

[Terraform] Déploiement d'une instance

Introduction

Nous allons voir comment déployer une instance sur le cloud Infomaniak en utilisant Terraform via le provider OpenStack.

Source

- [Tutoriel Xavki](#)

Déploiement

Tout d'abord, il faudra récupérer le fichier **openrc** depuis l'interface web.

Pour cela, on peut se rendre dans la liste des projets et cliquer sur les trois points puis sur **Gérer les utilisateurs** :

The screenshot shows the Neopipe dashboard. At the top, there's a navigation bar with 'Neopipe' and 'Projets' tabs, and a 'Créer un projet' button. Below the navigation bar, there's a section for 'Neopipe' with a 'Gérer' link. A prominent banner offers '300,00 € offerts pour découvrir notre Public Cloud !' with a progress bar showing '300,00 € restant sur 300,00 €'. Below this, a 'Niveau de ressources' section indicates 'vous disposez du niveau 1' and provides a link to 'Comparer tous les quotas OpenStack (EN)'. The main area displays a table of projects. The first project is 'neopipe-public-cloud-01' with a cost of '0,08 €' and status 'Actif'. A dropdown menu for this project shows options: 'Renommer le projet', 'Gérer les utilisateurs', 'Ouvrir le dashboard OpenStack', and 'Supprimer le projet'. A search bar and a '1 projet' indicator are also visible.

Projet	Coûts actuels	Statut	Utilisateurs OpenStack	Niveau de ressources
neopipe-public-cloud-01 PCP-A3G7TLJ	0,08 €	Actif	1 Gérer	Niveau 1

Je vous recommande de renommer ce fichier en **openrc.sh** pour des soucis de simplicité.

Ensuite je vous invite à le modifier pour ajouter le mot de passe de votre utilisateur OpenStack dedans et commenter la ligne qui demande votre mot de passe (sinon vous devrez le saisir à chaque fois :

```
#[ -z "$OS_PASSWORD" ] && read -e -p "Please enter your OpenStack Password for project $OS_PROJECT_NAME as user $OS_USERNAME: " OS_PASSWORD
```

On peut maintenant sourcer ce fichier pour prendre en compte les nouvelles variables d'environnement :

```
source openrc.sh
```

On peut maintenant créer la paire de clé SSH qui nous servira à nous connecter à notre instance :

```
ssh-keygen
```

On peut maintenant passer à l'édition du fichier **main.tf** qui sert à décrire la configuration de notre instance :

```
# Define required providers
terraform {
  required_providers {
    openstack = {
      source = "terraform-provider-openstack/openstack"
      version = "1.44.0"
```

```
}  
}  
}
```

```
# Configure the OpenStack Provider
```

```
provider "openstack" {  
}
```

```
resource "openstack_compute_keypair_v2" "ssh-alpine-02" {
```

```
  name = "ssh-alpine-02"
```

```
  public_key = "ssh-rsa
```

```
AAAAB3NzaC1yc2EAAAADAQABAAQGDix1Bbfu8YjHpp0AS60EwpQ4oZWjo+YeT3FyQlby324HIJsfS0h3DDIgP4G  
OcuybijApxthiWAzJ9xgPnSAD6yc7wzrInE0/2SeQ/CkFjQLowgkuRv4DBg3t+A8tai9Yv/zYXmCZdmG7ydVfHXCWSW  
n2k/m18SJZju0CWRojFe4t8M3cOXMERq0FHuYUhRgCEki5tImQCGFd/8HqCld3Zv47+ocldwo2p2mEi8+FAYeAgb6J2  
DACyHpg68NUa8xQhf8tPghIJ7XME8p81JhE0yanKMr4RG4s9vT63fnwqCrQclHPWsyeFaBrXJYX1kNUs5aoZY71cCYYj  
5nIVEMxQfstkbpzwwDosaRTXiYgCyKP31k+Kx37J6XhZfLpQtq/z+b62BvnBi+Z+3T/sOPLjDoq5GRIHO3aYJjTxOv5Nk  
wDxjLVVWISDRjw2TP5EZgvtC/eHZHsCFaxt9e1/ROOiCTcD63Vao3ZgCb7jl9FaTNe+MW2yrXFpoL9yhSbtX0U=  
elie@work-l-elie"
```

```
}
```

```
# Create a security group
```

```
resource "openstack_compute_secgroup_v2" "bookstack-test" {
```

```
  name      = "bookstack-test"
```

```
  description = "Security Group Description"
```

```
  rule {
```

```
    from_port = 22
```

```
    to_port   = 22
```

```
    ip_protocol = "tcp"
```

```
    cidr       = "0.0.0.0/0"
```

```
  }
```

```
  rule {
```

```
    from_port = 6875
```

```
    to_port   = 6875
```

```
    ip_protocol = "tcp"
```

```
    cidr       = "0.0.0.0/0"
```

```
  }
```

```
rule {
  from_port = 443
  to_port   = 443
  ip_protocol = "tcp"
  cidr      = "0.0.0.0/0"
}

# Create a web server
resource "openstack_compute_instance_v2" "bookstack" {
  name          = "bookstack-test"
  image_id      = "ec665219-cc8c-4df0-aa45-a886c6ea9701"
  flavor_name   = "a1-ram2-disk20-perf1"
  key_pair      = "ssh-alpine-02"
  security_groups = ["bookstack-test"]

  metadata = {
    application = "bookstack"
  }

  network {
    name = "ext-net1"
  }
}
```

Vous pouvez modifier les champs dans ce fichier pour l'adapter à vos besoins.

Vous avez simplement besoin de savoir que nous allons déployer 3 ressources :

- Une paire de clé SSH
- Un groupe de sécurité (qui permet de gérer les règles de pare-feu de notre instance).
- Notre instance

À noter que vous pouvez retrouver les IDs d'images et de flavors depuis l'interface graphique OpenStack en simulant la création d'une instance.

On peut désormais lancer notre instance :

```
terraform init
```

```
terraform plan
```

```
terraform apply
```

Une confirmation sera demandée, tapez **yes**.

Pour récupérer l'adresse IP de notre instance il nous faut d'abord lister les ressources créés :

```
terraform state list
```

Ici on obtient quelque chose de la sorte :

```
openstack_compute_instance_v2.bookstack
openstack_compute_keypair_v2.ssh-alpine-02
openstack_compute_secgroup_v2.bookstack-test
```

Ce qui nous intéresse, ce sont les informations de l'instance donc on peut taper cette commande :

```
terraform state show openstack_compute_instance_v2.bookstack
```

Vous devriez voir apparaître l'adresse IPv4 ce qui vous permettra de vous connecter grâce à votre clé SSH sur l'instance.

Modification de l'instance

Vous pouvez modifier la configuration de votre instance en éditant le fichier **main.tf** puis en faisant :

```
terraform plan
```

Vérifiez les configurations modifiées.

Puis vous pouvez appliquer :

```
terraform apply
```

Suppression de l'instance

Voici la commande qui vous permettra de supprimer votre instance :

```
terraform destroy
```

Une confirmation sera demandée, tapez **yes**.

Si un plantage arrive, relancez la commande, cela devrait fonctionner.

Revision #3

Created 11 March 2024 09:57:42 by Elieroc

Updated 12 March 2024 11:59:35 by Elieroc