Red Hat OpenStack Platform 16.1

Backing up and restoring the undercloud and control plane nodes

Creating and restoring backups of the undercloud and the overcloud control plane nodes

Last Updated: 2023-08-31

Red Hat OpenStack Platform 16.1 Backing up and restoring the undercloud and control plane nodes

Creating and restoring backups of the undercloud and the overcloud control plane nodes

Legal Notice

Abstract

This guide explains how to create and restore backups of the undercloud and control plane nodes, and how to troubleshoot backup and restore problems. Backups are required when you upgrade or update Red Hat OpenStack Platform. You can also optionally create periodic backups of your environment to minimize downtime in case of issues.

Table of Contents

MAKING OPEN SOURCE MORE INCLUSIVE 3

PROVIDING FEEDBACK ON RED HAT DOCUMENTATION 4

CHAPTER 1. BACKING UP THE UNDERCLOUD NODE 5
1.1. SUPPORTED BACKUP FORMATS AND PROTOCOLS 5
1.2. INSTALLING AND CONFIGURING AN NFS SERVER ON THE BACKUP NODE 5
1.3. INSTALLING REAR ON THE UNDERCLOUD NODE 6
1.4. CONFIGURING OPEN VSWITCH (OVS) INTERFACES FOR BACKUP 8
1.5. CREATING A BACKUP OF THE UNDERCLOUD NODE 8

CHAPTER 2. BACKING UP THE CONTROL PLANE NODES 10
2.1. SUPPORTED BACKUP FORMATS AND PROTOCOLS 10
2.2. INSTALLING AND CONFIGURING AN NFS SERVER ON THE BACKUP NODE 10
2.3. INSTALLING REAR ON THE CONTROL PLANE NODES 11
2.4. CONFIGURING OPEN VSWITCH (OVS) INTERFACES FOR BACKUP 13
2.5. CREATING A BACKUP OF THE CONTROL PLANE NODES 13
2.6. SCHEDULING CONTROL PLANE NODE BACKUPS WITH CRON 14

CHAPTER 3. BACKING UP CONTROL PLANE NODES THAT USE COMPOSABLE ROLES 17
3.1. SUPPORTED BACKUP FORMATS AND PROTOCOLS 17
3.2. INSTALLING AND CONFIGURING AN NFS SERVER ON THE BACKUP NODE 17
3.3. INSTALLING REAR ON THE CONTROL PLANE NODES 18
3.4. CONFIGURING OPEN VSWITCH (OVS) INTERFACES FOR BACKUP 20
3.5. CREATING A BACKUP OF CONTROL PLANE NODES THAT USE COMPOSABLE ROLES 20
3.6. SCHEDULING CONTROL PLANE NODE BACKUPS WITH CRON 23
3.7. ADDITIONAL RESOURCES 24

CHAPTER 4. RESTORING THE UNDERCLOUD AND CONTROL PLANE NODES 25
4.1. PREPARING A CONTROL PLANE WITH COLOCATED CEPH MONITORS FOR THE RESTORE PROCESS 25
4.2. RESTORING THE UNDERCLOUD NODE 26
4.3. RESTORING THE CONTROL PLANE NODES 27
4.4. RESTORING THE GALERA CLUSTER MANUALLY 29
4.5. RESTORING THE UNDERCLOUD NODE DATABASE MANUALLY 32

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. BACKING UP THE UNDERCLOUD NODE

To back up the undercloud node, you configure the backup node, install the Relax-and-Recover tool on the undercloud node, and create the backup image. You can create backups as a part of your regular environment maintenance.

In addition, you must back up the undercloud node before performing updates or upgrades. You can use the backups to restore the undercloud node to its previous state if an error occurs during an update or upgrade.

1.1. SUPPORTED BACKUP FORMATS AND PROTOCOLS

The undercloud and backup and restore process uses the open-source tool Relax-and-Recover (ReaR) to create and restore bootable backup images. ReaR is written in Bash and supports multiple image formats and multiple transport protocols.

The following list shows the backup formats and protocols that Red Hat OpenStack Platform supports.

Bootable media formats

- ISO

File transport protocols

- SFTP
- NFS

1.2. INSTALLING AND CONFIGURING AN NFS SERVER ON THE BACKUP NODE

You can install and configure a new NFS server to store the backup file. To install and configure an NFS server on the backup node, create an inventory file, set up an SSH key, and run the **openstack undercloud backup** command with the NFS server options.

IMPORTANT

- If you previously installed and configured an NFS or SFTP server, you do not need to complete this procedure. You enter the server information when you set up ReaR on the node that you want to back up.
- By default, the Relax 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, add the parameter **tripleo_backup_and_restore_server** to the setup ReaR command.

