



Red Hat OpenStack Platform 16.2

Backing up and restoring the undercloud and control plane nodes

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

Red Hat OpenStack Platform 16.2 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

Copyright © 2021 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

XFS[®] is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

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. CREATING A STANDALONE DATABASE BACKUP OF THE UNDERCLOUD NODE	7
1.5. CONFIGURING OPEN VSWITCH (OVS) INTERFACES FOR BACKUP	9
1.6. CREATING A BACKUP OF THE UNDERCLOUD NODE	9
CHAPTER 2. BACKING UP THE CONTROL PLANE NODES	11
2.1. SUPPORTED BACKUP FORMATS AND PROTOCOLS	11
2.2. INSTALLING AND CONFIGURING AN NFS SERVER ON THE BACKUP NODE	11
2.3. INSTALLING REAR ON THE CONTROL PLANE NODES	12
2.4. CONFIGURING OPEN VSWITCH (OVS) INTERFACES FOR BACKUP	13
2.5. CREATING A BACKUP OF THE CONTROL PLANE NODES	14
2.6. SCHEDULING CONTROL PLANE NODE BACKUPS WITH CRON	15
CHAPTER 3. RESTORING THE UNDERCLOUD AND CONTROL PLANE NODES	17
3.1. PREPARING A CONTROL PLANE WITH COLOCATED CEPH MONITORS FOR THE RESTORE PROCESS	17
3.2. RESTORING THE UNDERCLOUD NODE	18
3.3. RESTORING THE CONTROL PLANE NODES	19
3.4. RESTORING THE GALERA CLUSTER MANUALLY	20
3.5. RESTORING THE UNDERCLOUD NODE DATABASE MANUALLY	24

MAKING OPEN SOURCE MORE INCLUSIVE

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

PROVIDING FEEDBACK ON RED HAT DOCUMENTATION

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

Using the Direct Documentation Feedback (DDF) function

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

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

CHAPTER 1. 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

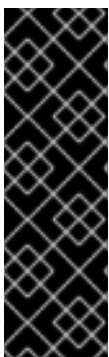
- ISO

File transport protocols

- SFTP
- NFS

1.2. 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, you must add the parameter **tripleo_backup_and_restore_nfs_server** to the setup ReaR command.

Procedure

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

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

- 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.

- 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.

- 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
```



NOTE

- By default, the **tripleo_backup_and_restore_nfs_server** parameter is set to **192.168.24.1** in the **bar-vars.yaml** environment file. If you installed and configured your own NFS server, add the **tripleo_backup_and_restore_nfs_server** parameter to **bar-vars.yaml** and set the value to the IP address of your NFS server.
- By default, the **tripleo_backup_and_restore_nfs_clients_nets** parameter is set to **['192.168.24.0/24', '10.0.0.0/24', '172.16.0.0/24']** in the **bar-vars.yaml** environment file. Add the network ranges where your undercloud and overcloud are deployed to ensure that the NFS backup directory is exported to the correct location.

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

- 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, create an inventory file and use the `tripleo-ansible-inventory` command to generate a static inventory file that contains hosts and variables for all the overcloud nodes:

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

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. In the `bar-vars.yaml` file, configure the backup storage location:
 - a. If you use an NFS server and you installed and configured your own NFS server, add the `tripleo_backup_and_restore_nfs_server` parameter and set the value to the IP address of your NFS server:

```
tripleo_backup_and_restore_nfs_server: <ip_address>
```

By default, the `tripleo_backup_and_restore_nfs_server` parameter value is **192.168.24.1**.

- b. If you use an SFTP server to store the backup, 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.

5. 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. CREATING A STANDALONE DATABASE BACKUP OF THE UNDERCLOUD NODE

If you are upgrading your Red Hat OpenStack Platform environment from 13 to 16.2, you must create a standalone database backup after you perform the undercloud upgrade and before you perform the Leapp upgrade process on the overcloud nodes. In other scenarios, you can optionally create the standalone database backup in addition to a full backup of the undercloud node.



NOTE

When you create a full backup of the undercloud node, it includes a database backup of the undercloud node. Therefore, you do not need to create a standalone database backup if you plan to restore the undercloud from the full backup. However, if the full undercloud restore process fails and you want to restore the undercloud database manually, you must have access to the standalone database backup that you create with this procedure.

Procedure

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

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

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

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

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

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

5. Create a database backup of the undercloud node:

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

Additional resources

- [Framework for Upgrades \(13 to 16.2\)](#)

- [Section 1.6, “Creating a backup of the undercloud node”](#)
- [Section 3.5, “Restoring the undercloud node database manually”](#)

1.5. 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.

1.6. 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.

If you are upgrading your Red Hat OpenStack Platform environment from 13 to 16.2, you must create a separate database backup after you perform the undercloud upgrade and before you perform the Leapp upgrade process on the overcloud nodes. For more information, see [Section 1.4, “Creating a standalone database backup of the undercloud node”](#).



NOTE

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

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 control plane nodes. 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 1.5, “Configuring Open vSwitch \(OVS\) interfaces for backup”](#).

Procedure

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

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

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

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

3. Create a backup of the undercloud node:

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

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

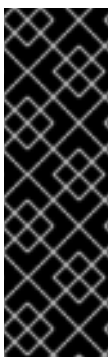
- ISO

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, 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, you must add the parameter **tripleo_backup_and_restore_nfs_server** to the setup ReaR command.

Procedure

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

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

- 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.

- 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.

- 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
```



NOTE

- By default, the **tripleo_backup_and_restore_nfs_server** parameter is set to **192.168.24.1** in the **bar-vars.yaml** environment file. If you installed and configured your own NFS server, add the **tripleo_backup_and_restore_nfs_server** parameter to **bar-vars.yaml** and set the value to the IP address of your NFS server.
- By default, the **tripleo_backup_and_restore_nfs_clients_nets** parameter is set to **['192.168.24.0/24', '10.0.0.0/24', '172.16.0.0/24']** in the **bar-vars.yaml** environment file. Add the network ranges where your undercloud and overcloud are deployed to ensure that the NFS backup directory is exported to the correct location.

2.3. INSTALLING REAR ON THE CONTROL PLANE NODES

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

Procedure

- On the undercloud node, source the undercloud credentials:

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


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

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

3. Install ReaR on the control plane nodes:

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

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

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

```
tripleo_backup_and_restore_nfs_server: <ip_address>
```

By default, the **tripleo_backup_and_restore_nfs_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.

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**.

2.4. 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 **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 2.3, “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, source the undercloud credentials:

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

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

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

4. Create a backup of the control plane nodes:



IMPORTANT

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

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

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}
/usr/bin/ansible-playbook -v -i /home/stack/tripleo-inventory.yaml --
extra="ansible_ssh_common_args='-o StrictHostKeyChecking=no'" --become --become-user
root --tags bar_create_recover_image --
extra="tripleo_backup_and_restore_service_manager=false"
/home/stack/bar_rear_create_restore_images.yaml 2>&1 >> ${FILE}
CURRENTTIME=`date`
echo "[$CURRENTTIME] rear end" >> ${FILE}
EOF
```

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. 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 1.6, “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 [Section 1.5, “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 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 [Section 1.5, “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.

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 2.5, “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 [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.

- 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
```

- 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>
```

- Power off the node:

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

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

- 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 \(tempest\)](#).

3.4. 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

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

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
```

- 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 \(tempest\)](#).
- 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 1.4, "Creating a standalone database backup of the undercloud node"](#).

Procedure

- Log in to the director undercloud node as the **root** user.
- Create a backup of the current **/var/lib/mysql** directory and then delete the directory:

```
[root@director ~]# systemctl stop tripleo_mysql_healthcheck.timer
[root@director ~]# systemctl stop tripleo_mysql.service
[root@director ~]# cp -a /var/lib/mysql /var/lib/mysql_bck
[root@director ~]# rm -rf /var/lib/mysql
```

- 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
```

- 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

5. Initialize the `/var/lib/mysql` directory with the container:

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

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

```
[root@director ~]# cp /home/stack/openstack-backup-mysql.sql /var/lib/mysql
```

7. Start the database service to import the data and replace `<container_id>` with the container ID value applicable in your environment:

```
[root@director ~]# podman run -dt -v /var/lib/mysql:/var/lib/mysql localhost/mariadb
/usr/libexec/mysqld
<container_id>
```

8. 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/openstack-backup-mysql.sql
- ()[mysql@5a4e429c6f40 /]$ mysql -u root < /var/lib/mysql/openstack-backup-mysql-
grants.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 ~]#
```

9. Restart the MySQL services:

```
[root@director ~]# systemctl start tripleo_mysql.service
[root@director ~]# systemctl start tripleo_mysql_healthcheck.timer
```