~]$ source stackrc

2. If your system uses the UEFI boot loader, mount the EFI partition on each one of the controllers:

   [heat-admin@controller-x ~]$ mkdir /boot/efi_stack
   [heat-admin@controller-x ~]$ mount <EFI_partition> /boot/efi_stack
   [heat-admin@controller-x ~]$ dd if=<openstack_partition> of=/boot/<partition_name>

3. If you have not done so before, use the tripleo-ansible-inventory command to generate a static inventory file that contains hosts and variables for all the overcloud nodes:

   (undercloud) [stack@undercloud ~]$ tripleo-ansible-inventory
   --ansible_ssh_user heat-admin
   --static-yaml-inventory /home/stack/tripleo-inventory.yaml

4. Create a backup of the control plane nodes:

   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.

   (undercloud) [stack@undercloud ~]$ openstack overcloud backup --inventory
   /home/stack/tripleo-inventory.yaml

   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.
CHAPTER 5. RESTORING THE UNDERCLOUD AND CONTROL PLANE NODES

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

To restore the undercloud and control plane nodes using backups, complete the following procedures that are applicable to your deployment:

1. Section 5.1, "Restoring the undercloud node"
2. Section 5.2, "Restoring the control plane nodes"
3. Section 5.3, "Restoring the undercloud and control plane nodes with colocated Ceph monitors"
4. Section 5.4, "Restoring the Galera cluster manually"

5.1. RESTORING THE UNDERCLOUD NODE

If an error occurs during an update or upgrade, you can restore the undercloud node to its previous state using the backup ISO image that you created using ReaR. You can find the backup ISO images on the backup node. Burn the bootable ISO image to a DVD or download it to the undercloud node through Integrated Lights-Out (iLO) remote access.

Prerequisites

- You have created a backup of the undercloud node. For more information, see Creating a backup of the undercloud node.
- You have access to the backup node.
- If you use an OVS bridge for your network interfaces, ensure that you have the network configuration information that you set in the NETWORKING_PREPARATION_COMMANDS parameter when you created the backup of the undercloud node. The restoration process uses this information to restore the network interfaces.

Procedure

1. Power off the undercloud node. Ensure that the undercloud node is powered off completely before you proceed.
2. Boot the undercloud node with the backup ISO image.
3. When the Relax-and-Recover boot menu displays, select Recover <UNDERCLOUD_NODE> where <UNDERCLOUD_NODE> is the name of your undercloud node.

   **NOTE**

   If your system uses UEFI, select the Relax-and-Recover (no Secure Boot) option.

4. Log in as the root user and restore the node:

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

When the undercloud node restoration process completes, the console displays the following message:

Finished recovering your system
Exiting rear recover
Running exit tasks

5. If your system uses UEFI, manually restore the EFI partition and boot entry when the command line console is available:

# once completed, restore the second partition (which is ISO9660)
RESCUE <UNDERCLOUD_NODE>:~ $ dd if=/mnt/local/boot/<partition_name> of=<openstack_partition>
# restore the EFI boot entry (the restored one points to /boot/EFI/redhat/grubx64.efi, which is not on a vfat partition
RESCUE <UNDERCLOUD_NODE>:~ $ efibootmgr --create --gpt --disk <uefi disk> --part 1 --write-signature --label "red" --loader "\EFI\red\grubx64.efi"

6. Power off the node:

RESCUE <UNDERCLOUD_NODE>:~ # poweroff

On boot up, the node resumes its previous state.

5.2. RESTORING THE CONTROL PLANE NODES

If an error occurs during an update or upgrade, you can restore the control plane nodes to their previous state using the backup ISO image that you have created using ReaR.

To restore the control plane, you must restore all control plane nodes to ensure state consistency.

You can find the backup ISO images on the backup node. Burn the bootable ISO image to a DVD or download it to the undercloud node through Integrated Lights-Out (iLO) remote access.

