

Administrer votre OpenStack sans aspirine avec Kolla

- RÉMI CAILLETAUD (OSUG - UAR 832 / CNRS)
- ANTHONY DEFIZE (GRICAD - UAR 3458 / UGA)
- GABRIEL MOREAU (LEGI - UMR 5519 / CNRS)
- NICOLAS GIBELIN (GRICAD - UAR 3458 / CNRS)

🔍 Projet Nova

Multi-tutelles

- Université Grenoble Alpes
- Grenoble INP
- CNRS
- INRIA Rhône-Alpes

Objectifs

- OpenStack
- Reproductible
- Haute disponibilité
- Simple

- Infrastructure OpenStack standard + Kolla Ansible



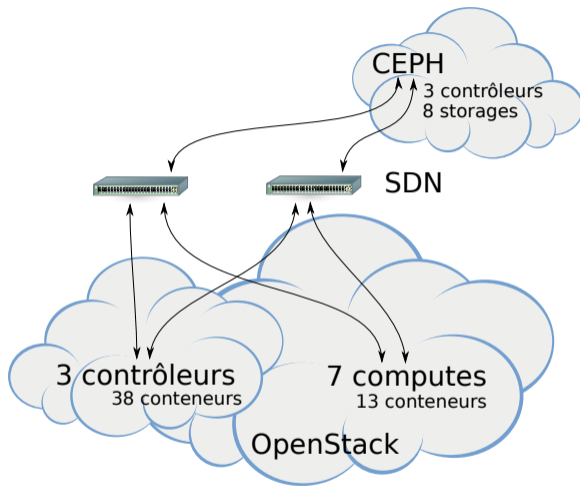


Figure 1: Vue globale de l'architecture

🎯 Projet Kolla

Objectifs

- Fournir les conteneurs et les outils de déploiement pour opérer les clouds OpenStack
- Plan de contrôle hautement disponible
- Infrastructure immuable
- Choix de la distribution (**CentOS**, **Debian**, ~~Oracle Linux~~, **RHEL**, **Ubuntu**)
- Dépendances limitées (**Ansible**, **Docker**, **docker-py**)
- Configuration fonctionnelle par défaut et complètement personnalisable



Source <https://www.stackalytics.io> : « module Kolla Official commits »

Commits by Company

#	Company	Commits
1	StackHPC	2808
2	99cloud	1809
3	Red Hat	1512
4	Cisco Systems	1155
	*independent	764
5	Oracle	683
6	Rackspace	507
7	Linaro	495
8	University of Bialystok	486
9	Servosity	444

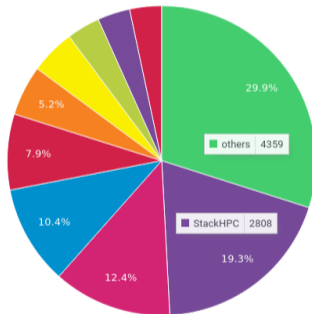


Figure 2: Toutes versions (Kolla depuis 2015 'Liberty')

Composants principaux

- HAProxy et Keepalived pour les endpoints
- MariaDB en mode Galera pour les base MySQL
- Fluentd pour la collecte de logs
- Elasticsearch et Kibana pour l'analyse de logs
- Chrony pour le NTP
- Les services nécessaires classiques : RabbitMQ, Memcached, Open vSwitch
- Les composants OpenStack désirés

Des images Docker

Kolla

- Les principaux projets (38 actuellement)
- Une image par *microservice*, ex: `neutron_server`, `neutron_dhcp_agent`, `neutron_13_agent`, ...
- Image de base au choix (CentOS, Debian, RHEL, Ubuntu)
- Disponible en mode *binaire* (`apt/dnf`) ou *source* (`pip` depuis les sources)
- Personnalisation possible

