Red Hat OpenStack Platform 12

Back Up and Restore the Director Undercloud

Back up and restore the director undercloud

Last Updated: 2019-03-05

Red Hat OpenStack Platform 12 Back Up and Restore the Director Undercloud

Back up and restore the director undercloud

OpenStack Team
rhos-docs@redhat.com

Legal Notice

Abstract

A guide to backing up and restoring the undercloud in Red Hat OpenStack Platform director.

Table of Contents

CHAPTER 1. ABOUT THIS GUIDE

WARNING

Red Hat is currently reviewing the information and procedures provided in this guide for this release.

This document is based on the Red Hat OpenStack Platform 11 document, available at https://access.redhat.com/documentation/en-us/red_hat_openstack_platform/?version=11.

If you require assistance for the current Red Hat OpenStack Platform release, please contact Red Hat support.

CHAPTER 2. BACK UP AND RESTORE THE UNDERCLOUD

This guide describes how to back up the undercloud used in the Red Hat OpenStack Platform director. The undercloud is usually a single physical node (although high availability options exist using a two-node pacemaker cluster that runs director in a VM) that is used to deploy and manage your OpenStack environment.

2.1. BACKUP CONSIDERATIONS

Formulate a robust back up and recovery policy in order to minimize data loss and system downtime. When determining your back up strategy, you will need to answer the following questions:

- *How quickly will you need to recover from data loss?* If you cannot have data loss at all, you should include high availability in your deployment strategy, in addition to using backups. You'll need to consider how long it will take to obtain the physical backup media (including from an offsite location, if used), and how many tape drives are available for restore operations.
- *How many backups should you keep?* You will need to consider legal and regulatory requirements that affect how long you are expected to store data.
- *Should your backups be kept off-site?* Storing your backup media offsite will help mitigate the risk of catastrophe befalling your physical location.
- *How often should backups be tested?* A robust back up strategy will include regular restoration tests of backed up data. This can help validate that the correct data is still being backed up, and that no corruption is being introduced during the back up or restoration processes. These drills should assume that they are being performed under actual disaster recovery conditions.
- *What will be backed up?* The following sections describe database and file-system backups for components, as well as information on recovering backups.

2.2. HIGH AVAILABILITY OF THE UNDERCLOUD NODE

You are free to consider your preferred high availability (HA) options for the Undercloud node; Red Hat does not prescribe any particular requirements for this. For example, you might consider running your Undercloud node as a highly available virtual machine within Red Hat Enterprise Virtualization (RHEV). You might also consider using physical nodes with Pacemaker providing HA for the required services.

When approaching high availability for your Undercloud node, you should consult the documentation and good practices of the solution you decide works best for your environment.

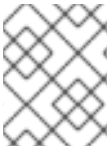
2.3. BACK UP THE UNDERCLOUD

A full undercloud backup includes the following databases and files:

- All MariaDB databases on the undercloud node
- MariaDB configuration file on the undercloud (so that you can accurately restore databases)
- All glance image data in */var/lib/glance/images*
- All swift data in */srv/node*
- All data in the stack user home directory: */home/stack*

- The undercloud SSL certificates:
  - */etc/pki/instack-certs/undercloud.pem*
  - */etc/pki/ca-trust/source/anchors/cacert.pem*
- If SSL is enabled in the overcloud and using a self-signed certificate, the overcloud CA certificate used by the undercloud. For example:
  - */etc/pki/ca-trust/source/anchors/overcloud.pem*
- All **`docker`** directories and files:
  - */var/lib/docker*
  - */var/lib/registry*
  - */etc/docker*
  - */etc/docker-distribution*
  - */etc/sysconfig/docker**

Run the following commands as the root user to dump the data from the undercloud node to a file named *undercloud-backup-[timestamp].tar.gz*.



NOTE

Confirm that you have sufficient disk space available before performing the backup process. The tarball can be expected to be at least 3.5 GB, but this is likely to be larger.

```
# mysqldump --opt --all-databases > /root/undercloud-all-databases.sql
# tar --xattrs -czf undercloud-backup-`date +%F`.tar.gz /root/undercloud-
all-databases.sql /etc/my.cnf.d/server.cnf /var/lib/glance/images
/srv/node /home/stack /etc/pki/instack-certs/undercloud.pem /etc/pki/ca-
trust/source/anchors/cacert.pem
```

2.4. VALIDATE THE COMPLETED BACKUP

You can validate the success of the completed back up process by running and validating the restore process. See the next section for further details on restoring from backup.

CHAPTER 3. RESTORE

This section describes how to restore the undercloud used in the Red Hat OpenStack Platform Director.

3.1. RESTORE THE UNDERCLOUD

The following restore process assumes you are recovering a failed undercloud node, and you need to reinstall it from scratch. It assumes that the hardware layout is identical, and the hostname and undercloud settings of the machine will also be identical.

