

TP: Déploiement, usage et outillage d'Openstack

Table of Contents

Partie 1 : Installation et découverte d'Openstack	1
Comment s'appelle l'image disponible par défaut ?	1
A quoi correspond la notion de projet dans Openstack ?	2
Décrire les actions que vous avez dû réaliser pour accéder à la VM.	2
A quoi sert une IP flottante ?	2
Partie 2: Création d'un réseau	3
A quel réseau est connecté le routeur router1 ?	3
Création d'un réseau	3
Ajout d'une interface au routeur	3
Partie 3: Utilisation de la CLI openstack	3
Liste des images disponibles	4
Création de la VM de la partie 1	4
Création de la VM de la partie 2	4
Partie 4 : Déployer une application Wordpress	5
Quel est l'utilisateur par défaut de l'image ubuntu ?	5
Import de l'image Ubuntu	6
Création du réseau	6
Création des groupes de sécurité	6
Allocation des IP flottantes	7
Création des instances	7
Configuration de l'instance bdd	7
Configuration de l'instance web	8
Partie 5 : Utilisation de Terraform avec Openstack	8
Commandes Terraform	8

Partie 1 : Installation et découverte d'Openstack

Comment s'appelle l'image disponible par défaut ?

Une image Cirros [<https://github.com/cirros-dev/cirros>]

A quoi correspond la notion de projet dans Openstack ?

Un projet est un espace de travail isolé permettant de regrouper des ressources (machines virtuelles, réseaux, volumes, etc.) et des utilisateurs. Il permet notamment de définir des quotas et des politiques d'accès.

Décrire les actions que vous avez dû réaliser pour accéder à la VM.

IP flottante

On va lui allouer une IP flottante pour pouvoir s'y connecter en SSH.

1. Accéder à Horizon
2. Aller dans **Projet** › **Compute** › **Instances** › **Actions** › **Associate Floating IP**
3. On choisit l'IP flottante et la VM à associer

Groupes de sécurité



Une VM Openstack appartient à un groupe de sécurité qui définit les règles d'accès réseau (i.e. firewall). On va donc devoir ajouter une règle pour autoriser le trafic SSH et ICMP.

1. Aller dans **Projet** › **Compute** › **Network** › **Security Groups** › **Default** › **Manage Rules** › **Add Rule**
 - a. Rule: SSH (== TCP, port 22)
 - b. Remote: CIDR
 - c. CIDR: 0.0.0.0/0 (tout le monde)
 - d. Rule: All ICMP (tout le protocole ICMP)
 - e. Remote: CIDR
 - f. CIDR: 0.0.0.0/0
2. On peut maintenant ping et se connecter en SSH à la VM.

A quoi sert une IP flottante ?

Une IP flottante est une adresse IP publique qui peut être associée à une machine virtuelle. Sa nature flottante permet de la détacher d'une machine virtuelle pour la rattacher à une autre pour de la haute disponibilité par exemple.

Partie 2: Création d'un réseau

A quel réseau est connecté le routeur router1 ?

Il est connecté au réseau public, ce qui équivaut à une connexion à l'extérieur du cloud. Ainsi qu'au réseau private qui est un réseau privé interne au cloud. Le routeur agit comme passerelle vers l'extérieur.

Création d'un réseau

1. Aller dans **Projet** > **Network** > **Networks** > **Create Network**

- a. **Network**

- i. Name: mynetwork

☒ Enable Admin State (activer le réseau administrativement)

☒ Create Subnet (créer un sous-réseau)

- b. **Subnet**

- i. Subnet Name: mysubnet

ii. Network Address source: ENTER NETWORK ADDRESS MANUALLY

iii. Network Address: 10.2.0.0/24

iv. Gateway IP: 10.2.0.254

- c. **Subnet Details**

☒ Enable DHCP

- i. DNS Name Servers: 8.8.8.8

Ajout d'une interface au routeur

1. Aller dans **Projet** > **Network** > **Routers** > **router1** > **Interfaces** > **Add Interface**

- a. Subnet: MYNETWORK: 10.2.0.0/24 (MYSUBNET)

On peut voir dans la topology que le routeur est maintenant connecté au réseau mynetwork.

Pour créer une instance, on associe une IP flottante comme précédemment et on la connecte au réseau mynetwork.

Partie 3: Utilisation de la CLI openstack

On récupère les informations de connexion à Openstack dans le **Projet** > **Compute** > **Access & Security** > **API Access** > **Download OpenStack RC File**.

On source le fichier pour définir les variables d'environnement nécessaires à l'utilisation de la CLI

Openstack.

```
. demo-openrc.sh
# Entrer le mot de passe
```

Liste des images disponibles

```
openstack image list
```

ID	Name	Status
2bac8bad-4f7a-4ce7-bfa3-ae17ab511b21	cirros-0.6.2-x86_64-disk	active

Création de la VM de la partie 1



On considère que la clé SSH a déjà été ajoutée dans Horizon sous le nom main. Ainsi que le pool de floating IP est nommé public.

```
# On alloue une IP flottante
openstack floating ip create public
openstack server create --flavor m1.nano \
  # On utilise l'image Cirros
  --image cirros-0.6.2-x86_64-disk \
  # On la connecte au réseau private
  --network private \
  # On lui assigne au groupe de sécurité par défaut
  --security-group default \
  # On associe la clé SSH main
  --key-name main \
  myinstance
# On associe l'IP flottante à l'instance
openstack server add floating ip myinstance $FLOATING_IP
# On supprime l'instance
openstack server delete myinstance
# On supprime l'IP flottante, la rendant disponible pour une autre utilisation dans le pool
openstack floating ip delete $FLOATING_IP
```

