Automating cloud deployment infrastructures with Kolla

Cacheiro, Javier[1] Feijoo Fraga, Alejandro[1] Díez, Rubén[1] Fernandez Sanchez, Carlos[1] Villasuso, Juan[1]

[1]Centro de Supercomputación de Galicia (CESGA)

IBERGRID
October 10-13, 2022

Outline

Introduction

Introduction

OpenStack deployment challenges

- Deployment of a **production grade** OpenStack based infrastructure is a complex task.
- Previous planing (mostly related to network) is mandatory.
- Regarding the configuration: OpenStack is modular and versatile. The cost for this is an interdependence in the configuration of different services: (more variables that degrees of freedom: "*playing card castle*" architecture).
- Interrelation between services make maintenance difficult.

Helpful strategies

- Specify configuration in a single site and use automatism to generate the configuration of the different services: `ansible`
- Use some kind of container technology for manage services: `docker`
- Redundancy and reliability: `HA`

We selected `kolla-ansible` as deployment frame work.

Requirements

Requirements

The main goal was overcome the shortcomings found in the previous OpenStack installed at CESGA.

- Unify all cloud orchestrators running at CESGA (OpenNebula, CloudStack, ...) for better resource optimization.
- Provide HA services.
- Use CEPH as main storage solution (also with HA).
- Easy maintenance.
- Easy addition of new resources (compute nodes).
- Better administration using OpenStack domains.

Hardware

Hardware

Computing

- 114 nodes with 48 cores, 192GB memory, 960GB storage.
- Total: 5.472 cores, about 21TB memory, about 100TB storage.

Storage (CEPH)

- 16 nodes with 48 cores, 192GB memory, 193TB storage.
- Total: 768 cores, about 3TB memory, about 3PB storage.

Cloud nodes

Storage: CEPH

Ceph Overview

Ceph is a parallel file system very well suited for cloud infraestructure:

- It is able to provide block storage (through *librbd*)
- It is also able to provide S3/Swift object storage (through *RGW*)
- It scales very well
- It is an open-source project with a very nice community behind
- It has a powerful management CLI

Ceph Installation

Our deployment:

- Based on Ceph Pacific
- Deployed using *cephadm* using Docker containers
- HA with 5 MON nodes
- 16 data nodes: 12x16TB SAS + 1.6TB NVMe
- 192 OSDs using *bluestore*
- We use the NVMe disk for *bluestore DB*
- 8 RGW daemons
- Total storage available: 2.8PB

Ceph Tips

What we have learned:

- Ceph is very stable
- If you plan to use the DiskPrediction module choose SATA disks (unfortunately SAS disks do not expose standard smart attributes)
- Probably 3 MON nodes are enough because they are very stable
- *bluestore* works very well
- The NVMe disk we use for *bluestore DB* in practice does not add a lot of performance to the solution but it added additional complexity to the configuration and makes maintenance more difficult

Deployment with Kolla

Overview Kolla Deployment

In a nutshell:

1. Installation of kolla-ansible (xena version).
2. Work on deployment configuration files
   - **passwords.yml** file.
   - **multinode** file.
   - **globals.yml** file.
3. kolla deployment

Kolla-ansible installation

Use python virtual environment.

- `pip install git+https://opendev.org/openstack/kolla-ansible@stable/xena`
- Create `/etc/kolla` directory.
- Copy configuration templates (multinode, globals.yml and passwords.yml)

Kolla-ansible configuration

Use python virtual environment.

- Fill inventory (file `multinode`).
- Generate random passwords (file `/etc/kolla/passwords.yml`) using the command `kolla-genpwd`.
- Modify global configuration (file `/etc/kolla/globals.yml`) according your installation.

Kolla-ansible deployment installation

Use python virtual environment.

- Bootstrap servers `kolla-ansible -i ./multinode bootstrap-servers`.
- Pre-deployment checks `kolla-ansible -i ./multinode prechecks`.
- OpenStack deployment `kolla-ansible -i ./multinode deploy`.

Fedcloud integration

Template patch

Fedcloud integration using kolla-ansible "way" is not directly possible and require some modifications in the templates used by kolla-ansible.

"Manual´´ intervention

Due kolla have not a mechanism to setup some fedcloud specific conf in "wsgi-keystone.conf", these can be hardcoded in the kolla "wsgi-keystone.conf.j2" template file.

Confession of sins: this a quick and dirty fix. The correct way: add support for addresses these variables in the kolla code...

Keep in mind possible bugs

Kolla-ansible project is very active and with rapid changes. Test it in all the possible scenarios and combinations of configurations is not possible. So little bugs and typo in the code seems to be usual.

These little bugs deepens in the particular considered version. As and **example**, here are the bugs we found for the kolla-ansible version (13.0.1) we used:

- A missing "=" in variable assignation in the file "register_identity_providers.yml".
- Missing conditional in file "config-federation-oidc.yml" that forced to setup an unnecessary variable.

