



Red Hat OpenStack Platform 17.1

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 17.1 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

Copyright © 2023 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 back up and restore your Red Hat Openstack Platform (RHOSP). This guide also explains how to troubleshoot backup and restore problems. You must create backups when you update or upgrade your RHOSP. To back up your RHOSP, you can use either of the following tools: The Snapshot and Revert tool. When you create a snapshot, you preserve the original disk state of your RHOSP cluster. Depending on the result of your update or upgrade, you can remove or revert the snapshots. The Relax-and-Recover (ReaR) tool. When you back up your environment with the ReaR tool, you create the backup images of the undercloud node and the control plane nodes. You can use these backups to restore the undercloud node and the control

plane nodes to their previous states if an error occurs during an upgrade or update. Also, you can regularly create backups of your environment with the ReaR tool to minimize downtime if there are issues.

Table of Contents

MAKING OPEN SOURCE MORE INCLUSIVE	3
PROVIDING FEEDBACK ON RED HAT DOCUMENTATION	4
CHAPTER 1. BACKING UP YOUR RED HAT OPENSTACK PLATFORM CLUSTER BY USING THE SNAPSHOT AND REVERT TOOL	5
1.1. CREATING A SNAPSHOT OF CONTROLLER AND COMPUTE NODES	5
CHAPTER 2. BACKING UP THE UNDERCLOUD AND THE CONTROL PLANE NODES BY USING THE RELAX-AND-RECOVER TOOL	7
2.1. BACKING UP THE UNDERCLOUD NODE BY USING THE RELAX-AND-RECOVER TOOL	7
2.1.1. Supported backup formats and protocols	7
2.1.2. Configuring the backup storage location	7
2.1.3. Optional: Configuring backup encryption	8
2.1.4. Installing and configuring an NFS server on the backup node	8
2.1.5. Installing ReaR on the undercloud node	9
2.1.6. Optional: Creating a standalone database backup of the undercloud nodes	10
2.1.7. Configuring Open vSwitch (OVS) interfaces for backup	11
2.1.8. Creating a backup of the undercloud node	11
2.1.9. Scheduling undercloud node backups with cron	12
2.2. BACKING UP THE CONTROL PLANE NODES BY USING THE RELAX-AND-RECOVER TOOL	13
2.2.1. Supported backup formats and protocols	13
2.2.2. Installing and configuring an NFS server on the backup node	13
2.2.3. Installing ReaR on the control plane nodes	14
2.2.4. Configuring Open vSwitch (OVS) interfaces for backup	15
2.2.5. Creating a backup of the control plane nodes	16
2.2.6. Scheduling control plane node backups with cron	17
2.3. RESTORING THE UNDERCLOUD NODE AND CONTROL PLANE NODES BY USING THE RELAX-AND-RECOVER TOOL	18
2.3.1. Restoring the undercloud node	18
2.3.2. Restoring the control plane nodes	19
2.3.3. Restoring the Galera cluster manually	21
2.3.4. 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 YOUR RED HAT OPENSTACK PLATFORM CLUSTER BY USING THE SNAPSHOT AND REVERT TOOL

Snapshots preserve the original disk state of your Red Hat OpenStack Platform (RHOSP) cluster before you perform an upgrade or an update from RHOSP 17.1 or later. You can then remove or revert the snapshots depending on the results. For example, if an upgrade completed successfully and you do not need the snapshots anymore, remove them from your nodes. If an upgrade fails, you can revert the snapshots, assess any errors, and start the upgrade procedure again. A revert leaves the disks of all the nodes exactly as they were when the snapshot was taken.

The RHOSP Snapshot and Revert tool is based on the Logical Volume Manager (LVM) snapshot functionality and is only intended to revert an unsuccessful upgrade or update.



IMPORTANT

The snapshots are stored on the same hard drives as the data you have stored on your disks. As a result, the Snapshot and Revert tool does not prevent data loss in cases of hardware failure, data center failure, or inaccessible nodes.

You can take snapshots of Controller nodes and Compute nodes. Taking snapshots of the undercloud is not supported.

1.1. CREATING A SNAPSHOT OF CONTROLLER AND COMPUTE NODES

Create a snapshot of your Controller and Compute nodes before performing an upgrade or update. You can then remove or revert the snapshots depending on the results of those actions.



NOTE

You can create only one snapshot of your Controller and Compute nodes. To create another snapshot, you must remove or revert your previous snapshot.

Prerequisites

- You have LVM enabled on the node.
- The following default set of LVM logical volumes defined by a RHOSP installation are present:
 - `/dev/vg/lv_audit`
 - `/dev/vg/lv_home`
 - `/dev/vg/lv_log`
 - `/dev/vg/lv_root`
 - `/dev/vg/lv_srv`
 - `/dev/vg/lv_var`

You can run the **lvs**, **lvscan**, or **lvdisplay** commands to confirm whether your environment includes these prerequisites before you make changes to the node disks.



