Back Up and Restore the Director Undercloud

Back up and restore the Red Hat OpenStack Platform director undercloud
Back up and restore the Red Hat OpenStack Platform director undercloud

OpenStack Team
rhos-docs@redhat.com
Abstract

You can back up the database and critical filesystems on your undercloud node and restore them to a newly installed undercloud node.
# Table of Contents

- **MAKING OPEN SOURCE MORE INCLUSIVE** .................................................. 3
- **DIRECT DOCUMENTATION FEEDBACK (DDF) FUNCTION NOT AVAILABLE IN THIS BETA RELEASE** ..... 4
- **CHAPTER 1. BACKING UP THE DIRECTOR UNDERCLOUD** .................................. 5  
  1.1. BACKING UP A CONTAINERIZED UNDERCLOUD ........................................ 5
- **CHAPTER 2. RESTORING THE DIRECTOR UNDERCLOUD** .................................. 7  
  2.1. RESTORING A CONTAINERIZED UNDERCLOUD ........................................ 7  
  2.2. VALIDATING THE UNDERCLOUD RESTORATION .................................... 10
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.
DIRECT DOCUMENTATION FEEDBACK (DDF) FUNCTION NOT AVAILABLE IN THIS BETA RELEASE

The Direct Documentation Feedback (DDF) function allows users to enter feedback directly on documentation pages on fully supported Red Hat documentation products. The DDF function is not available in this Red Hat OpenStack platform 16.2 beta documentation set.
CHAPTER 1. BACKING UP THE DIRECTOR UNDERCLOUD

To minimize data loss and system downtime, you can create and recover backups of the database and critical filesystems that run on the Red Hat OpenStack Platform (RHOSP) director undercloud node. The undercloud is usually a single physical node, although high availability (HA) options exist that use a two-node pacemaker cluster that runs director in a virtual machine instance.

Red Hat does not prescribe any particular requirements for HA on the undercloud node. You can implement undercloud HA that is suitable for your environment. For example, you can run your undercloud node as a highly available virtual machine with Red Hat Enterprise Virtualization (RHEV). You can also use physical nodes with Pacemaker that provide HA for the required services.

To validate the success of the completed backup process, you can run and validate the restoration process. For more information, see Chapter 2, Restoring the director undercloud.

1.1. BACKING UP A CONTAINERIZED 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
- The configuration data: /etc
- Log data: /var/log
- Image data: /var/lib/glance
- Certificate generation data if using SSL: /var/lib/certmonger
- Any container image data: /var/lib/containers and /var/lib/image-serve
- All swift data: /srv/node
- All data in the stack user home directory: /home/stack

Prerequisites

- You have a minimum of 3.5 GB of available space on the undercloud for the archive file.

Procedure

1. Log in to the undercloud as the root user.

2. Retrieve the MySql root password:

   ```
   [root@director ~]# PASSWORD=$(/bin/hiera -c /etc/puppet/hiera.yaml mysql::server::root_password)
   ```

3. Perform the backup:

   ```
   [root@director ~]# podman exec mysql bash -c "mysqldump -uroot -p$PASSWORD --opt -- all-databases" > /root/undercloud-all-databases.sql
   ```
4. Archive the database backup and the configuration files:

```
[root@director ~]# cd /backup
[root@director backup]# tar --xattrs --xattrs-include='*.*' --ignore-failed-read -cf \
    undercloud-backup-`date +%F`.tar \ 
    /root/undercloud-all-databases.sql \ 
    /etc \ 
    /var/log \ 
    /var/lib/glance \ 
    /var/lib/certmonger \ 
    /var/lib/containers \ 
    /var/lib/image-serve \ 
    /var/lib/config-data \ 
    /srv/node \ 
    /root \ 
    /home/stack
```

- The `--ignore-failed-read` option skips any directory that does not apply to your undercloud.
- The `--xattrs` option includes extended attributes, which are required to store metadata for Object Storage (swift).

This command creates a file named `undercloud-backup-<timestamp>.tar.gz`, where `<timestamp>` is the system date. Copy this tar file to a secure location.
CHAPTER 2. RESTORING THE DIRECTOR UNDERCLOUD

You can use your Red Hat OpenStack Platform (RHOSP) undercloud backup to restore the undercloud data on to a newly installed undercloud node in your deployment.

