



Protecting your OpenStack cloud with an automated backup and recovery strategy

Carlos Camacho Gonzalez
Senior Software Engineer
Red Hat

Dan Macpherson
Principal Technical Writer
Red Hat

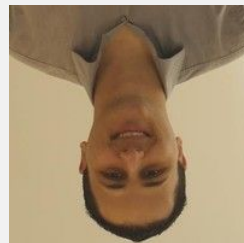
November 14, 2018

Agenda

- Introduction
- Defining the strategy
- Backup and Restoring the Undercloud
- Backup and Restoring the Overcloud
- Challenges and Ideas

Introduction

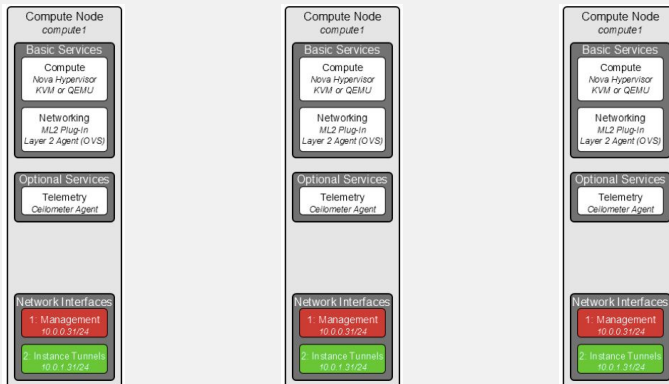
- How did we meet?
- Fast forward upgrades
- Problems we're trying to solve



Backup categories

Protect against maintenance tasks failures (Undercloud, Overcloud control plane)
Protect user space (Trilio, Freezer)

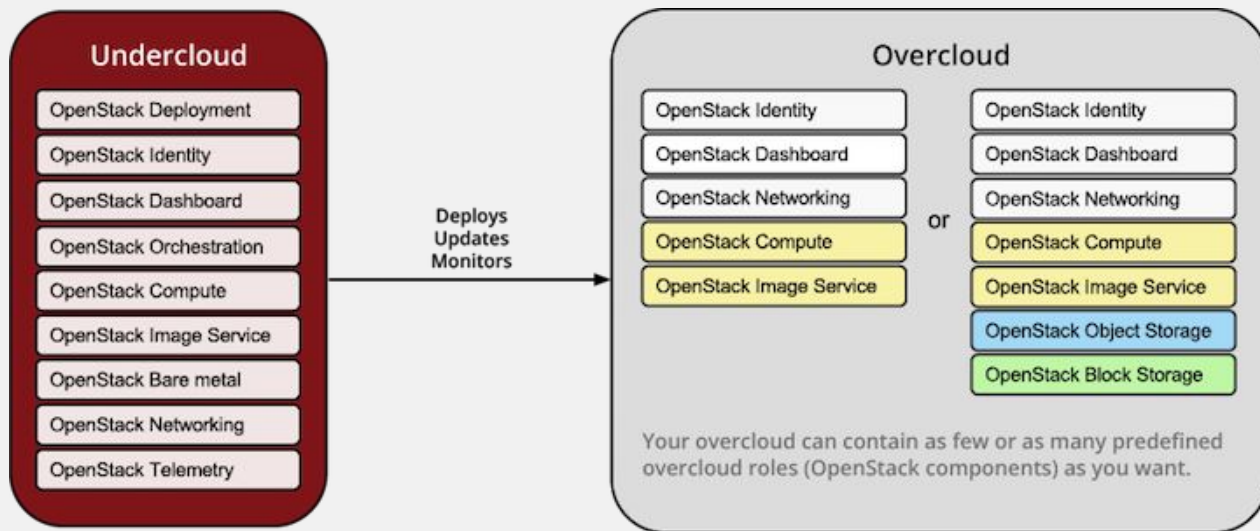
User workload



Backend services



Goal: Ensure you can restore the Undercloud and the Overcloud controllers no matter what... and all automatically!



Defining Backup Strategies for Individual Services

Database (Non-HA)

For example: backing up and restoring the undercloud.

Backup:

- Run the mysqldump command

Restore:

- Create a new database
- Start mariadb
- Increase the packet size
- Restore data from .sql files

Database (HA)

Backup:

- Select an idle node
- Backup the database
- Backup the grants

Restore:

- Disable VIP access to the database (iptables)
- Stop Galera
- Temporarily disable replication
- Create a new database on each node
- Set database permissions (root, clustercheck)
- Synchronize the nodes
- Enable replication
- Start Galera
- Import database and grants
- Restore VIP access (iptables)

MongoDB

Used for Telemetry storage in Newton

Backup:

- `mongodump`
- <https://docs.mongodb.com/manual/reference/program/mongodump/>

Restore:

- `mongorestore`
- <https://docs.mongodb.com/manual/reference/program/mongorestore/>

Redis

Used as an object store for services. TripleO overclouds use it for Telemetry object storage.

“Redis is very data backup friendly since you can copy RDB files while the database is running: the RDB is never modified once produced, and while it gets produced it uses a temporary name and is renamed into its final destination atomically using rename(2) only when the new snapshot is complete.”

