Red Hat OpenStack Platform 17.0

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: 2024-03-21

Red Hat OpenStack Platform 17.0 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

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.

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 when you use ReaR to back up and restore the undercloud and control plane.

Bootable media formats

- ISO

File transport protocols

- SFTP
- NFS

1.2. CONFIGURING THE BACKUP STORAGE LOCATION

Before you create a backup of the control plane nodes, configure the backup storage location in the **bar-vars.yaml** environment file. This file stores the key-value parameters that you want to pass to the backup execution.

Procedure

1. Log in to the undercloud as the **stack** user.
2. Source the **stackrc** file:

```
$ source ~/stackrc
```

3. Create the **bar-vars.yaml** file:

```
touch /home/stack/bar-vars.yaml
```

4. In the **bar-vars.yaml** file, configure the backup storage location:
   a. If you use an NFS server, add the following parameters and set the values of the IP address of your NFS server and backup storage folder:

```
tripleo_backup_and_restore_server: <ip_address>
tripleo_backup_and_restore_shared_storage_folder: <backup_dir>
```

By default, the **tripleo_backup_and_restore_server** parameter value is **192.168.24.1**.

b. If you use an SFTP server, add the **tripleo_backup_and_restore_output_url** parameter and set the values of the URL and credentials of the SFTP server:

```
tripleo_backup_and_restore_output_url: sftp://<user>:<password>@<backup_node>/
tripleo_backup_and_restore_backup_url: iso:///backup/
```

Replace **<user>**, **<password>**, and **<backup_node>** with the backup node URL and credentials.

1.3. OPTIONAL: CONFIGURING BACKUP ENCRYPTION

You can encrypt backups as an additional security measure to protect sensitive data.

Procedure

- In the **bar-vars.yaml** file, add the following parameters:

```
tripleo_backup_and_restore_crypt_backup_enabled: true
tripleo_backup_and_restore_crypt_backup_password: <password>
```

Replace **<password>** with the password you want to use to encrypt the backup.

1.4. 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, create 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) IP address parameter for the NFS server is **192.168.24.1**. You must add the parameter **tripleo_backup_and_restore_server** to set the IP address value that matches your environment.

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]
<backup_node> 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 <backup_node>
```

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

4. Configure the NFS server on the backup node:

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

1.5. 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 1.4, "Installing and configuring an NFS server on the backup node".

Procedure

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

```
[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, extract the static ansible inventory file from the location in which it was saved during installation:

```
(undercloud) [stack@undercloud ~]$ cp ~/overcloud-deploy/<stack>/tripleo-ansible-
inventory.yaml ~/tripleo-inventory.yaml
```

- Replace **<stack>** with the name of your stack. By default, the name of the stack is **overcloud**.

3. Install ReaR on the undercloud node:

```
(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:

```
$ 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**.

1.6. CREATING A STANDALONE DATABASE BACKUP OF THE UNDERCLOUD NODES

You can include standalone undercloud database backups in your routine backup schedule to provide additional data security. A full backup of an undercloud node includes a database backup of the undercloud node. But if a full undercloud restoration fails, you might lose access to the database portion of the full undercloud backup. In this case, you can recover the database from a standalone undercloud database backup.

Procedure

- Create a database backup of the undercloud nodes:

```
openstack undercloud backup --db-only
```

The db backup file is stored in **/home/stack with the name openstack-backup-mysql-<timestamp>.sql**.

Additional resources

- Section 1.8, “Creating a backup of the undercloud node”
- Section 3.5, “Restoring the undercloud node database manually”

1.7. 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.

