TP3 Grid & Cloud Computing

Lancer de rayons dans les nuages

Préambule

L'environnement Openstack à votre disposition est un cloud privé *all in one* (tous les services sur une seule machine). La version installée d'Openstack est Queens [1], distribuée par le projet RDO [2] (la version communautaire de la plateforme Openstack de Red Hat). Il y a donc un seul serveur pour héberger tous les composants :

- 56 coeurs logiques, 64 Go de RAM, 1 To de disque
- IP : 192.168.53.248
- DNS : testbed.mesocentre.uca.fr

Le dashboard est accessible à l'adresse : `https://testbed.mesocentre.uca.fr`.
Les ressources (VM, Images, Volumes, Stockage objet...) sont organisées par projet. Votre compte est associé à 2 projets :

- un projet commun pour tous (ZZ3)
- un projet propre à chacun (cf. login)

Utilisez votre propre projet pour votre TP.

Étape 1. Déployez une instance à partir de l'image CirrOS (distribution linux minimal pour les tests clouds).
Déterminez pour votre instance : les ressources utilisées, son IP, le login et le password par défaut (cf. logs).
Pouvez vous accéder à la VM (ping) ?

Étape 2. Associez une IP flottante à votre instance.
Pouvez vous accéder à l'instance avec un ping ? Par SSH ?
Déterminez les rgèles de filtrage IP appliquées à la VM

Étape 3. Créez vous un groupe de sécurité autorisant uniquement le ping et le ssh en entrée et autorisant tout le trafic en sortie. Associez votre groupe de sécurité à votre instance.
Pouvez vous accéder à l'instance avec un ping ? Par SSH ?
Terminez votre instance.

Étape 4. Créez vous une paire de clé ssh (`ssh-keygen -t rsa -f tpcloud`) et chargez la clé publique dans le dashboard (Compute / Accès et Sécurité).
Créez une instance Ubuntu en lui associant votre clé SSH.
Vérifiez que vous pouvez vous connecter par SSH sur votre instance.

Étape 5. Installez sur votre instance le repository openstack (package centos-release-openstack-queens) puis le client openstack (package python-openstackclient). Pensez à définir au préalable le proxy dans votre environnement root :
`export http_proxy=http://proxy.dsi.uca.fr:8080`
Depuis un compte utilisateur non root, créez le script openstackrc (cf. script Openstack RC File pour l'API v3 fournit par le dashboard dans `API Access`), et vérifiez que votre client est fonctionnel avec la commande `openstack server list`.
Cette instance sera votre **console** d'admin openstack pour la suite du TP.

Étape 6. Testez les clients openstack pour créer une instance, la terminer et pour transférer un objet sur un conteneur d'objet de votre projet.
Vous aurez besoin pour cela du client swift (stockage objet) : package `python-swiftclient` sous Debian, Ubuntu ou CentOS.

1. `https://releases.openstack.org/queens/index.html`
2. `https://www.rdoproject.org`

Préparation des instances

Nous allons reprendre l'exemple de raytracing vu dans le TP précédent pour le déployer sur un cloud Openstack.

Étape 7. Ecrivez le script User Data qui permette de déployer une instance PovRay. Le script sera exécuté automatiquement à l'initialisation de l'instance et doit :

- installer et configurer les clients openstack avec votre authentification
- récupèrer l'archive zzpovray.tar.gz depuis le stockage objet du projet commun ZZ3 (avec wget, curl ou client swift)
- genèrer un lot d'images avec PovRay
- transfèrer ensuite les images générées sur le stockage objet

L'instance peut être terminée à la fin de l'exécution du script User Data.

Étape 8. Ecrivez le script User Data qui permette de déployer une instance de post-traitement qui doit :

- installer et configurer les clients openstack avec votre authentification
- installer ImageMagick (pour obtenir la commande `convert`
- récupèrer les images générées par PovRay
- générer le GIF aggrégeant toutes les images
- transférer le résultat sur le stockage en mode objet

L'instance peut être terminée à la fin de l'exécution du script User Data.

Déploiement et orchestration

On souhaite automatiser l'exécution de PovRay sur plusieurs instances et le post-traitement.

Étape 9. Pour illustrer l'orchestration avec Heat, définissez une *stack* qui déploie toutes les instances PovRay nécessaire à la génération de toutes les images.

Étape 10. Ecrivez un code qui prend en argument le nombre d'images par lot et qui automatise le déploiement et l'exécution de notre chaîne de traitement PovRay sur le cloud Openstack.
Ce code peut être écrit en bash avec les clients en ligne de commande ou en Java avec JClouds [3] (attention le support de Heat n'est pas encore implémenté dans JClouds).

> Envoyez vos codes (1 fichier PDF) sur la plateforme Cours en Ligne de l'ENT (vous pouvez par exemple utiliser les commandes a2ps et ps2pdf pour générer le PDF).

3. cf. `http://developer.openstack.org/` et `https://jclouds.apache.org/`