- <https://redis.io/topics/persistence>

Backup

- Save the current state (redis-cli bgsave)
- Copy the /var/lib/redis/dump.rdb

Restore

- Stop Redis
- Copy dump.rdb back to /var/lib/redis/
- Start Redis

Pacemaker Configuration

Restore previous pacemaker configuration.

Backup:

- Config backup command (`pcs config backup pacemaker_backup`)
- Creates an archive file with configuration

Restore:

- Stop the cluster (`pcs cluster stop --all`)
- Restore config (`pcs config restore pacemaker_controller_backup.tar.bz2`)
- Start the cluster (`pcs cluster start --all`)

Swift

Swift object data as files. Usually part of a filesystem backup.

Backup:

- Backup object files on each node (usually in /srv/node)
- Don't forget the xattrs (Swift object metadata)
- Backup ringfiles and configuration (/etc/swift)

Restore:

- Restore each node's object files (usually to /srv/node)
- Don't forget the xattrs (Swift object metadata)
- Restore ringfiles and configuration (/etc/swift)
- Restart swift

Always important to include any xattrs option for rsync or tar commands:

```
# tar --xattrs ...  
# rsync --xattrs ...
```

Filesystem Backup

Backup relevant directories in your filesystem. You might need to restore a particular piece of configuration at some point.

Recommended directories:

- /etc/
- /var/lib/<service>/ (e.g. glance, cinder, heat, etc)
- kolla config (e.g. /var/lib/config-data)
- /srv/node/ (don't forget xattrs!)
- /var/log/
- /root/ (contains .my.cnf for root access to database)
- Your cloud admin user directory. In TripleO:
 - /home/stack for the undercloud
 - /home/heat-admin for the overcloud

Undercloud Backup and Restore

Backing up the Undercloud

Virtual node?

Create snapshots.

Baremetal node?

Backup the resources required to restore it back to a consistent state.

OpenStack < Queens

Manual backups based
on either bash or
Ansible.

OpenStack >= Queens

TripleO CLI option
“openstack undercloud
backup”

Backing up the Undercloud

Manual steps:

```
openstack undercloud backup [--add-path ADD_FILES_TO_BACKUP] [--exclude-path EXCLUDE_FILES_TO_BACKUP]
```

```
openstack undercloud backup --add-path /etc/ \  
--add-path /var/log/ \  
--add-path /root/ \  
--add-path /var/lib/glance/ \  
--add-path /var/lib/docker/ \  
--add-path /var/lib/certmonger/ \  
--add-path /var/lib/registry/ \  
--add-path /srv/node/ \  
--exclude-path /home/stack/
```


Backing up the Undercloud

CLI driven:

```
mysqldump --opt --single-transaction --all-databases >  
/root/undercloud-all-databases.sql
```

```
sudo tar --xattrs --ignore-failed-read -cf \  
UC-backup-`date +%F`.tar \  
/root/undercloud-all-databases.sql \  
/etc \  
/var/log \  
/root \  
/var/lib/glance \  
/var/lib/docker \  
/var/lib/certmonger \  
/var/lib/registry \  
/srv/node \  
/home/stack
```

Restoring the Undercloud

Strategy?

- Restore the snapshot or nuke the node and install from scratch [1]

Reasons?

- Transaction history might be hard to rollback after an upgrade
- Single node no HA, easy to reinstall

How to do it?

- Restore the configuration files
- Restore the certificates files
- Restore the databases
- Run: ``openstack undercloud install``

[1]:https://docs.openstack.org/tripleo-docs/latest/install/controlplane_backup_restore/03_undercloud_restore.html

Overcloud Backup and Restore

Overcloud Backup and Restore Strategy

- Composable and agnostic automated backup and restore system
- Ansible role - ansible-role-openstack-operations [1]
- Foundational ansible tasks [2]
 - Allows you to set an external backup server and automatically configure it
 - Bootstrap node assignment
 - Ansible synchronize module (rsync wrapper)
 - Provides temporary SSH access to nodes
 - Tasks for database backup
 - Tasks for database restore (containerized HA)
 - Tasks to validate the database
- Future goals:
 - More services (Pacemaker, Redis, Swift, etc)
 - Different backend architectures (Non-HA, non-containerized)

[1] <http://git.openstack.org/cgiit/openstack/ansible-role-openstack-operations/>