Procedure

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

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

2. On the undercloud node, create an inventory file for the backup node and replace the **<ip_address>** and **<user>** with the values that apply to your environment:

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

3. On the undercloud node, create the following Ansible playbook and replace **<backup_node>** with the host name of the backup node:

```
(undercloud) [stack@undercloud ~]$ cat <<'EOF' > ~/bar_nfs_setup.yaml
# Playbook
# Substitute <backup_node> with the host name of your backup node.
- become: true
  hosts: <backup_node>
  name: Setup NFS server for ReaR
  roles:
  - role: backup-and-restore
EOF
```

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 <backup_node>
```

   Replace **<backup_node>** with the path and name of the backup node.

5. On the undercloud node, enter the following **ansible-playbook** commands to configure the backup node:

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

1.3. INSTALLING REAR ON THE UNDERCLOUD NODE

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

Prerequisites

- You have an NFS or SFTP server installed and configured on the backup node. For more information about creating a new NFS server, see Section 1.2, “Installing and configuring an NFS server on the backup node”.

Procedure

1. On the undercloud node, source the undercloud credentials and use the **tripleo-ansible-inventory** command to generate a static inventory file that contains hosts and variables for all the overcloud nodes:

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

If you use a custom stack name, add the **--stack <stack_name>** option to the **tripleo-ansible-inventory** command.

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

```
(undercloud) [stack@undercloud ~]$ cat <<'EOF' > ~/bar_rear_setup-undercloud.yaml
# Playbook
# Installing and configuring ReaR on the undercloud node
- become: true
  hosts: undercloud
  name: Install ReaR
  roles:
  - role: backup-and-restore
EOF
```

3. Choose one of the following options:

   a. If you use NFS, enter the following Ansible command to install ReaR on 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 \
    -e tripleo_backup_and_restore_server=<nfs-ip> \
    --tags bar_setup_rear \
    ~/bar_rear_setup-undercloud.yaml
```

   b. If you use SFTP, enter the following Ansible command to install ReaR on 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 \
    -e tripleo_backup_and_restore_output_url=sftp://<user>:
<password>@<backup_node_ip>/ \
    -e tripleo_backup_and_restore_backup_url=iso:///backup/ \
    --tags bar_setup_rear \
    ~/bar_rear_setup-undercloud.yaml
```

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

   a. Install the following tools:

```
$ 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**.

1.4. CONFIGURING OPEN VSWITCH (OVS) INTERFACES FOR BACKUP

If you use an Open vSwitch (OVS) bridge in your environment, you must manually configure the OVS interfaces before you create a backup of the undercloud or control plane nodes. The restoration process uses this information to restore the network interfaces.

Procedure

- In the **/etc/rear/local.conf** file, add the **NETWORKING_PREPARATION_COMMANDS** parameter in the following format:

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

Replace **<command_1>** and **<command_2>** with 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.

1.5. CREATING A BACKUP OF THE UNDERCLOUD NODE

To create a backup of the undercloud node, use the **backup-and-restore** Ansible role. You can then use the backup to restore the undercloud node to its previous state in case the node becomes corrupted or inaccessible. The backup of the undercloud node includes the backup of the database that runs on the undercloud node.

Prerequisites

- You have an NFS or SFTP server installed and configured on the backup node. For more information about creating a new NFS server, see Section 1.2, “Installing and configuring an NFS server on the backup node”.
- You have installed ReaR on the undercloud node. For more information, see Section 1.3, “Installing ReaR on the undercloud node”.
- If you use an OVS bridge for your network interfaces, you have configured the OVS interfaces. For more information, see Section 1.4, “Configuring Open vSwitch (OVS) interfaces for backup” .

Procedure

1. Log in to the undercloud as the **stack** user.
2. Retrieve the MySQL root password:

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

3. Create a database backup of the undercloud node:

```
[stack@undercloud ~]$ sudo podman exec mysql bash -c "mysqldump -uroot -
p$PASSWORD --opt --all-databases" | sudo tee /root/undercloud-all-databases.sql
```

4. Source the undercloud credentials:

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

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

```
(undercloud) [stack@undercloud ~]$ cat <<'EOF' > ~/bar_rear_create_restore_images-
undercloud.yaml
# Playbook
# Using ReaR on the undercloud node.
- become: true
  hosts: undercloud
  name: Create the recovery images for the undercloud
  roles:
  - role: backup-and-restore
EOF
```

6. To create a backup of the undercloud node, enter the following **ansible-playbook** command:

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

CHAPTER 2. BACKING UP THE CONTROL PLANE NODES

To back up the control plane nodes, you configure the backup node, install the Relax-and-Recover tool on the control plane nodes, and create the backup image. You can create backups as a part of your regular environment maintenance.