As a result, the restored undercloud uses the latest packages.

2.1. RESTORING A CONTAINERIZED UNDERCLOUD

If the undercloud node in your deployment has failed and is in an unrecoverable state, you can restore the database and critical file systems on to a newly deployed undercloud node.

Prerequisites

- You have created an undercloud backup archive of your director undercloud databases and files. For more information, see Section 1.1, “Backing up a containerized undercloud”
- You have re-installed the Red Hat Enterprise Linux (RHEL) 8.4.
- The new undercloud node has the same hardware resources as the failed node.
- Ensure that, for the new undercloud node in your deployment, you use the same hostname and undercloud settings as the failed node.

Procedure

1. Log in to your new undercloud node as the root user.

2. Register your system with the Content Delivery Network and enter your Customer Portal user name and password at the prompt:

   ```bash
   [root@director ~]# subscription-manager register
   ```

3. Attach the RHOSP entitlement:

   ```bash
   [root@director ~]# subscription-manager attach --pool=<pool_number>
   ```

4. Disable all default repositories, and enable the required RHEL repositories:

   ```bash
   [root@director ~]# subscription-manager repos --disable=* 
   [root@director ~]# subscription-manager repos --enable=rhel-8-for-x86_64-baseos-eus-rpms 
   --enable=rhel-8-for-x86_64-appstream-eus-rpms 
   --enable=rhel-8-for-x86_64-highavailability-eus-rpms 
   --enable=ansible-2.9-for-rhel-8-x86_64-rpms 
   --enable=openstack-beta-for-rhel-8-x86_64-rpms 
   --enable=fast-datapath-for-rhel-8-x86_64-rpms
   ```

5. Set the container-tools repository module to version 3.0:

   ```bash
   [root@director ~]# sudo dnf module disable -y container-tools:rhel8
   [root@director ~]# sudo dnf module enable -y container-tools:3.0
   ```

6. Set the virt repository module to version av:
7. Perform an update of your base operating system:

```
[root@director ~]# dnf update -y
[root@director ~]# reboot
```

8. Ensure that the time on your undercloud is synchronized:

```
[root@director ~]# dnf install -y chrony
[root@director ~]# systemctl start chronyd
[root@director ~]# systemctl enable chronyd
```

9. Copy the undercloud backup archive to the root directory of the newly deployed undercloud node.

10. Install the tar and policycoreutils-python-utils packages:

```
[root@director ~]# dnf install -y tar policycoreutils-python-utils
```

11. Create the stack user:

```
[root@director ~]# useradd stack
```

12. Set a password for the stack user:

```
[root@director ~]# passwd stack
```

13. Configure the stack user account with sudo privileges:

```
[root@director ~]# echo "stack ALL=(root) NOPASSWD:ALL" | tee -a /etc/sudoers.d/stack
[root@director ~]# chmod 0440 /etc/sudoers.d/stack
```

14. Extract the following databases and files, and replace `<timestamp>` with the value of your archive file name:

```
[root@director ~]# tar --xattrs -xvC / -f undercloud-backup-<timestamp>.tar root/undercloud-all-databases.sql var/lib/glance srv/node etc/pki/undercloud-certs/undercloud.pem etc/pki/ca-trust/source/anchors/* etc/puppet home/stack var/lib/config-data/ var/lib/image-serve var/lib/containers --exclude=var/lib/containers/storage/overlay/*/merged/*
```

- `/root/undercloud-all-databases.sql` is the database backup
- `/var/lib/glance` stores the Image service (glance) data
- `/srv/node` stores the Object service (swift) data
- `/etc/pki/undercloud-certs/undercloud.pem` and `/etc/pki/ca-trust/source/anchors/*` stores certificates
- `/etc/puppet` stores the hieradata that has already been generated
- /home/stack stores the stack user data and configuration
- /var/lib/config-data stores configuration container files
- /var/lib/image-serve and /var/lib/containers stores container image database

15. Restore the necessary file properties for the certificates:
   ```bash
   [root@director ~]# restorecon -R /etc/pki
   [root@director ~]# semanage fcontext -a -t etc_t "/etc/pki/undercloud-certs(/.*)?"
   [root@director ~]# restorecon -R /etc/pki/undercloud-certs
   [root@director ~]# update-ca-trust extract
   ```