NOTE

These prerequisites are included with the default installation of a 17.1 cluster. However, if you upgraded to RHOSP 17.1 from an earlier RHOSP version, your control plane does not include these prerequisites because they require reformatting of the disk.

Procedure

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

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

3. 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**.

4. Take the snapshots:

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

5. If your upgrade or update was successful, remove the snapshots:

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



IMPORTANT

Removing snapshots is a critical action. Remove the snapshots if you do not intend to revert the nodes, for example, after an upgrade completes successfully. If you retain snapshots on the nodes for too long, they degrade disk I/O performance.

6. If your upgrade or update failed, revert the snapshots:

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

- a. Reboot each node that you reverted so the changes are applied to the filesystem. The revert option automatically deletes the snapshots.

CHAPTER 2. BACKING UP THE UNDERCLOUD AND THE CONTROL PLANE NODES BY USING THE RELAX-AND-RECOVER TOOL

You must back up your undercloud node and your control plane nodes when you upgrade or update your Red Hat Openstack Platform (RHOSP). You can backup your undercloud node and your control plane nodes using the Relax-and-Recover (ReaR) tool. To back up and restore your undercloud and your control plane nodes using the ReaR tool, you must complete the following procedures:

1. Backing up the undercloud node
2. Backing up the control plane nodes
3. Restoring the undercloud and control plane nodes

2.1. BACKING UP THE UNDERCLOUD NODE BY USING THE RELAX-AND-RECOVER TOOL

To back up the undercloud node, you configure the backup node, install the Relax-and-Recover tool on the undercloud node, and then 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.

2.1.1. 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

2.1.2. 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>
```

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

2.1.3. 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.

2.1.4. 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 **<backup_node>**, **<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
```

2.1.5. 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 2.1.4, "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 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**.

- 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
```

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

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

- 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**.

2.1.6. Optional: 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.

You can create a standalone undercloud database backup in conjunction with the ReaR tool and the Snapshot and Revert tool. However, it is recommended that you back up the entire undercloud. For more information about creating a backup of the undercloud node, see [Creating a backup of the undercloud node](#).

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 2.1.8, "Creating a backup of the undercloud node"](#)
- [Section 2.3.4, "Restoring the undercloud node database manually"](#)

2.1.7. 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.

2.1.8. 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.



NOTE

It is recommended that you create a backup of the undercloud node by using the following procedure. However, if you completed [Creating a standalone database backup of the undercloud nodes](#), you can skip this procedure.

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 2.1.4, “Installing and configuring an NFS server on the backup node”](#).
- You have installed ReaR on the undercloud node. For more information, see [Section 2.1.5, “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 2.1.7, “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. Create a backup of the undercloud node:

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

2.1.9. 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

- 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 2.1.4, “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 2.2.3, “Installing ReaR on the control plane nodes”](#).
- You have sufficient available disk space at your backup location to store the backup.

Procedure

1. 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"}
```

2.2. BACKING UP THE CONTROL PLANE NODES BY USING THE RELAX-AND-RECOVER TOOL

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.

2.2.1. 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

2.2.2. 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 **<backup_node>**, **<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
```

2.2.3. 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 2.2.2, "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:

- 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>
tripleo_backup_and_restore_shared_storage_folder: <backup_dir>
```

- Replace **<ip_address>** and **<backup_dir>** with the values that apply to your environment. By default, the **tripleo_backup_and_restore_server** parameter value is **192.168.24.1**.*
- 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**.

2.2.4. 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.

2.2.5. 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 2.2.2, “Installing and configuring an NFS server on the backup node”](#).
- You have installed ReaR on the control plane nodes. For more information, see [Section 2.2.3, “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 2.2.4, “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. 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.

2.2.6. 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 2.1.4, “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 2.2.3, “Installing ReaR on the control plane nodes”](#).
- You have sufficient available disk space at your backup location to store the backup.

Procedure

1. 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"}
```

2.3. RESTORING THE UNDERCLOUD NODE AND CONTROL PLANE NODES BY USING THE RELAX-AND-RECOVER TOOL

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.

2.3.1. 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 2.1.8, "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 [Section 2.1.7, "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> -xzf - '*.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. 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.

2.3.2. Restoring the control plane nodes

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

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

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



NOTE

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

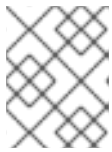
Prerequisites

- You have created a backup of the control plane nodes. For more information, see [Section 2.2.5, "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 2.2.4, "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.

- 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 \(tempest\)](#).

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
```

2.3.3. 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 $(sudo podman container ls --all --format "{{ .Names }}" \
--filter=name=galera-bundle) bash -c "mysql -uroot -e'use mysql;set password for root@localhost = password('$ROOTPASSWORD');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
```

- Restart the MySQL container on all Controller nodes:

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

- Restart the **clustercheck** container on all Controller nodes:

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

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

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

Verification

- 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
```

2.3.4. 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 2.1.6, "Optional: Creating a standalone database backup of the undercloud nodes"](#).

Procedure

- Log in to the director undercloud node as the **root** user.
- Stop all tripleo services:

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

- 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
```