You must back up the control plane nodes before performing updates or upgrades. You can use the backups to restore the control plane nodes to their previous state if an error occurs during an update or upgrade. You can also create backups as a part of your regular environment maintenance.

2.1. SUPPORTED BACKUP FORMATS AND PROTOCOLS

The undercloud and backup and restore process uses the open-source tool Relax-and-Recover (ReaR) to create and restore bootable backup images. ReaR is written in Bash and supports multiple image formats and multiple transport protocols.

The following list shows the backup formats and protocols that Red Hat OpenStack Platform supports.

Bootable media formats

- ISO

File transport protocols

- SFTP
- NFS

2.2. INSTALLING AND CONFIGURING AN NFS SERVER ON THE BACKUP NODE

You can install and configure a new NFS server to store the backup file. To install and configure an NFS server on the backup node, create an inventory file, set up an SSH key, and run the **openstack undercloud backup** command with the NFS server options.

IMPORTANT

- If you previously installed and configured an NFS or SFTP server, you do not need to complete this procedure. You enter the server information when you set up ReaR on the node that you want to back up.
- By default, the Relax 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, add the parameter **tripleo_backup_and_restore_server** to the setup ReaR command.

Procedure

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

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

2. On the undercloud node, create an inventory file for the backup node and replace the **<ip_address>** and **<user>** with the values that apply to your environment:

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

3. On the undercloud node, create the following Ansible playbook and replace **<backup_node>** with the host name of the backup node:

```
(undercloud) [stack@undercloud ~]$ cat <<'EOF' > ~/bar_nfs_setup.yaml
# Playbook
# Substitute <backup_node> with the host name of your backup node.
- become: true
  hosts: <backup_node>
  name: Setup NFS server for ReaR
  roles:
  - role: backup-and-restore
EOF
```

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 <backup_node>
```

Replace **<backup_node>** with the path and name of the backup node.

5. On the undercloud node, enter the following **ansible-playbook** commands to configure the backup node:

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

2.3. INSTALLING REAR ON THE CONTROL PLANE NODES

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

IMPORTANT

Due to a known issue, the ReaR backup of overcloud nodes continues even if a Controller node is down. Ensure that all your Controller nodes are running before you run the ReaR backup. A fix is planned for a later Red Hat OpenStack Platform (RHOSP) release. For more information, see BZ#2077335 - Back up of the overcloud ctlplane keeps going even if one controller is unreachable.

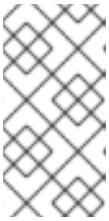
Prerequisites

- You have an NFS or SFTP server installed and configured on the backup node. For more information about creating a new NFS server, see Section 2.2, "Installing and configuring an NFS server on the backup node".

Procedure

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

```
(undercloud) [stack@undercloud ~]$ cat <<'EOF' > ~/bar_rear_setup-controller.yaml
# Playbook
# Install and configuring ReaR on the control plane nodes
- become: true
  hosts: Controller
  name: Install ReaR
  roles:
  - role: backup-and-restore
EOF
```



NOTE

If you deployed the control plane nodes with composable roles, replace the host type **Controller** with the types of nodes in your control plane. For example, if you deployed the database, messaging, and networking on separate nodes, enter **ControllerOpenstack,Database,Messaging,Networker**.

2. Choose one of the following options:

   a. If you use NFS and the IP address of the NFS server is the default value **192.168.24.1**, on the undercloud node, enter the following Ansible command to install ReaR on the control plane nodes:

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

   b. If you use SFTP and the IP address of the NFS server is not the default value **192.168.24.1**, enter the following Ansible command to install ReaR on the control plane nodes:

```
(undercloud) [stack@undercloud ~]$ ansible-playbook \
    -v -i ~/tripleo-inventory.yaml \
    --extra="ansible_ssh_common_args='-o StrictHostKeyChecking=no'" \
    --become \
    --become-user root \
    -e tripleo_backup_and_restore_server=<nfs_ip> \
    --tags bar_setup_rear \
    ~/bar_rear_setup-controller.yaml
```

   Replace **<nfs_ip>** with the IP address of your NFS server.

c. If you use SFTP, enter the following Ansible command do install ReaR on the control plane nodes:

```
(undercloud) [stack@undercloud ~]$ ansible-playbook \
  -v -i ~/tripleo-inventory.yaml \
  --extra="ansible_ssh_common_args='-o StrictHostKeyChecking=no'" \
  --become \
  --become-user root \
  -e tripleo_backup_and_restore_output_url=sftp://<user>:
<password>@<backup_node_ip>/ \
  -e tripleo_backup_and_restore_backup_url=iso:///backup/ \
  --tags bar_setup_rear \
  ~/bar_rear_setup-undercloud.yaml
```

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

a. Install the following tools:

```
$ 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**.