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.

Prerequisites

- You have created a backup of the control plane nodes. For more information, see Creating a backup of the control plane nodes.
- You have access to the backup node.
- If you use an OVS bridge for your network interfaces, ensure that you have the network configuration information that you set in the networking_preparation_commands parameter when you created the backup of the control plane nodes. The restoration process uses this information to restore the network interfaces.
Procedure

1. Power off each control plane node. Ensure that the control plane nodes are powered off completely before you proceed.

2. Boot each control plane node with the corresponding backup ISO image.

3. When the Relax-and-Recover boot menu displays, on each control plane node, select Recover `<CONTROL PLANE NODE>`. Replace `<CONTROL PLANE NODE>` with the name of the corresponding control plane node.

   **NOTE**

   If your system uses UEFI, select the Relax-and-Recover (no Secure Boot) option.

4. On each control plane node, log in as the root user and restore the node:
   The following message displays:

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

   When the control plane node restoration process completes, the console displays the following message:

   Finished recovering your system
   Exiting rear recover
   Running exit tasks

5. If your system uses UEFI, manually restore the EFI partition and boot entry when the command line console is available:

   # once completed, restore the second partition (which is ISO9660)
   RESCUE <CONTROL PLANE NODE>:~ $ dd if=/mnt/local/boot/<partition_name> of=<<openstack_partition>

   # restore the EFI boot entry (the restored one points to /boot/EFI/redhat/grubx64.efi, which is not on a vfat partition
   RESCUE <CONTROL PLANE NODE>:~ $ efibootmgr --create --gpt --disk <uefi disk> --part 1 --write-signature --label "red" --loader "EFI\red\grubx64.efi"

6. Power off the node:

   RESCUE <CONTROL PLANE NODE>:~ # poweroff

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

8. To ensure that the services are running correctly, check the status of pacemaker. Log in to a Controller node as the root user and enter the following command:

   # pcs status
5.3. 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 the undercloud or overcloud control plane nodes with colocated Ceph monitors to their previous state.

Prerequisites

- Install ReaR on the undercloud and control plane nodes. For more information, see Installing ReaR on the undercloud and control plane nodes.
- Configure the backup node. For more information, see Configuring the backup node.
- Create a backup of the undercloud and control plane nodes. For more information, see Creating a backup of the undercloud and control plane nodes.

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:

   ```bash
   [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:

   ```bash
   (undercloud) [stack@undercloud ~]$ source stackrc
   ```

   ```bash
   #!/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-mgr@$(hostname -s) ceph-mon@$(hostname -s)'; done
   ```

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

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

3. On the undercloud node, source the undercloud credentials and run the following script. Replace `<BACKUP_NODE_IP_ADDRESS>` with the IP address of the backup node:

   ```bash
   (undercloud) [stack@undercloud ~]$ source stackrc
   ```

   ```bash
   #!/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
   ```

   ```bash
   #!/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> and replace <control_plane_node> with 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  8.4M   16G   5% /run
tmpfs            16G  8.4M   16G   5% /run
/dev/vda2        30G  13G   13G  41% /mnt/local
   ```
   The /dev/vda2 file system is mounted on /mnt/local.
   b. Create a temporary directory:
RESOLVE <CONTROL_PLANE_NODE>:~ # mkdir /tmp/restore
RESOLVE <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:

RESOLVE <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:

RESOLVE <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:

RESOLVE <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:

    RESOLVE <CONTROL_PLANE_NODE>:~ # poweroff

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

5.4. RESTORING THE GALERA CLUSTER MANUALLY

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 podman container stop $(sudo podman 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/mysql** 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 podman container start $(sudo podman container ls --all --format "{{ .Names }}" --filter=name=galera-bundle)

9. Create the MySQL database on all Controller nodes:

   $ sudo podman exec -i $(sudo podman container ls --all --format "{{ .Names }}" --filter=name=galera-bundle) bash -c "mysqld_safe --skip-networking --wsrep-on=OFF --log-error=/var/log/mysql/mysql_safe.log" &

   The output might include error messages about **auth_ed25519.so**, which are not critical. You can ignore these error messages.

10. Start the database on all Controller nodes:

    $ sudo podman exec $(sudo podman container ls --all --format "{{ .Names }}" --filter=name=galera-bundle) bash -c "mysqld_safe --skip-networking --wsrep-on=OFF --log-error=/var/log/mysql/mysql_safe.log" &

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

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

13. Restore the .my.cnf Galera configuration file inside the Galera container on all Controller nodes:

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

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

16. Shut down the databases on all Controller nodes:

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

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

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

   b. Check the status of the MYSQL cluster:

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

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

   ```
   $ sudo podman container restart $(sudo podman 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
   ```
Verification

1. To ensure that services are running correctly, check the status of pacemaker:

   $ sudo pcs status

2. To view the status of the overcloud, use Tempest. For more information about Tempest, see Configuring the Integration Test Suite in the OpenStack Integration Test Suite Guide.

3. If you suspect an issue with a particular node, check the state of the cluster with `clustercheck`:

   $ sudo podman exec clustercheck /usr/bin/clustercheck
CHAPTER 6. SCHEDULING CONTROL PLANE NODE BACKUPS WITH CRON

IMPORTANT

This feature is available in this release as a Technology Preview, and therefore is not fully supported by Red Hat. It should only be used for testing, and should not be deployed in a production environment. For more information about Technology Preview features, see Scope of Coverage Details.

You can configure a cron job to create backups of the control plane nodes with ReaR using the Ansible `backup-and-restore` role. You can view the logs in the `/var/log/rear-cron` directory.

Prerequisites

- You have installed ReaR on the undercloud and control plane nodes. For more information, see Installing ReaR on the undercloud and control plane nodes.
- You have configured the backup node. For more information, see Configuring the backup node.
- You have sufficient available disk space at your backup location to store the backup.

Procedure

1. On the undercloud node, enter the following command to create the backup script:

   ```bash
   [stack@undercloud ~]$ cat <<'EOF' > /home/stack/execute-rear-cron.sh
   #!/bin/bash
   OWNER="stack"
   TODAY=`date +%Y%m%d`
   FILE="/var/log/rear-cron.${TODAY}"
   sudo touch ${FILE}
   sudo chown ${OWNER}:${OWNER} ${FILE}
   CURRENTTIME=`date`
   echo "[$CURRENTTIME] rear start" >> ${FILE}
   /usr/bin/ansible-playbook -v -i /home/stack/tripleo-inventory.yaml --
   extra="ansible_ssh_common_args='-o StrictHostKeyChecking=no'" --become --become-user root --tags bar_create_recover_image --
   extra="tripleo_backup_and_restore_service_manager=false" /home/stack/bar_rear_create_restore_images.yaml 2>&1 >> ${FILE}
   CURRENTTIME=`date`
   echo "[$CURRENTTIME] rear end" >> ${FILE}
   EOF
   
   [stack@undercloud ~]$ chmod 755 /home/stack/execute-rear-cron.sh
   ```

2. Set executable privileges for the `/home/stack/execute-rear-cron.sh` script:

   ```bash
   [stack@undercloud ~]$ chmod 755 /home/stack/execute-rear-cron.sh
   ```

3. Edit the crontab file with the `crontab -e` command and use an editor of your choice to add the following cron job. Ensure you save the changes to the file:
The `/home/stack/execute-rear-cron.sh` script is scheduled to be executed by the stack user at midnight.

4. To verify that the cron job is scheduled, enter the following command:

   ```bash
   [stack@undercloud ~]$ crontab -l
   ```

   The command output displays the scheduled cron jobs:

   ```
   0 0 * * * /home/stack/execute-rear-cron.sh
   ```