Main configuration

The main configuration file is “globals.yml´´. Here the sections you must **add** for EGI federation support.

```
[.....]
keystone_identity_providers:
  - name: "egi.eu"
    openstack_domain: "egi"
    protocol: "openid"
    identifier: "https://aai.egi.eu/auth/realms/egi"
    public_name: "OpenID Connect"
    attribute_mapping: "egi-mapping"
    metadata_folder: "/etc/kolla/openid/metadata"
keystone_identity_mappings:
  - name: "egi-mapping"
    file: "/etc/kolla/openid/egi-mapping.json"
```

The name “egi.eu” is mandatory.

metadata folder

The `metadata` folder contains the necessary files for multiple identity providers support.

Please note

The kolla support for multiple identity provider is mandatory (unique identity provider not contemplated), but **it only works** for authentication from horizon and **it does not work** for the protocol used by robot authentication (as nagios probes). There are two possibilities to address this:

- Use ESACO (not tested, but should works)
- Put the necessary `OIDCOAuth*` variables in the `wsgi-keystone.conf` file. This is the solution we used.

So: We use the kolla default method (multiple identity providers, but with only one configured) for horizon authentication **and** hard coded configuration in `wsgi-keystone.conf` for robot authentication.

metadata files

Three files for each identity provider are necessary in the `metadata` folder. The main part of their names **must** match the URL-encoded name of the *identifier*. For EGI keycloak these should be:

- `aai.egi.eu%2Fauth%2Frealms%2Fegi.client`
- `aai.egi.eu%2Fauth%2Frealms%2Fegi.conf`
- `aai.egi.eu%2Fauth%2Frealms%2Fegi.provider`

metadata/aai.egi.eu%2Fauth%2Frealms%2Fegi.client

metadata/aai.egi.eu%2Fauth%2Frealms%2Fegi.client

```
{
  "client_id":"xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
  "client_secret":"XXXXXXXXXXXXXXX"
}
```

metadata/aai.egi.eu%2Fauth%2Frealms%2Fegi.conf

metadata/aai.egi.eu%2Fauth%2Frealms%2Fegi.conf

```
{
  "response_type":"code",
  "scope": "openid profile email eduperson_entitlement"
}
```

metadata/aai.egi.eu%2Fauth%2Frealms%2Fegi.provid

metadata/aai.egi.eu%2Fauth%2Frealms%2Fegi.provider

This file can be downloaded from the *provider metadata URL*:

```
# wget https://aai.egi.eu/auth/realms/egi/.well-known/openid-configu
-O aai.egi.eu%2Fauth%2Frealms%2Fegi.provider
```

Mapping: the egi-mapping.json (1/2)

```
[ {
    "local": [
      {
        "domain": {
          "name": "egi"
        },
        "user": {
          "name": "{1} {0}",
          "email": "{2}",
          "domain": {
            "name": "egi"
          }
        },
        "group": {
          "name": "egi-eosc-synergy.eu",
          "domain": {
            "name": "egi"
          }
        }
      }
    ],
```

Mapping: the egi-mapping.json (2/2)

```
  "remote": [
    {
      "type": "HTTP_OIDC_SUB"
    },
    {
      "type": "HTTP_OIDC_ISS",
      "any_one_of": [
        "https://aai.egi.eu/auth/realms/egi",
        "https://aai.egi.eu/oidc/"
      ]
    },
    { "type": "OIDC-name"},
    { "type": "OIDC-email"},
    {
      "type": "OIDC-eduperson_entitlement",
      "regex": true,
      "any_one_of": [
        "^urn:mace:egi.eu:group:eosc-synergy.eu:role=vm_operator#aai.egi.eu$"
      ]
    }
  ]
}]
```