2.4. CONFIGURING OPEN VSWITCH (OVS) INTERFACES FOR BACKUP

If you use an Open vSwitch (OVS) bridge in your environment, you must manually configure the OVS interfaces before you create a backup of the undercloud or control plane nodes. The restoration process uses this information to restore the network interfaces.

Procedure

- In the **/etc/rear/local.conf** file, add the **NETWORKING_PREPARATION_COMMANDS** parameter in the following format:

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

Replace **<command_1>** and **<command_2>** with 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.

2.5. CREATING A BACKUP OF THE CONTROL PLANE NODES

To create a backup of the control plane nodes, use the **backup-and-restore** Ansible role. You can then use the backup to restore the control plane nodes to their previous state in case the nodes become corrupted or inaccessible. The backup of the control plane nodes includes the backup of the database that runs on the control plane nodes.

Prerequisites

- You have an NFS or SFTP server installed and configured on the backup node. For more information about creating a new NFS server, see Section 2.2, "Installing and configuring an NFS server on the backup node".
- You have installed ReaR on the control plane nodes. For more information, see Section 2.3, "Installing ReaR on the control plane nodes".
- If you use an OVS bridge for your network interfaces, you have configured the OVS interfaces. For more information, see Section 2.4, "Configuring Open vSwitch (OVS) interfaces for backup".

Procedure

1. On each control plane node, back up the **config-drive** partition of each node as the **root** user:

```
[root@controller-x ~]$ dd if=<config_drive_partition> of=/mnt/config-drive
```

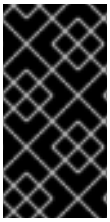
2. On the undercloud node, create the following Ansible playbook:

```
(undercloud) [stack@undercloud ~]$ cat <<'EOF' > ~/bar_rear_create_restore_images-controller.yaml
# Playbook
# Using ReaR on the control plane nodes.
- become: true
  hosts: ceph_mon
  name: Backup ceph authentication
  tasks:
    - name: Backup ceph authentication role
      include_role:
        name: backup-and-restore
        tasks_from: ceph_authentication
      tags:
      -  bar_create_recover_image
- become: true
  hosts: Controller
  name: Create the recovery images for the control plane
  roles:
  - role: backup-and-restore
EOF
```

3. On the undercloud node, enter the following **ansible-playbook** command to create a backup of the control plane nodes:

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

2.6. 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 an NFS or SFTP server installed and configured on the backup node. For more information about creating a new NFS server, see Section 1.2, “Installing and configuring an NFS server on the backup node”.
- You have installed ReaR on the undercloud and control plane nodes. For more information, see Section 2.3, “Installing ReaR on the control plane nodes” .
- 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:

```
[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}
source /home/stack/stackrc && /usr/bin/openstack overcloud backup 2>&1 >> ${FILE}
CURRENTTIME=`date`
echo "[$CURRENTTIME] rear end" >> ${FILE}
EOF
```

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

```
[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:

```
[stack@undercloud ~]# $ crontab -e
#adding the following line
0 0 * * * /home/stack/execute-rear-cron.sh
```

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:

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

The command output displays the scheduled cron jobs:

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

CHAPTER 3. BACKING UP CONTROL PLANE NODES THAT USE COMPOSABLE ROLES

If you deployed the control plane with composable roles, also known as custom roles, configure the backup process to capture each type of node based on your composable role configuration. To back up the control plane nodes, you configure the backup node, install the Relax-and-Recover tool on the control plane nodes, and create the backup image.

You must back up the control plane nodes before performing updates or upgrades. You can use the backups to restore the control plane nodes to their previous state if an error occurs during an update or upgrade. You can also create backups as a part of your regular environment maintenance.

3.1. SUPPORTED BACKUP FORMATS AND PROTOCOLS

The undercloud and backup and restore process uses the open-source tool Relax-and-Recover (ReaR) to create and restore bootable backup images. ReaR is written in Bash and supports multiple image formats and multiple transport protocols.

The following list shows the backup formats and protocols that Red Hat OpenStack Platform supports.

Bootable media formats

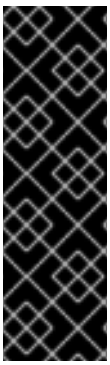
- ISO

File transport protocols

- SFTP
- NFS

3.2. INSTALLING AND CONFIGURING AN NFS SERVER ON THE BACKUP NODE

You can install and configure a new NFS server to store the backup file. To install and configure an NFS server on the backup node, create an inventory file, set up an SSH key, and run the **openstack undercloud backup** command with the NFS server options.



IMPORTANT

- If you previously installed and configured an NFS or SFTP server, you do not need to complete this procedure. You enter the server information when you set up ReaR on the node that you want to back up.
- By default, the Relax 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, add the parameter **tripleo_backup_and_restore_server** to the setup ReaR command.

Procedure

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

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

2. On the undercloud node, create an inventory file for the backup node and replace the **<ip_address>** and **<user>** with the values that apply to your environment:

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

3. On the undercloud node, create the following Ansible playbook and replace **<backup_node>** with the host name of the backup node:

