Infrastructures distribuées pour le passage à l'échelle

TP4 *Infrastructure as Code*
(avec HashiCorp Terraform)

Rappel

L'environnement Openstack à votre disposition est un cloud privé *all in one* (tous les services sur une seule machine).

- 40 coeurs logiques, 240 Go de RAM, 2 To de disque
- IP : 10.20.22.18 (uniquement depuis `zzcluster`)
- DNS : zzcloud.oscarlocal (uniquement depuis `zzcluster`)

Eléments fournis

Le conteneur `zz3-tp4` du projet commun `ZZ3` contient :

- **hello-api** : un binaire qui lance une API de test sur le port 8080 avec 2 opérations :
  `GET /`
  `POST /hello/<name>`
- **hello-api.service** : fichier de configuration du service avec systemd
- **user-data-haproxy.yml** : script cloud-config qui installe haproxy et complète la configuration du *load balancer*

Ce conteneur est public, et accessible avec curl directement sans authentification :

```
$ curl -k -s https://10.20.22.18:8080/v1/AUTH_4b315635c6a0412198f0638b8edccec9/zz3-tp4/hello-api -o hello-api
```

Sur `zzcluster`, sont installés :

- Terraform 1.6.6 (`/usr/local/bin/terraform`),
  sous Business Source License, maintenu par HashiCorp
- OpenTofu 1.6.2 (`/usr/bin/tofu`) ;
  sous licence libre (Mozilla Public Licence), maintenu par la communauté sous l'égide de la fondation Linux.

Les deux outils sont similaires et largement compatibles.

1 Déploiement de l'API Hello

Étape 1. Sur `zzcluster`, initialisez un projet Terraform **hello**, qui :

1. déploie une instance ubuntu,
2. lui associe une IP flottante,
3. configure ses groupes de sécurité pour autoriser l'accès ssh et le port 8080.

Vérifiez votre déploiement Terraform.

Étape 2. Modifiez votre code Terraform pour ajouter un script user-data à votre instance qui :

- copie le binaire de l'API Hello dans le répertoire `/usr/local/bin/`
- ajoute le fichier de service `/etc/systemd/system/hello-api.service`
- recharge la config systemd avec la commande : `systemctl daemon-reload`
- démarre le service `hello-api`, et active le service au boot
  (`systemctl enable hello-api.service`)

Testez l'accès à l'API Hello avec la commande curl.

2 Passage à l'échelle de l'API Hello

Étape 3. Créez un nouveau projet Terraform qui :
- déploie plusieurs instances de l'api `hello-api`
- déploie un load balancer en adaptant le script user data fourni (`user-data-haproxy.yml`)

Testez l'accès à l'API Hello avec la commande curl.
Testez le changement du nombre d'instance avec Terraform.

3 Déploiement et orchestration

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

Étape 4. Ecrivez dans le langage de votre choix une API web qui :
- permet de générer une image avec povray (en précisant son id) et copie l'image sur le stockage objet,
- permet de récupérer une image générée (ou renvoie une erreur 404 si l'image n'a pas été générée),
- permet de générer la vidéo au format GIF si toutes les images ont été générée, de la copier sur le stockage objet,
- permet de récupérer la vidéo si elle a été générée (erreur 404 sinon)

Veillez bien a utiliser des *Application Credentials* spécifique à votre API pour l'accès au stockage objet, et non vos identifiants personnels.

Étape 5. Ecrivez le code Terraform qui permet de déployer votre API sur une infrastructure qui passe à l'échelle (avec 1 load-balancer, et N workers).

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