1.8. CREATING A BACKUP OF THE UNDERCLOUD NODE

To create a backup of the undercloud node, use the **openstack undercloud backup** command. 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 1.4, "Installing and configuring an NFS server on the backup node".
- You have installed ReaR on the undercloud node. For more information, see Section 1.5, "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 1.7, "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. On the undercloud node, source the undercloud credentials:

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

5. 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 tripleo-admin \
--static-yaml-inventory /home/stack/tripleo-inventory.yaml
```

6. Create a backup of the undercloud node:

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

1.9. SCHEDULING UNDERCLOUD NODE BACKUPS WITH CRON

You can schedule backups of the undercloud nodes with ReaR by using the Ansible **backup-and-restore** role. You can view the logs in the **/var/log/rear-cron** directory.

Prerequisites

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 1.4, "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 2.4, "Installing ReaR on the control plane nodes" .
- You have sufficient available disk space at your backup location to store the backup.

Procedure

1. To schedule a backup of your control plane nodes, run the following command. The default schedule is Sundays at midnight:

```
openstack undercloud backup --cron
```

2. Optional: Customize the scheduled backup according to your deployment:
   - To change the default backup schedule, pass a different cron schedule on the **tripleo_backup_and_restore_cron** parameter:

```
openstack undercloud backup --cron --extra-vars
'{"tripleo_backup_and_restore_cron": "0 0 * * 0"}'
```

   - To define additional parameters that are added to the backup command when cron runs the scheduled backup, pass the **tripleo_backup_and_restore_cron_extra** parameter to the backup command, as shown in the following example:

```
openstack undercloud backup --cron --extra-vars
'{"tripleo_backup_and_restore_cron_extra":"--extra-vars bar-vars.yaml --inventory
/home/stack/tripleo-inventory.yaml"}'
```

   - To change the default user that executes the backup, pass the **tripleo_backup_and_restore_cron_user** parameter to the backup command, as shown in the following example:

```
openstack undercloud backup --cron --extra-vars
'{"tripleo_backup_and_restore_cron_user": "root"}
```

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.

In addition, 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.

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 when you use ReaR to back up and restore the undercloud and control plane.

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, create 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) IP address parameter for the NFS server is **192.168.24.1**. You must add the parameter **tripleo_backup_and_restore_server** to set the IP address value that matches your environment.

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]
<backup_node> 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 <backup_node>
```

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

4. Configure the NFS server on the backup node:

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

2.3. CONFIGURING A BACKUP WITH COMPOSABLE ROLES

If your deployment uses composable roles, you can select the groups of nodes that you want to include in the backup. The names of the groups of nodes are found in the inventory file. You can back up a single group of nodes or multiple groups.

Procedure

1. In the **bar-vars.yaml** environment file, add the **tripleo_controller_group_name** parameter:

```
tripleo_controller_group_name: <group_names_list>
```

2. Set the **<group_names_list>** value to the list of names of the groups of nodes that you want to back up.

2.4. 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, source the undercloud credentials:

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

2. If you have not done so before, extract the static ansible inventory file from the location in which it was saved during installation:

```
(undercloud) [stack@undercloud ~]$ cp ~/overcloud-deploy/<stack>/tripleo-ansible-inventory.yaml ~/tripleo-inventory.yaml
```

   - Replace **<stack>** with the name of your stack. By default, the name of the stack is **overcloud**.

3. In the **bar-vars.yaml** file, configure the backup storage location:

   a. If you installed and configured your own NFS server, add the **tripleo_backup_and_restore_server** parameter and set the value to the IP address of your NFS server:

```
tripleo_backup_and_restore_server: <ip_address>
```

   By default, the **tripleo_backup_and_restore_server** parameter value is **192.168.24.1**.

   b. If you use an SFTP server, add the **tripleo_backup_and_restore_output_url** parameter and set the values of the URL and credentials of the SFTP server:

```
tripleo_backup_and_restore_output_url: sftp://<user>:<password>@<backup_node>/
tripleo_backup_and_restore_backup_url: iso:///backup/
```

   Replace **<user>**, **<password>**, and **<backup_node>** with the backup node URL and credentials.

4. Install ReaR on the control plane nodes:

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

5. 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**.

2.5. 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.

2.6. CREATING A BACKUP OF THE CONTROL PLANE NODES

To create a backup of the control plane nodes, use the **openstack overcloud backup** command. 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 2.4, “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 2.5, “Configuring Open vSwitch (OVS) interfaces for backup”.