```
(undercloud) [stack@undercloud ~]$ cat <<'EOF' > ~/bar_nfs_setup.yaml
# Playbook
# Substitute <backup_node> with the host name of your backup node.
- become: true
  hosts: <backup_node>
  name: Setup NFS server for ReaR
  roles:
  - role: backup-and-restore
EOF
```

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 <backup_node>
```

Replace **<backup_node>** with the path and name of the backup node.

5. On the undercloud node, enter the following **ansible-playbook** commands to configure the backup node:

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

3.3. INSTALLING REAR ON THE CONTROL PLANE NODES

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

IMPORTANT

Due to a known issue, the ReaR backup of overcloud nodes continues even if a Controller node is down. Ensure that all your Controller nodes are running before you run the ReaR backup. A fix is planned for a later Red Hat OpenStack Platform (RHOSP) release. For more information, see BZ#2077335 - Back up of the overcloud ctlplane keeps going even if one controller is unreachable.

Prerequisites

- You have an NFS or SFTP server installed and configured on the backup node. For more information about creating a new NFS server, see Section 3.2, "Installing and configuring an NFS server on the backup node".

Procedure

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

```
(undercloud) [stack@undercloud ~]$ cat <<'EOF' > ~/bar_rear_setup-controller.yaml
# Playbook
# Install and configuring ReaR on the control plane nodes
- become: true
  hosts: Controller
  name: Install ReaR
  roles:
  - role: backup-and-restore
EOF
```

NOTE

If you deployed the control plane nodes with composable roles, replace the host type **Controller** with the types of nodes in your control plane. For example, if you deployed the database, messaging, and networking on separate nodes, enter **ControllerOpenstack,Database,Messaging,Networker**.

2. Choose one of the following options:

   a. If you use NFS and the IP address of the NFS server is the default value **192.168.24.1**, on the undercloud node, enter the following Ansible command to install ReaR on the control plane nodes:

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

   b. If you use SFTP and the IP address of the NFS server is not the default value **192.168.24.1**, enter the following Ansible command to install ReaR on the control plane nodes:

```
(undercloud) [stack@undercloud ~]$ ansible-playbook \
    -v -i ~/tripleo-inventory.yaml \
    --extra="ansible_ssh_common_args='-o StrictHostKeyChecking=no'" \
    --become \
    --become-user root \
    -e tripleo_backup_and_restore_server=<nfs_ip> \
    --tags bar_setup_rear \
    ~/bar_rear_setup-controller.yaml
```

   Replace **<nfs_ip>** with the IP address of your NFS server.

c. If you use SFTP, enter the following Ansible command do install ReaR on the control plane nodes:

```
(undercloud) [stack@undercloud ~]$ ansible-playbook \
  -v -i ~/tripleo-inventory.yaml \
  --extra="ansible_ssh_common_args='-o StrictHostKeyChecking=no'" \
  --become \
  --become-user root \
  -e tripleo_backup_and_restore_output_url=sftp://<user>:
<password>@<backup_node_ip>/ \
  -e tripleo_backup_and_restore_backup_url=iso:///backup/ \
  --tags bar_setup_rear \
  ~/bar_rear_setup-undercloud.yaml
```

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

a. Install the following tools:

```
$ 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**.

3.4. CONFIGURING OPEN VSWITCH (OVS) INTERFACES FOR BACKUP

If you use an Open vSwitch (OVS) bridge in your environment, you must manually configure the OVS interfaces before you create a backup of the undercloud or control plane nodes. The restoration process uses this information to restore the network interfaces.

Procedure

- In the **/etc/rear/local.conf** file, add the **NETWORKING_PREPARATION_COMMANDS** parameter in the following format:

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

Replace **<command_1>** and **<command_2>** with 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.

3.5. CREATING A BACKUP OF CONTROL PLANE NODES THAT USE COMPOSABLE ROLES

To create a backup of control plane nodes that use composable roles, use the **backup-and-restore** Ansible role. You can then use the backup to restore the control plane nodes to their previous state in case the nodes become corrupted or inaccessible. The backup of the control plane nodes includes the backup of the database that runs on the control plane nodes.

Prerequisites

- You have an NFS or SFTP server installed and configured on the backup node. For more information about creating a new NFS server, see Section 3.2, "Installing and configuring an NFS server on the backup node".
- You have installed ReaR on the control plane nodes. For more information, see Section 3.3, "Installing ReaR on the control plane nodes".
- If you use an OVS bridge for your network interfaces, you have configured the OVS interfaces. For more information, see Section 3.4, "Configuring Open vSwitch (OVS) interfaces for backup".

Procedure

1. On each Controller node, back up the **config-drive** partition of each node:

```
[heat-admin@controller-x ~]$ mkdir /mnt/config-drive
[heat-admin@controller-x ~]$ dd if=<config_drive_partition> of=/mnt/config-drive
```

NOTE

You only need to perform this step on the Controller nodes.

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

```
(undercloud) [stack@undercloud ~]$ cat <<'EOF' > ~/bar_rear_create_restore_images-controller.yaml
# Playbook
# Using ReaR on the Contorl-Plane - Composable Roles

- become: true
  hosts: ControllerOpenstack,Database,Messaging,Networker
  name: Stop service management
  tasks:
    - include_role:
        name: backup-and-restore
        tasks_from: ../backup/tasks/service_manager_pause
      when:
       - tripleo_backup_and_restore_service_manager

- become: true
  hosts: Database
  name: Database Backup
  tasks:
    - include_role:
        name: backup-and-restore
        tasks_from: ../backup/tasks/db_backup

- become: true
  hosts: pacemaker
  name: Backup pacemaker configuration
  tasks:
    - include_role:
        name: backup-and-restore
        tasks_from: pacemaker_backup
```

```
- become: true
  hosts: ControllerOpenstack,Database,Messaging,Networker
  name: Create recovery images with ReaR
  tasks:
    - include_role:
        name: backup-and-restore
        tasks_from: ../backup/tasks/main

- become: true
  hosts: pacemaker
  name: Enabled pacemaker
  tasks:
    - name: Enable pacemaker
      command: pcs cluster start --all
      when: enabled_galera
      run_once: true
      tags:
       - bar_create_recover_image

- become: true
  hosts: Database
  name: Restart galera
  tasks:
    - name: unPause database container
      command: "{{ tripleo_container_cli }} unpause {{
tripleo_backup_and_restore_mysql_container }}"
      when:
       - tripleo_container_cli is defined
        - not enabled_galera
        - tripleo_backup_and_restore_mysql_container is defined
      tags:
       - bar_create_recover_image

- become: true
  hosts: ControllerOpenstack,Database,Messaging,Networker
  name: Unpause everything
  tasks:
    - name: Gather Container Service Name
      shell: |
       set -o pipefail
        /usr/bin/{{ tripleo_container_cli }} ps -a --filter='status=paused' --format '{{ '{{' }}.Names {{
'}}' }} '
      register: container_services
      changed_when: container_services.stdout is defined
      tags:
       - bar_create_recover_image

    - name: Unpause containers for database backup.
      command: "{{ tripleo_container_cli }} unpause {{ item }}"
      with_items: "{{ container_services.stdout_lines }}"
      when: tripleo_container_cli is defined
      tags:
       - bar_create_recover_image
```

3. On the undercloud node, enter the following **ansible-playbook** command to 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 or SFTP 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 ~]$ ansible-playbook \
  -v -i ~/tripleo-inventory.yaml \
  --extra="ansible_ssh_common_args='-o StrictHostKeyChecking=no'" \
  --become \
  --become-user root \
  --tags bar_create_recover_image \
  ~/bar_rear_create_restore_images-controller.yaml
```

3.6. 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 an NFS or SFTP server installed and configured on the backup node. For more information about creating a new NFS server, see Section 1.2, “Installing and configuring an NFS server on the backup node”.
- You have installed ReaR on the undercloud and control plane nodes. For more information, see Section 2.3, “Installing ReaR on the control plane nodes” .
- 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:

```
[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}
source /home/stack/stackrc && /usr/bin/openstack overcloud backup 2>&1 >> ${FILE}
CURRENTTIME=`date`
echo "[$CURRENTTIME] rear end" >> ${FILE}
EOF
```

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

```
[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:

