RDO and Tripleo: exploit ansible to customize both the Undercloud and the Overcloud
A journey through the installation of a hyperconverged OpenStack cluster

Franca Debole  Andrea Dell'Amico
CNR, ISTI InfraScience laboratory
franca.debole@isti.cnr.it andrea.dellamico@isti.cnr.it

CS3 2023 Barcellona
6-8 March 2023
Overview

1 Where we started
   SaaS Oriented Research projects
   An old infrastructure looking to the future

2 So, OpenStack
   Why OpenStack
   How Tripleo works

3 And now
   Our setup
   Our contribution

4 The outcome
   Our OpenStack on premise
   Headaches

5 The future
   Federation? Still OpenStack?
D-Net is a research group that develops the core of the OpenAIRE https://www.openaire.eu services. Big data for real, we host the development infrastructure and services.

D4Science https://www.d4science.org/ is a Data Infrastructure connecting +15,000 scientists in +50 countries, integrating +50 heterogeneous data providers, executing +55,000 models & algorithms/month. It hosts +175 Virtual Research Environments (VREs) to serve the biological, ecological, environmental, social mining, culture heritage, and statistical communities world-wide.
The needs of a large computing and storage infrastructure

The old computing and storage infrastructure is based on Xen and ATA o Ethernet:

• requires a lot of manual effort to setup machines and services for the users;
• the services made available to the institute and to the research partners require a significative amount of computing and storage resources.

**OBJECTIVE**: have research services available in a simple way, fast, secure eccetera.
Where we started
So, OpenStack
And now
The outcome
The future

SaaS Oriented Research projects
An old infrastructure looking to the future

What now: on premise OpenStack

We wanted flexibility, a set of APIs that could be also used out of our institute, and the possibility to federate our resources.

A lot of services: Rstudio, JupyterHub, TagME, Geoserver/Geonetwork.
So, OpenStack ...

We chose a CentOS base distribution for the baremetal and RDO. The Ansible [link](https://www.rdoproject.org/) was a natural choice.

Debole, Dell’Amico
and TripleO

*TripleO* is a set of tools for the deployment and management of OpenStack which utilizes the *RDO* repositories.

- *TripleO* Stable versions: *Wallaby*, then *Zed*
- Uses a mix of *Puppet* and *Ansible*
- *Ceph* can be installed either using *ceph-Ansible* or *cephadm*
- Everything is containerized
TripleO requires a dedicated machine, called *undercloud*. It installs a all-in-one OpenStack into it, and then uses that OpenStack installation to deploy the main (*overcloud*) OpenStack.

The *undercloud* uses *ironic* to install the physical machines, and then configure the *OpenStack* services.

To do that, we must tell the *undercloud* which machines to install, what roles (Controller, Compute, Storage, HCI, etc.)
The architecture

- Hyperconverged configuration (HCI), to simplify the hardware choices
- Separate networks for different needs:
  - VLANs, and 6 network interfaces on each server
  - Bonding on the Linux side, LACP on the switches
Our contribution: Ansible to help on the installation of Undercloud and the Overcloud

We use Ansible, [https://www.ansible.com](https://www.ansible.com) to distribute the configuration files required by the undercloud, the overcloud, and for some tasks after the overcloud deployment completed successfully.
Preparing for Undercloud deployment

The *undercloud* wants the list of the baremetal nodes to extract information from them:

- Network interfaces configuration (other than the main one)
- Undercloud main configuration file
- List of the target nodes, with their BMC credentials
Preparing for Overcloud deployment

The *overcloud* deployment requires a lot of information, that we provide populating template files.

Controllers and Compute nodes

- Hosts roles
- Enabled services and their configuration
- Controller nodes IP addresses and hostnames
- Special images configuration (machines with GPUs, for example)

Ceph configuration

- Ceph installation technology (*ceph-ansible* vs *cephadm*)
- Ceph services
- List of storage nodes
- List of OSDs for each nodes
- HCI setups: reserve RAM and CPU

Debole, Dell’Amico (ansible for the Undercloud and the Overcloud)
After the deploy

When the *overcloud deploy* completes successfully, a lot of work has still to be done [customization]

- Keystone OIDC: transparent association between users and projects
- Public endpoints (HAPROXY): certificates managed by Letsencrypt
- Creation of a basic set of flavours
- Upload a set of ready to use Linux distribution images
- Setup a backup procedure for both the undercloud and the controller nodes
- External Prometheus to gather metrics from the baremetal nodes
What went well ➞ hypervisors

Hypervisor run and going, dashboard to allow is quite easy self-service on baremetal.
What went well \(\rightarrow\) data services/backbone services

Researchers and technicians enjoy it 😊
What went wrong

Almost *everything*

- Documentation: often obsolete or missing or confusing
- Code bugs, each time our configuration did not match the setups tested by the developers (example: bonding with vlans)
- The use of both *Puppet* and *Ansible* makes the troubleshooting more complicated than it should be
- The upgrade procedure only works under specific conditions (tested: *Victoria* ⇒ *Wallaby*)
Next steps

- Integrate a multitenant Kubernetes cluster (*OKD* [https://www.okd.io?](https://www.okd.io?))
- Share
  - Publish our ansible roles and documentation (almost ready)
  - Federation of our cluster with other institutions (INFN, CERN 😊)?
  - Contribute back to the *OpenStack* community
  - Share/collaborate with the GÉANT community 😊
The End

Questions? Comments?