Once the machine is installed and is in a clean state, re-enable all the subscriptions/repositories needed to install and run director. Run the following commands as the root user:

1. Install the *mariadb* server:

```
# yum install -y mariadb-server
```

2. Restore the MariaDB configuration file and database backup, then start the MariaDB server and load the backup data:

a. As the *root* user, restore the MariaDB files:

```
# tar -xzC / -f undercloud-backup-$DATE.tar.gz etc/my.cnf.d/server.cnf
# tar -xzC / -f undercloud-backup-$DATE.tar.gz root/undercloud-all-
databases.sql
```

b. Edit */etc/my.cnf.d/server.cnf* and comment out the **`bind-address`** entry.

c. Start the *mariadb* service and temporarily update the **`max_allowed_packet`** setting:

```
# systemctl start mariadb
# mysql -uroot -e"set global max_allowed_packet = 16777216;"
# cat /root/undercloud-all-databases.sql | mysql
```

d. Clean up certain permissions (to be recreated later):

```
# for i in ceilometer glance heat ironic keystone neutron nova;do mysql -e
"drop user $i";done
# mysql -e 'flush privileges'
```

3. Create the *stack* user account:

```
# sudo useradd stack
# sudo passwd stack  # specify a password
# echo "stack ALL=(root) NOPASSWD:ALL" | sudo tee -a /etc/sudoers.d/stack
# sudo chmod 0440 /etc/sudoers.d/stack
```

4. Restore the *stack* user home directory:

```
# tar -xzC / -f undercloud-backup-$DATE.tar.gz home/stack
```

5. Install the *swift* and *glance* base packages, and then restore their data:

```
# yum install -y openstack-glance openstack-swift
# tar --xattrs -xzC / -f undercloud-backup-$DATE.tar.gz srv/node
var/lib/glance/images
```

6. Confirm the data is owned by the correct user:

```
# chown -R swift: /srv/node
# chown -R glance: /var/lib/glance/images
```

7. Restore your undercloud SSL certificates:

```
# tar -xzC / -f undercloud-backup-$DATE.tar.gz /etc/pki/instack-
certs/undercloud.pem
# tar -xzC / -f undercloud-backup-$DATE.tar.gz /etc/pki/ca-
trust/source/anchors/cacert.pem
# tar -xzC / -f undercloud-backup-$DATE.tar.gz /etc/pki/ca-
trust/source/anchors/overcloud.pem
# update-ca-trust extract
```

8. Re-run the undercloud installation as the *stack* user, making sure to run it in the *stack* user home directory:

```
# su - stack
$ sudo yum install -y python-tripleoclient
```

9. Confirm that the hostname is correctly set in */etc/hosts*.

10. Reinstall the undercloud:

```
$ openstack undercloud install
```

10. Restore the **`docker`** directories and files and restart the service:

```
# tar -xzC / -f undercloud-backup-$DATE.tar.gz /var/lib/docker
# tar -xzC / -f undercloud-backup-$DATE.tar.gz /var/lib/registry
# tar -xzC / -f undercloud-backup-$DATE.tar.gz /etc/docker
# tar -xzC / -f undercloud-backup-$DATE.tar.gz /etc/docker-distribution
# tar -xzC / -f undercloud-backup-$DATE.tar.gz /etc/sysconfig/docker*
# systemctl restart docker docker-distribution
```

3.2. RECONNECT THE RESTORED UNDERCLOUD TO THE OVERCLOUD

Having completed the steps above, the undercloud can be expected to automatically restore its connection to the overcloud. The nodes will continue to poll Orchestration (heat) for pending tasks, using a simple HTTP request issued every few seconds.

3.3. VALIDATE THE COMPLETED RESTORE

Use the following commands to perform a healthcheck of your newly restored environment:

3.3.1. Check Identity Service (Keystone) Operation

This step validates Identity Service operations by querying for a list of users.

```
# source overcloudrc
# openstack user list
```

When run from the controller, the output of this command should include a list of users created in your environment. This action demonstrates that keystone is running and successfully authenticating user requests. For example:

```
# openstack user list
+----------------------------------+------------+---------+------------
----------+
|                id                |    name    | enabled |        email
|
+----------------------------------+------------+---------+------------
----------+
| 9e47bb53bb40453094e32eccce996828 |   admin    |   True  |
root@localhost    |
| 9fe2466f88cc4fa0ba69e59b47898829 | ceilometer |   True  |
ceilometer@localhost |
| 7a40d944e55d422fa4e85daf47e47c42 |   cinder   |   True  |
cinder@localhost   |
| 3d2ed97538064f258f67c98d1912132e |    demo    |   True  |
|
| 756e73a5115d4e9a947d8aadc6f5ac22 |   glance   |   True  |
glance@localhost   |
| f0d1fcee8f9b4da39556b78b72fdafb1 |  neutron   |   True  |
neutron@localhost   |
| e9025f3faeee4d6bb7a057523576ea19 |    nova    |   True  |
nova@localhost    |
| 65c60b1278a0498980b2dc46c7dcf4b7 |   swift    |   True  |
swift@localhost    |
+----------------------------------+------------+---------+------------
----------+
```

3.3.2. Check the OpenStack Services

Run the *openstack-status* command to view the status of OpenStack services:

```
# openstack-status
```