```
[stack@undercloud ~]# $ crontab -e
#adding the following line
0 0 * * * /home/stack/execute-rear-cron.sh
```

   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:

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

   The command output displays the scheduled cron jobs:

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

3.7. ADDITIONAL RESOURCES

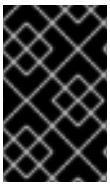
- Composable services and custom roles

CHAPTER 4. RESTORING THE UNDERCLOUD AND CONTROL PLANE NODES

If your undercloud or control plane nodes become corrupted or if an error occurs during an update or upgrade, you can restore the undercloud or overcloud control plane nodes from a backup to their previous state. If the restore process fails to automatically restore the Galera cluster or nodes with colocated Ceph monitors, you can restore these components manually.

4.1. PREPARING A CONTROL PLANE WITH COLOCATED CEPH MONITORS FOR THE RESTORE PROCESS

Before you restore a control plane with colocated Ceph monitors, prepare your environment by creating a script that mounts the Ceph monitor backup file to the node file system and another script that ReaR uses to locate the backup file.



IMPORTANT

If you cannot back up the **/var/lib/ceph** directory, you must contact the Red Hat Technical Support team to rebuild the **ceph-mon** index. For more information, see Red Hat Technical Support Team.

Prerequisites

- You have created a backup of the undercloud node. For more information, see Section 1.5, "Creating a backup of the undercloud node".
- You have created a backup of the control plane nodes. For more information, see Section 2.5, "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, you have access to the network configuration information that you set in the **NETWORKING_PREPARATION_COMMANDS** parameter. For more information, see see Section 1.4, "Configuring Open vSwitch (OVS) interfaces for backup".

Procedure

1. On each node that you want to restore, create the script **/usr/share/rear/setup/default/011_backup_ceph.sh** and add the following content:

```
mount -t <file_type> <device_disk> /mnt/local
cd /mnt/local
[ -d "var/lib/ceph" ] && tar cvfz /tmp/ceph.tar.gz var/lib/ceph --xattrs --xattrs-include='.' --acls
cd /
umount <device_disk>
```

   Replace **<file_type>** and **<device_disk>** with the type and location of the backup file. Normally, the file type is **xfs** and the location is **/dev/vda2**.

2. On the same node, create the script **/usr/share/rear/wrapup/default/501_restore_ceph.sh** and add the following content:

```
if [ -f "/tmp/ceph.tar.gz" ]; then
  rm -rf /mnt/local/var/lib/ceph/*
  tar xvC /mnt/local -f /tmp/ceph.tar.gz var/lib/ceph --xattrs --xattrs-include='.'
fi
```

Additional resources

- Section 4.2, “Restoring the undercloud node”
- Section 4.3, “Restoring the control plane nodes”

4.2. RESTORING THE UNDERCLOUD NODE

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 Section 2.5, “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, you have access to the network configuration information that you set in the **NETWORKING_PREPARATION_COMMANDS** parameter. For more information, see see Section 1.4, “Configuring Open vSwitch (OVS) interfaces for backup”.

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>**. Replace **<undercloud_node>** with 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. Power off the node:

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

On boot up, the node resumes its previous state.

4.3. 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

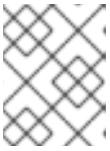
Red Hat supports backups of Red Hat OpenStack Platform with native SDNs, such as Open vSwitch (OVS) and the default Open Virtual Network (OVN). For information about third-party SDNs, refer to the third-party SDN documentation.

Prerequisites

- Choose one of the following options:
  - You have created a backup of the control plane nodes without composable roles. For more information, see Section 2.5, “Creating a backup of the control plane nodes” .
  - You have created a backup of control plane nodes that use composable roles. For more information, see Section 3.5, “Creating a backup of control plane nodes that use composable roles”.
- You have access to the backup node.
- If you use an OVS bridge for your network interfaces, you have access to the network configuration information that you set in the **NETWORKING_PREPARATION_COMMANDS** parameter. For more information, see see Section 2.4, “Configuring Open vSwitch (OVS) interfaces for backup”.

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:

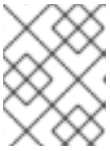
```
Welcome to Relax-and-Recover. Run "rear recover" to restore your system!
RESCUE <control_plane_node>:~ # rear recover
```

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. When the command line console is available, restore the **config-drive** partition of each control plane node:

```
# once completed, restore the config-drive partition (which is ISO9660)
RESCUE <control_plane_node>:~ $ dd if=/mnt/local/mnt/config-drive of=
<config_drive_partition>
```



NOTE

If you deployed a control plane with composable roles, perform this step only on the Controller nodes.

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
```

9. To view the status of the overcloud, use the OpenStack Integration Test Suite (tempest). For more information, see Validating your OpenStack cloud with the Integration Test Suite (tempest).