Procedure

1. Locate the **config-drive** partition on each control plane node:

```
[stack@undercloud-0 ~]$ lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
vda    253:0    0   55G  0 disk
├─vda1 253:1    0    1M  0 part          1
├─vda2 253:2    0  100M  0 part /boot/efi
└─vda3 253:3    0 54.9G  0 part /
```

1 The **config-drive** partition is the 1M partition that is not mounted.

2. 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
```

Replace **<config_drive_partition>** with the name of the **config-drive** partition that you located in step 1.

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

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

4. 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 tripleo-admin \
--static-yaml-inventory /home/stack/tripleo-inventory.yaml
```

5. Create a backup of the control plane nodes:

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

The backup process runs sequentially on each control plane node without disrupting the service to your environment.

2.7. SCHEDULING CONTROL PLANE NODE BACKUPS WITH CRON

You can schedule backups of the control plane nodes with ReaR by 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 1.4, “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 2.4, “Installing ReaR on the control plane nodes” .
- You have sufficient available disk space at your backup location to store the backup.

Procedure

1. To schedule a backup of your control plane nodes, run the following command. The default schedule is Sundays at midnight:

```
openstack overcloud backup --cron
```

2. Optional: Customize the scheduled backup according to your deployment:
   - To change the default backup schedule, pass a different cron schedule on the **tripleo_backup_and_restore_cron** parameter:

```
openstack overcloud backup --cron --extra-vars
'{"tripleo_backup_and_restore_cron": "0 0 * * 0"}'
```

- To define additional parameters that are added to the backup command when cron runs the scheduled backup, pass the **tripleo_backup_and_restore_cron_extra** parameter to the backup command, as shown in the following example:

```
openstack overcloud backup --cron --extra-vars
'{"tripleo_backup_and_restore_cron_extra":"--extra-vars bar-vars.yaml --inventory
/home/stack/tripleo-inventory.yaml"}'
```

- To change the default user that executes the backup, pass the **tripleo_backup_and_restore_cron_user** parameter to the backup command, as shown in the following example:

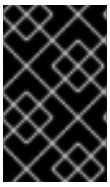
```
openstack overcloud backup --cron --extra-vars
'{"tripleo_backup_and_restore_cron_user": "root"}
```

CHAPTER 3. 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.

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

Before you restore a control plane nodes 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 1.8, "Creating a backup of the undercloud node".
- You have created a backup of the control plane nodes. For more information, see Section 2.6, "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 1.7, "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 3.2, “Restoring the undercloud node”
- Section 3.3, “Restoring the control plane nodes”

3.2. 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 1.8, “Creating a backup of the undercloud node”.
- 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 1.7, “Configuring Open vSwitch (OVS) interfaces for backup”.
- If you configured backup encryption, you must decrypt the backup before you begin the restoration process. Run the following decrypt step in the system where the backup file is located:

```
$ dd if=backup.tar.gz | /usr/bin/openssl des3 -d -k "<encryption key>" | tar -C
<backup_location> -xzvf - '*.conf'
```

  - Replace **<encryption key>** with your encryption key.
  - Replace **<backup_location>** with the folder in which you want to save the **backup.tar.gz** file, for example, **/ctl_plane_backups/undercloud-0/**.

Procedure

1. Log in to the hypervisor:

```
$ ssh root@<hypervisor>
```

2. Power off the undercloud node. Ensure that the undercloud node is powered off completely before you proceed:

```
$ virsh destroy undercloud-0
```

3. Boot the undercloud node with the backup ISO image.

4. 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.

5. 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
```

6. Power off the node:

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

On boot up, the node resumes its previous state.

3.3. 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

- You have created a backup of the control plane nodes. For more information, see Section 2.6, “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 2.5, “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>
```

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:

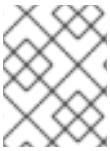
```
# pcs resource clean
```

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

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

3.4. 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/tripleo-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
```

3.5. 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 standalone backup of the undercloud database. For more information, see Section 1.6, “Creating a standalone database backup of the undercloud nodes” .

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
```