[2] <https://review.openstack.org/#/c/604439/>

Backing up the overcloud

```
---
- name: Initialize backup host
  hosts: "{{ backup_hosts | default('backup') }}"[0]"
  Tasks:
    - import_role:
      name: ansible-role-openstack-operations
      tasks_from: initialize_backup_host

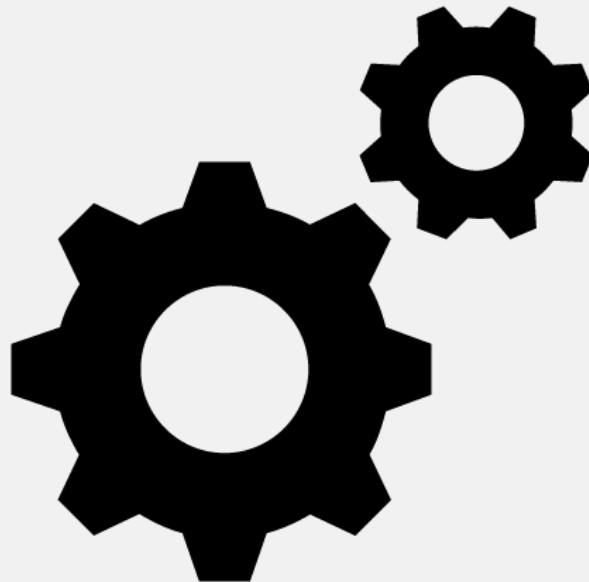
- name: Backup MySQL database
  hosts: "{{ target_hosts | default('mysql') }}"[0]"
  vars:
    backup_server_hostgroup: "{{ backup_hosts | default('backup') }}"
  tasks:
    - import_role:
      name: ansible-role-openstack-operations
      tasks_from: validate_galera
    - import_role:
      name: ansible-role-openstack-operations
      tasks_from: enable_ssh
    - import_role:
      name: ansible-role-openstack-operations
      tasks_from: backup_mysql
    - import_role:
      name: ansible-role-openstack-operations
      tasks_from: disable_ssh
```

Restoring the overcloud

```
---
- name: Initialize backup host
  hosts: "{{ backup_hosts | default('backup') }}"[0]
  tasks:
    - import_role:
      name: ansible-role-openstack-operations
      tasks_from: initialize_backup_host

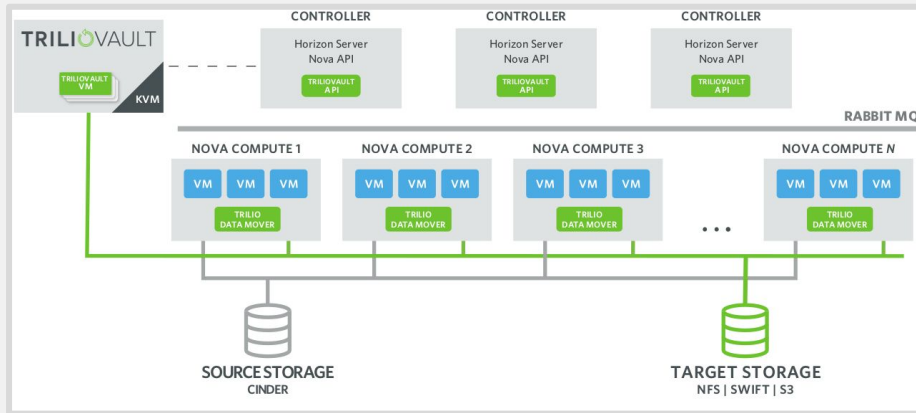
- name: Restore MySQL database on galera cluster
  hosts: "{{ target_hosts | default('mysql') }}"
  vars:
    backup_server_hostgroup: "{{ backup_hosts | default('backup') }}"
  tasks:
    - import_role:
      name: ansible-role-openstack-operations
      tasks_from: set_bootstrap
    - import_role:
      name: ansible-role-openstack-operations
      tasks_from: enable_ssh
    - import_role:
      name: ansible-role-openstack-operations
      tasks_from: restore_galera
    - import_role:
      name: ansible-role-openstack-operations
      tasks_from: disable_ssh
    - import_role:
      name: ansible-role-openstack-operations
      tasks_from: validate_galera
```

Overcloud restore demo

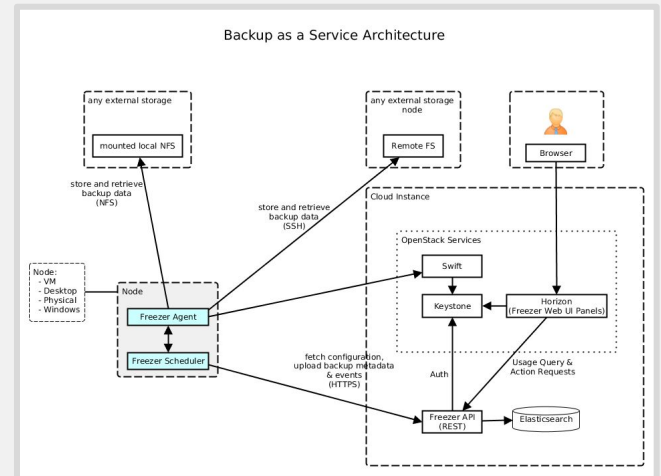


User workloads

Trilio



Freezer



Challenges

- Testing.
- Adapting the tasks to several versions and services.
- Maintenance over new releases.

Ideas

- Including the ansible tasks per service configuration template.
- Create an additional repository to store the backup/restore workflow.
- Composable backups.
- Each squad testing their own backup/restore methodology.
- Create a new CLI option to backup the Overcloud controllers?
 - `openstack overcloud backup --controllers`
- TripleO UI options?



THANK YOU



plus.google.com/+RedHat



facebook.com/redhatinc



linkedin.com/company/red-hat



twitter.com/RedHatNews



youtube.com/user/RedHatVideos