User identification

Overview | Role assignments | Groups

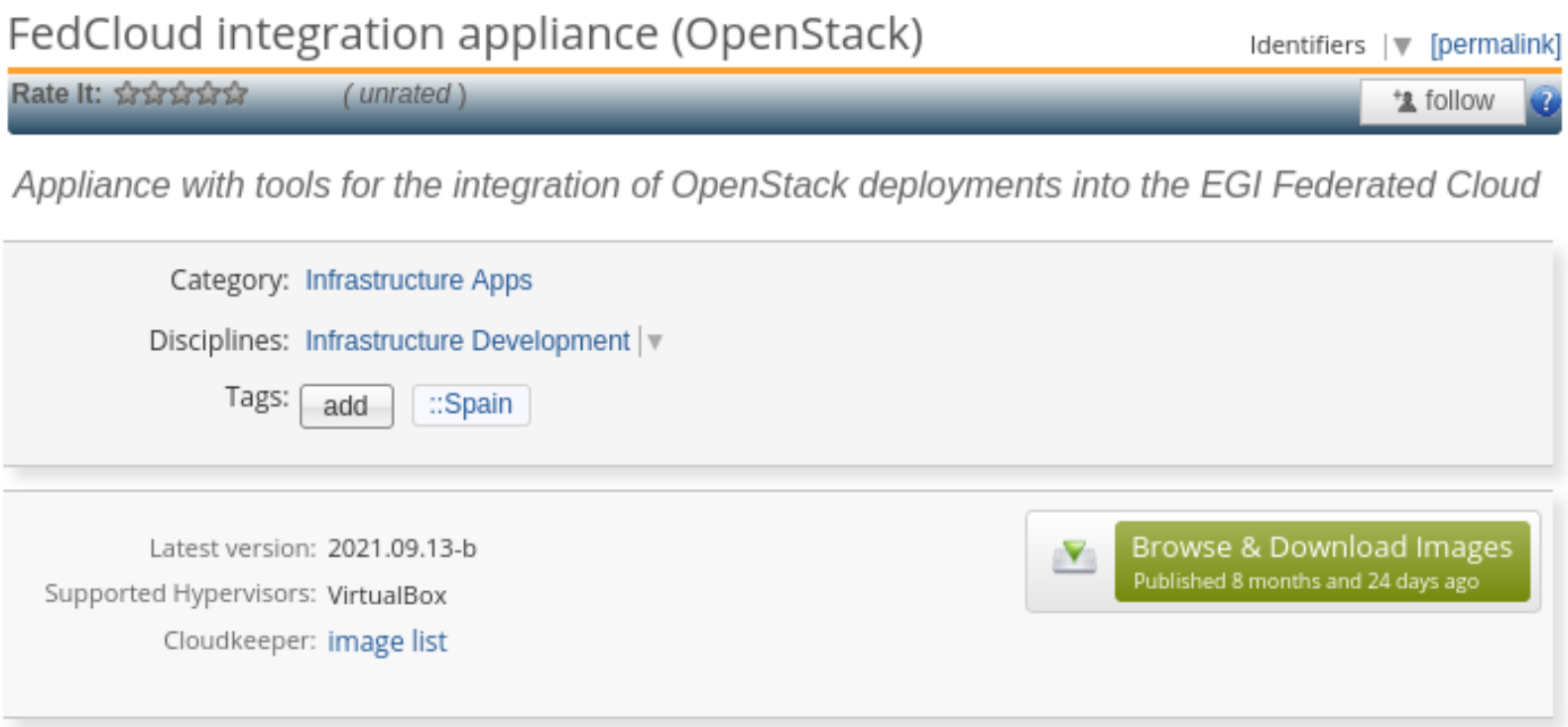
Name	Ruben Diez-Lazaro 3ffe27740a93b9955a2405ec4b9fdf3abbd29cc9082a920e6fdd1b466686cb6f@egi.eu
ID	a92947788f45f51ba08b079505788b0b84750ff53f02fda21c15fd1fdd5a4d86
Domain Name	egi
Domain ID	9170ffcc773745f5a4c0247da13e0ba2
Description	-
Email	rdiez@cesga.es
Enabled	Yes
Password Expires At	None
Lock password	No
Primary Project	-

Auxiliary services

What about auxiliary services?

- CloudKeeper
- Cloud Info Provider
- cASO
- SSM

FedCloud Integration Appliance



Conclusion

Conclusion

- A semi automated tool for installing `OpenStack`, like `kolla`, is a great help...
- ...but it's not a silver bullet.
- The more helpful characteristics of `kolla-ansible` are:
  - Avoid redundancy in configuration; this is: avoid possible configuration inconsistencies.
  - The use of docker containers facilitates management and possible disaster recovery.
  - Facilitates a high availability deployment.
  - Facilitates apply changes in configuration.

BONUS: Why not use k8s?

Use a OpenStack deployment tool based in k8s instead of one based in docker can be considered, but:

- In the time when the installation was planned (about 1.5 year ago) no mature k8s based tool was available.
- Is a "fine grade dockerization" based in k8s really necessary?? May be this would be "over engineering" (KISS principle).
- We have a bigger knowledge in docker that in k8s.

References I

Kolla installation Quick Start https://docs.openstack.org/kolla-ansible/wallaby/user/quickstart.html#install-kolla-ansible

Keystone - Identity service https://docs.openstack.org/kolla-ansible/xena/reference/shared-services/keystone-guide.html#federated-identity

OpenID Client Migration to Keycloak https://docs.egi.eu/providers/cloud-compute/openstack/aai/#client-migration-to-keycloak

FedClod Integration Appliance https://appdb.egi.eu/store/vappliance/fedcloud.integration.appliance.openstack

Thank you!

E-mail: grid-admin@cesga.es