Terraform

Infrastructure As Code

- [Terraform] Installation
- [Terraform] Déploiement d'une instance

[Terraform] Installation

Introduction

Débutez l'**Infrastructure As Code avec Terraform** pour déployer de manière automatiser des instances ou des réseaux entiers dans le cloud.



Installation

Debian

```
sudo apt-get update && sudo apt-get install -y gnupg software-properties-common && wget -O- https://apt.releases.hashicorp.com/gpg | gpg --dearmor | sudo tee /usr/share/keyrings/hashicorp-archive-keyring.gpg > /dev/null && gpg --no-default-keyring --keyring /usr/share/keyrings/hashicorp-archive-keyring.gpg --fingerprint && echo "deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] https://apt.releases.hashicorp.com $(lsb_release -cs) main" | sudo tee /etc/apt/sources.list.d/hashicorp.list && sudo apt update && sudo apt-get install terraform
```

[Terraform] Déploiement d'une instance

Introduction

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

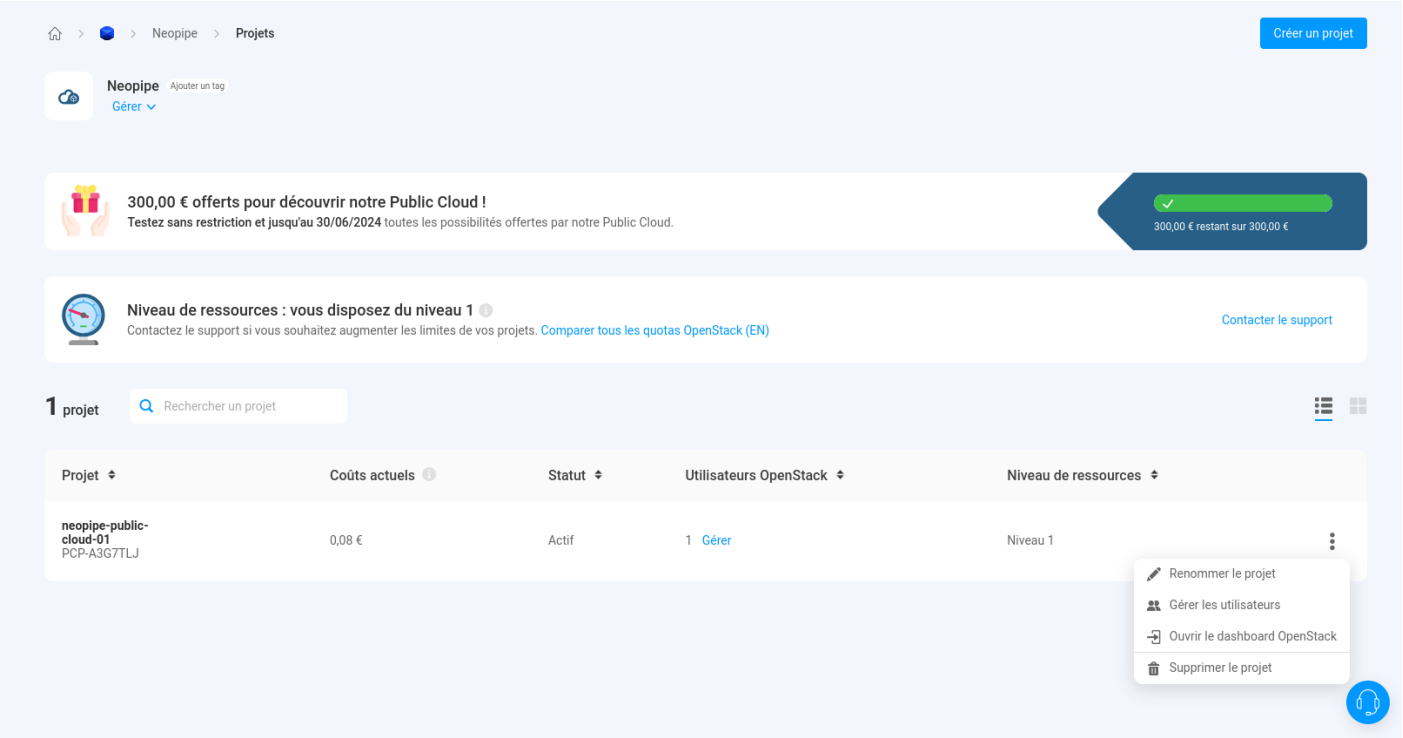
Source

- 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** :



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
AAAAB3NzaC1yc2EAAAADAQABAAABgQDiX1Bbfu8YjHpp0AS60EwpQ4oZWjo+YeT3FyQIby324HIJsfS0h3DDlgP4GOcuyb
iJApxthiWAzJ9xgPnSAD6yc7wzrInE0/2SeQ/CkFjQLOWgkuRv4DBg3t+A8tai9Yv/zYXmCZdmG7ydVfHXCWSWn2k/m18S
JZju0CWRoJFe4t8M3cOXMErq0FHuYUhRgCEki5tImQCGFd/8HqCld3Zv47+ocIdwo2p2mEi8+FAYeAgb6J2DACyHpg68NU
a8xQhf8tPghlJ7XME8p81JhE0yanKMr4RG4s9vT63fnwqCrQclHPWsyeFaBrXJYX1kNUs5aoZY71cCYYj5nIVEMxQfstkb
pzwwDosaRTXiYgCyKP31k+Kx37J6XhZfLpQtq/z+b62BvnBi+Z+3T/sOPLjDoq5GRlHO3aYJjTxOv5NkwDxjLVVWISDRjw
2TP5EZgvwTc/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 dé 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 devirez 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.