Troubleshooting

- Clear resource alarms that are displayed by **pcs status** by running the following command:

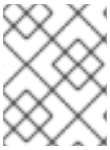
```
# pcs resource clean
```

- Clear STONITH fencing action errors that are displayed by **pcs status** by running the following commands:

```
# pcs resource clean
# pcs stonith history cleanup
```

4.4. 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/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 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 "mysql_install_db --datadir=/var/lib/mysql --
user=mysql --log_error=/var/log/mysql/mysql_init.log"
```

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:

```
$ 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:

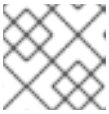
```
$ sudo podman exec $(sudo podman container ls --all --format "{{ .Names }}"  \
    --filter=name=galera-bundle) bash -c "mysql -uroot -e'use mysql;update user set
password=PASSWORD(\"$ROOTPASSWORD\")where User=\"root\";flush privileges;'"
```

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 "mysqladmin 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 exec $(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 the OpenStack Integration Test Suite (tempest). For more information, see Validating your OpenStack cloud with the Integration Test Suite (tempest).

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
```

4.5. RESTORING THE UNDERCLOUD NODE DATABASE MANUALLY

If the undercloud database does not restore as part of the undercloud restore process, you can restore the database manually. You can only restore the database if you previously created a standalone database backup.

Prerequisites

- You have created a backup of the undercloud database. For more information about backing up the undercloud database, see Section 1.5, “Creating a backup of the undercloud node” .

Procedure

1. Log in to the director undercloud node as the **root** user.

2. Stop all tripleo services:

```
[root@director ~]# systemctl  stop  tripleo_*
```

3. Ensure that no containers are running on the server by entering the following command:

```
[root@director ~]# podman ps
```

If any containers are running, enter the following command to stop the containers:

```
[root@director ~]# podman stop <container_name>
```

4. Create a backup of the current **/var/lib/mysql** directory and then delete the directory:

```
[root@director ~]# cp -a /var/lib/mysql /var/lib/mysql_bck
[root@director ~]# rm -rf /var/lib/mysql
```

5. Recreate the database directory and set the SELinux attributes for the new directory:

```
[root@director ~]# mkdir /var/lib/mysql
[root@director ~]# chown 42434:42434 /var/lib/mysql
[root@director ~]# chmod 0755 /var/lib/mysql
[root@director ~]# chcon -t container_file_t /var/lib/mysql
[root@director ~]# chcon -r object_r /var/lib/mysql
[root@director ~]# chcon -u system_u /var/lib/mysql
```

6. Create a local tag for the **mariadb** image. Replace **<image_id>** and **<undercloud.ctlplane.example.com>** with the values applicable in your environment:

```
[root@director ~]# podman images | grep mariadb
<undercloud.ctlplane.example.com>:8787/rh-osbs/rhosp16-openstack-mariadb
16.2_20210322.1   <image_id>   3 weeks ago   718 MB
```

```
[root@director ~]# podman tag <image_id> mariadb
```

```
[root@director ~]# podman images | grep maria
localhost/mariadb                                                    latest      <image_id>   3
weeks ago   718 MB
<undercloud.ctlplane.example.com>:8787/rh-osbs/rhosp16-openstack-mariadb
16.2_20210322.1   <image_id>   3 weeks ago   718 MB
```

7. Initialize the **/var/lib/mysql** directory with the container:

```
[root@director ~]# podman run --net=host -v /var/lib/mysql:/var/lib/mysql localhost/mariadb
mysql_install_db --datadir=/var/lib/mysql --user=mysql
```

8. Copy the database backup file that you want to import to the database:

```
[root@director ~]# cp /root/undercloud-all-databases.sql /var/lib/mysql
```

9. Start the database service to import the data:

```
[root@director ~]# podman run --net=host -dt -v /var/lib/mysql:/var/lib/mysql
localhost/mariadb  /usr/libexec/mysqld
```

10. Import the data and configure the **max_allowed_packet** parameter:

a. Log in to the container and configure it:

```
[root@director ~]# podman exec -it <container_id> /bin/bash
    ()[mysql@5a4e429c6f40 /]$ mysql -u root -e "set global max_allowed_packet =
1073741824;"
    ()[mysql@5a4e429c6f40 /]$ mysql -u root < /var/lib/mysql/undercloud-all-
databases.sql
    ()[mysql@5a4e429c6f40 /]$ mysql -u root -e 'flush privileges'
    ()[mysql@5a4e429c6f40 /]$ exit
    exit
```

b. Stop the container:

```
[root@director ~]# podman stop <container_id>
```

c. Check that no containers are running:

```
[root@director ~]# podman ps
CONTAINER ID  IMAGE  COMMAND  CREATED  STATUS  PORTS  NAMES
[root@director ~]#
```

11. Restart all tripleo services:

```
[root@director ~]# systemctl start multi-user.target
```