Exemples d'image : keystone-fernet

```
FROM {{ namespace }}/{{ image_prefix }}keystone-base:{{ tag }}
{% block labels %}
LABEL maintainer="{{ maintainer }}" name="{{ image_name }}" build-date="{{ build_date }}"
{% endblock %}

{% block keystone_fernet_header %}{% endblock %}

{% import "macros.j2" as macros with context %}

{% if base_package_type == 'rpm' %}
    {% set keystone_fernet_packages = [
        'cronie', 'openssh-clients', 'rsync'
    ] %}
{% elif base_package_type == 'deb' %}
    {% set keystone_fernet_packages = [
        'cron', 'openssh-client', 'rsync'
    ] %}
{% endif %}
{{ macros.install_packages(keystone_fernet_packages | customizable("packages")) }}

COPY fetch_fernet_tokens.py /usr/bin/
COPY keystone_bootstrap.sh /usr/local/bin/kolla_keystone_bootstrap
COPY extend_start.sh /usr/local/bin/kolla_extend_start
RUN chmod 755 /usr/local/bin/kolla_extend_start /usr/local/bin/kolla_keystone_bootstrap /usr/bin/fetch_fernet_tokens.py

{% block keystone_fernet_footer %}{% endblock %}
{% block footer %}{% endblock %}
```



Personnaliser les images

Surcharge de l'image de base

```
{% extends parent_template %}  
  
{% set neutron_server_packages_append = ['python-pip'] %}  
  
{% block neutron_server_footer %}  
RUN pip install externaldns-solidserver==0.1.6  
{% endblock %}
```

Construction des images

```
$ kolla-build --config-file kolla-build.conf \  
  --template-override template-overrides.j2 neutron-server
```

Ⓐ Des playbooks Ansible : Kolla Ansible

Ansible main.yml

- Fichiers de configuration `config.yml`
 - Fusion des personnalisations et des valeurs par défaut
 - Copie de `config.json` pour chaque service
 - Copie des fichiers de configuration pour chaque service
- Bootstrap `bootstrap.yml`
 - Création de la base de données et de l'utilisateur
 - Bootstrap conteneur (`db sync`)
- Démarrage des services `start.yml`

A Des playbooks Ansible : Kolla Ansible

Configuration générale

- Liste des nœuds et de leurs rôles dans `inventory.yml`
- Configuration du déploiement dans `globals.yml`
 - Saveurs des conteneurs et images personnalisées
 - Version d'OpenStack
 - Options réseaux
 - Services à déployer
 - ...

👁 Exemple de configuration

```
nova-compute
```

```
[DEFAULT]
```

```
vnc_keymap=fr
```

```
[libvirt]
```

```
cpu_mode = custom
```

```
cpu_model = Broadwell-noTSX-IBRS
```

```
images_rbd_pool=ephemeral-vms
```

```
images_type=rbd
```

```
images_rbd_ceph_conf=/etc/ceph/ceph.conf
```

```
rbd_user=nova
```

```
disk_cachemodes='\"network=writeback\"'
```

```
{% if gpu is defined %}
```

```
[devices]
```

```
enabled_vgpu_types = {{ gpu }}
```

```
{% endif %}
```

1 - Kolla / ELK

Gestion de logs

- OpenStack ↔ beaucoup de logs
- Chaque conteneur Kolla ↔ plusieurs logs
- Syslog centralisé mais difficilement exploitable

Fluentd

- Collecteur de données
- Kolla supporte Fluentd et Elasticsearch
- 2 modes d'installation
 - Kolla déploie Elasticsearch et Kibana (documenté)
 - Utilisation d'une base ELK externe (pas vraiment documenté)

1 - Kolla / ELK

Problème

- Manque de documentation

Simplicité

- Toute la configuration dans un fichier
- Ansible et Jinga2 simple et structuré
- Fluentd se charge de remplir la base Elasticsearch
- Configuration de Fluentd : `td-agent.conf`



1 - Kolla / ELK : à défaut de documentation

kolla-ansible/ansible/roles/common/templates/td-agent.conf.j2
kolla-ansible/ansible/roles/common/templates/conf/output/01-es.conf.j2

```
<match *.*.*>
  @type copy
  <store>
    @type elasticsearch
    host {{ elasticsearch_address }}
    port {{ elasticsearch_port }}
    scheme {{ fluentd_elasticsearch_scheme }}
{% if fluentd_elasticsearch_path != '' %}
    path {{ fluentd_elasticsearch_path }}
{% endif %}
{% if fluentd_elasticsearch_scheme == 'https' %}
    ssl_version {{ fluentd_elasticsearch_ssl_version }}
    ssl_verify {{ fluentd_elasticsearch_ssl_verify }}
{% if fluentd_elasticsearch_cacert | length > 0 %}
    ca_file {{ fluentd_elasticsearch_cacert }}
{% endif %}
{% endif %}
{% if fluentd_elasticsearch_user != '' and fluentd_elasticsearch_password != '' %}
    user {{ fluentd_elasticsearch_user }}
    password {{ fluentd_elasticsearch_password }}
{% endif %}
  [...]
</store>
</match>
```