16. Install the **python3-tripleoclient** and the **ceph-ansible** packages:
   ```bash
   [root@director ~]# dnf -y install python3-tripleoclient ceph-ansible
   ```

17. Delete the containers from the previous undercloud:
   ```bash
   [root@director ~]# podman ps -a --filter='status=created' --format '{{ .Names }}' | xargs podman rm {}
   [root@director ~]# podman ps -a --filter='status=exited' --format '{{ .Names }}' | xargs podman rm {}
   ```

18. To restore the database, complete the following steps:
   a. Create and set the SELinux attributes for the database directory:
      ```bash
      [root@director ~]# sudo mkdir /var/lib/mysql
      [root@director ~]# sudo chown 42434:42434 /var/lib/mysql
      [root@director ~]# sudo chmod 0755 /var/lib/mysql
      [root@director ~]# sudo chcon -t container_file_t /var/lib/mysql
      [root@director ~]# sudo chcon -r object_r /var/lib/mysql
      [root@director ~]# sudo chcon -u system_u /var/lib/mysql
      ```

   b. Create a local tag for the **mariadb** image. Replace `<image_id>` and `<undercloud.ctlplane.example.com>` with the values applicable in your environment:
      ```bash
      [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
      ```
      ```bash
      [root@director ~]# podman images | grep mariadb
      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
      ```

   c. Initialize the /var/lib/mysql directory with the container:
d. Copy the database backup file that you want to import to the database:

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

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

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

f. To import the data and configure the `max_allowed_packet` parameter, you must log in to the container to configure it, stop the container, and ensure that there are no containers running:

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

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

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

19. Set the name of the server and replace `<undercloud.example.com>` with the value applicable in your environment:

```bash
[root@director ~]# hostnamectl set-hostname <undercloud.example.com>
```

20. Run the undercloud installation command:

```bash
[root@director ~]# openstack undercloud install
```

Wait until the installation procedure completes. The undercloud automatically restores the connection to the overcloud. The overcloud nodes continue to poll the OpenStack Orchestration service (heat) for pending tasks.

### 2.2. VALIDATING THE UNDERCLOUD RESTORATION

A robust backup strategy includes regular restoration tests of backed up data. This can help validate that the correct data is backed up and that no corruption is introduced during the back up or restoration processes. To validate that the undercloud restoration process is successful, you can query the Identity service (keystone).
Procedure

1. Log in to the new undercloud host as the stack user.

2. Source the stackrc credentials file:

   ```bash
   [stack@director ~]$ source ~/stackrc
   ```

3. Retrieve a list of users in your environment:

   ```bash
   [stack@director ~]$ openstack user list
   ```

   The output of this command includes a list of users in your environment. This validation demonstrates that the Identity service (keystone) is running and successfully authenticating user requests.

   +----------------------------------+-------------------------------------------------------+
   | ID                               | Name                                                  |
   +----------------------------------+-------------------------------------------------------+
   | f273be1a982b419591ccc7f89c1a5c0d | admin                                                 |
   | a0e1efeb5b654d61a393bcef92c505d2 | heat_admin                                            |
   | 59604e2d56424f9bb4f7c825d0bdc615 | heat                                                  |
   | 35d36ebc2f7043148943d0e070733ed9 | heat-cfn                                              |
   | 233ff3b22c884fa289f7a9a6ec2de326 | neutron                                               |
   | db7af369a9ed4f7fa8d8179ceae323f | glance                                                |
   | d883b3690d7f434d9eb9cab6b5db8f5 | nova                                                  |
   | 3dc52d74feb6402983863c8e9effbf5c | placement                                             |
   | e3bdcc9465254cbea86222191c88edd3 | swift                                                 |
   | 8e68fcc40215446c8c1412fb736522de | ironic                                                |
   | 366ccd100176495cb409dba872516cb2 | ironic-inspector                                      |
   | fe99722603fe424d99d618c368dc0257 | mistral                                               |
   | 5ae21556b4043b6a60208ccd203644a | zaqar                                                 |
   | 83813ec920fe4b01b198770bfa538962 | zaqar-websocket                                       |
   | 5fc6bc527364131b1e36fd4321325e6 | heat_stack_domain_admin                               |
   +----------------------------------+-------------------------------------------------------+