Création de la VM de la partie 2

Réseau et sous-réseau

```
# On crée le réseau de la partie 2
openstack network create mynetwork
# On crée le sous-réseau
openstack subnet create \
  # Dans le réseau mynetwork
  --network mynetwork \
  # Avec le range d'IP
  --subnet-range 10.2.0.0/24 \
  # On fixe la passerelle (normalement la première IP du range)
  --gateway 10.2.0.254 \
  # On fixe le serveur DNS
  --dns-nameserver 8.8.8.8 \
  mysubnet
# On crée une interface liant le routeur au réseau
openstack router add subnet router1 mysubnet
```

Instance

```
# On alloue une IP flottante
openstack floating ip create public
openstack server create \
  # Une flavor est un template de ressources (CPU, RAM, etc.)
  --flavor m1.nano \
  --image cirros-0.6.2-x86_64-disk \
  --network mynetwork \
  --security-group default \
  --key-name main \
  myinstance2
# On associe l'IP flottante à l'instance
openstack server floating ip set myinstance2 $FLOATING_IP
# On supprime l'instance
openstack server delete myinstance2
# On supprime l'IP flottante
openstack floating ip delete $FLOATING_IP
```

Partie 4 : Déployer une application Wordpress

Quel est l'utilisateur par défaut de l'image ubuntu ?

ubuntu, mais on peut aussi utiliser la solution de provisionnement cloud-init pour créer un utilisateur.

Import de l'image Ubuntu

```
wget https://cloud-images.ubuntu.com/releases/22.04/release/ubuntu-22.04-server-cloudimg-amd64.img
openstack image create \
  # L'image est une image disque au format QEMU (qcow2)
  --disk-format qcow2 \
  # Ce n'est pas une image pour un cloud public (AWS, Azure, etc.), le format est donc direct (bare)
  --container-format bare \
  # On ne veut pas que l'image soit partagée avec d'autres projets
  --private \
  --file ubuntu-22.04-server-cloudimg-amd64.img \
  ubuntu-22.04
```

Création du réseau

```
# On crée le réseau et le sous-réseau
openstack network create mynetwork
openstack subnet create --network mynetwork \
  --subnet-range 10.2.0.0/24 \
  --gateway 10.2.0.254 \
  --dns-nameserver 8.8.8.8 \
  mysubnet
# On lie le routeur au réseau
openstack router add subnet router1 mysubnet
```

Création des groupes de sécurité

```
# On crée les groupes de sécurité
openstack security group create --description "Allow SSH" ssh
# S'assurer que les règles par défaut sont bien présentes
# egress: sortant, ingress: entrant
openstack security group rule create --egress --protocol tcp ssh
openstack security group rule create --egress --protocol icmp ssh
openstack security group rule create --egress --protocol udp ssh
# Autoriser le trafic SSH
openstack security group rule create --ingress --protocol tcp --dst-port 22 ssh
```

```
# On crée un groupe pour le trafic web
openstack security group create --description "Allow HTTP" web
openstack security group rule create --ingress --protocol tcp --dst-port 80 web
openstack security group rule create --ingress --protocol tcp --dst-port 443 web
```

```
# On crée un groupe pour le trafic SQL
openstack security group create --description "Allow MariaDB" bdd
openstack security group rule create --ingress --remote-group web --protocol tcp --dst
-port 3306 bdd
# Autoriser MYSQL X Protocol
openstack security group rule create --ingress --remote-group web --protocol tcp --dst
-port 33060 bdd
```

Allocation des IP flottantes

```
# On alloue une IP flottante, utilisant jq (un outil de manipulation de JSON) pour
récupérer l'adresse IP flottante
IP1=$(openstack floating ip create public -f json | jq -r '.floating_ip_address')
IP2=$(openstack floating ip create public -f json | jq -r '.floating_ip_address')
```

Création des instances

```
# On crée les instances
openstack server create --flavor ds1G \
  --image ubuntu-22.04 \
  --network mynetwork \
  --security-group ssh \
  --security-group web \
  --key-name main \
  web
# On attend que l'instance soit provisionnée
sleep 10
# Associer l'IP flottante à l'instance
openstack server add floating ip web $IP1
```

```
openstack server create --flavor ds1G \
  --image ubuntu-22.04 \
  --network mynetwork \
  --security-group ssh \
  --security-group bdd \
  --key-name main \
  bdd
sleep 10
openstack server add floating ip bdd $IP2
```

Configuration de l'instance bdd

```
# On se connecte à l'instance bdd
ssh ubuntu@$IP2 -i ~/.ssh/id_ed25519
# On récupère le script de configuration dans le dépôt
git clone https://github.com/signed-log/ostack-tp.git
cd ostack-tp/scripts
# Script de configuration de la base de données dans scripts/
sudo ./db.sh
```

Configuration de l'instance web

```
# On se connecte à l'instance web
ssh ubuntu@$IP1 -i ~/.ssh/id_ed25519
git clone https://github.com/signed-log/ostack-tp.git
cd ostack-tp/scripts
sudo ./web.sh
```

Partie 5 : Utilisation de Terraform avec Openstack

Commandes Terraform

```
# On initialise Terraform dans un répertoire avec un fichier main.tf
terraform init
# On vérifie le plan que Terraform va appliquer
terraform plan
# On applique le plan
terraform apply
# On détruit les ressources
terraform destroy
```