📁 1 - Kolla / ELK : configuration

global.yml

```
enable_fluentd: "yes"
elasticsearch_address: "elastic.example.fr"
fluentd_elasticsearch_path: "/mon-elastic/"
elasticsearch_port: 443
fluentd_elasticsearch_user: "elastic"
fluentd_elasticsearch_scheme: "https"
fluentd_elasticsearch_ssl_verify: "false"
```

password.yml

```
fluentd_elasticsearch_password: xxxxx
```


2 - Mise à jour : Queens vers Rocky

Mise à jour

- Test sur infrastructure minimale
 - Pas de problème
- Configuration :
 - fusion de configuration
 - `global.yml` et de l'existant
 - `password.yml` et de l'existant (chiffré sur le gitlab)
 - `kolla-ansible -i inventory.yml pull`
 - `kolla-ansible -i inventory.yml upgrade`
 - attendre ...

Résultat

- Tout semble ok
- Quelques paquets perdus sur les machines virtuelles
- Les nouveaux conteneurs tournent

2 - Débogage

Mais

- L'authentification de fonctionne pas

Débogage

- Les logs de Keystone confirment
- Les jetons d'authentification ne sont pas distribués
- Et nouvelle technologie : `fernet_token`

Sur un contrôleur

```
docker exec -it keystone_fernet bash
```

Finalement

- Non documenté
- *A posteriori* dans une 'issue' : migrer vers les tokens Fernet avant mise à jour OpenStack
- Mais simple à déboguer.

Tips

- Mieux si l'hôte et les images sont sur le même OS
 - (`Host_OS == Ubuntu && Image == Centos`) \leftrightarrow désagréments
- Sauvegarder la base MariaDB
 - `kolla-ansible mariadb_backup`
 - `kolla-ansible mariadb_recovery`

★ En conclusion

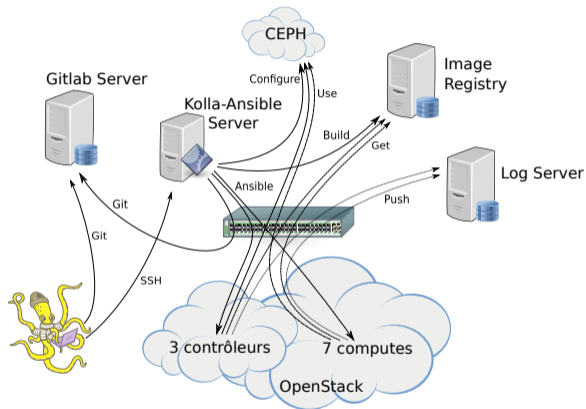


Figure 3: Vue globale en production

★ En conclusion



- Documentation pas toujours présente
- Connaissance de Docker et d'Ansible indispensable
- Lecture du code Ansible



- Configuration centralisée et facilement versionable
- Dépendances minimales
- Procédure de mise à jour facilitées
- Facilité de personnalisation
- Reproductibilité ↔ test de déploiement facile
- Ansible/Jinga2 facile à lire
- Projet actif et qui évolue
- Montée/saut de version OpenStack simple

★ En conclusion

📅 À venir

- Mise à jour vers **Wallaby**
- Bare Metal As a Service : **Ironic**
- Monitoring interne
 - **Prometheus**
 - **OpenStack Exporter**
 - Node Exporter (actuellement installation manuelle)
 - **Alertmanager**



Questions !



Pour aller plus loin

- 2022 : Kolla
 - <https://wiki.openstack.org/wiki/Kolla>
- 2019 : Nova - Un nuage arc-en-ciel au-dessus des Alpes
 - <https://hal.archives-ouvertes.fr/hal-02387868>
- 2017 : Production Deployment Tools for IaaS: an Overall Model and Survey
 - <https://doi.org/10.1109/FiCloud.2017.51>