# Configuration de l'Environnement OpenStack et des Améliorations du Code Terraform

## 1. Préparation de l'Environnement de Base

- **Installation du Système :** Le système Ubuntu 22.04.4 LTS Desktop a été installé en tant que machine virtuelle dans VMware Workstation Pro sur un hôte Windows. La machine virtuelle s'est vu attribuer 16 Go de RAM, 4 vCPU et 80 Go d'espace disque.
- **Configuration SSH sous Ubuntu :** Sur le système Ubuntu, le script `setup_ssh_2222.sh` a été exécuté, installant le serveur OpenSSH, le configurant pour écouter sur le port 2222 et définissant les règles de pare-feu UFW appropriées pour autoriser les connexions entrantes sur ce port.
- **Installation de DevStack :** Après connexion à la machine virtuelle Ubuntu depuis l'hôte Windows via SSH (sur le port 2222), le script `install_devstack_en.sh` a été lancé. Ce script a cloné le dépôt DevStack, créé un fichier de configuration de base `local.conf` et initialisé l'installation d'OpenStack à l'aide de `./stack.sh`. L'installation s'est terminée avec succès et le pare-feu UFW a été configuré pour permettre l'accès aux services OpenStack.

## 2. Configuration Initiale de Terraform

- **Création du Projet Terraform :** Un répertoire de travail pour le projet Terraform a été créé sur le système Ubuntu.
- **Fichiers de Configuration :** Les fichiers suivants ont été placés dans le répertoire :
  - `main.tf` : Version initiale du fichier contenant la configuration du fournisseur OpenStack et la logique de création dynamique de machines virtuelles.
  - `vm.csv` : Fichier contenant les données d'entrée pour les machines (noms, images, saveurs (flavors), noms de paires de clés SSH, noms de groupes de sécurité et noms de réseaux).

## 3. Identification et Résolution des Problèmes

- **Problèmes de Copie de Code :** Des problèmes ont été rencontrés lors de la copie de code depuis un éditeur de texte (par ex. le Bloc-notes Windows) et de son collage dans l'éditeur `nano` sous Ubuntu. Cela provoquait l'apparition de "caractères invisibles" (probablement liés aux différences d'encodage de fin de ligne entre Windows CRLF et Unix LF). Ce problème a été résolu en utilisant un outil en ligne de commande sous Linux tel que `dos2unix`, qui convertit le formatage des fichiers texte. La commande `sudo apt install dos2unix` installe cet outil, et `dos2unix nom_du_fichier.tf` corrige le fichier.

  **Identification des Faiblesses du Code :**

  Plusieurs faiblesses ont été identifiées dans la version initiale du code `main.tf`, notamment :

- Absence de validation des données d'entrée du fichier CSV.
- Nécessité de créer/importer manuellement les paires de clés SSH dans OpenStack avant d'exécuter Terraform. *(Corrigé dans la version finale)*
- Absence de vérification de l'existence des images, saveurs, réseaux et groupes de sécurité avant de tenter de les utiliser. *(Corrigé dans la version finale)*
- Stockage des informations d'identification directement dans le code. *(Corrigé dans la version finale)*
- Problèmes potentiels avec des noms d'instances dupliqués dans le CSV.
- Absence de configuration du cycle de vie des ressources (`lifecycle`).
- Absence de dépendances explicites (`depends_on`). *(Corrigé dans la version finale)*

- Manque de commentaires suffisants dans le code. *(Corrigé dans la version finale)*

**4. Amélioration du Code Terraform (Version Finale)**

En réponse aux problèmes identifiés, le code Terraform a été considérablement amélioré, aboutissant à la structure et à la logique finales :

**Séparation des Données Sensible**

- Les informations d'identification et autres variables de configuration ont été déplacées hors de main.tf.
- Un fichier variables.tf a été créé pour déclarer toutes les variables (y compris les informations d'accès, les chemins, les noms par défaut) avec des descriptions et, le cas échéant, des valeurs par défaut.
- Un fichier secrets.tfvars a été créé pour stocker les valeurs réelles des données sensibles (mot de passe, URL, utilisateur). Ce fichier est destiné à un usage local.
- Un fichier .gitignore a été créé pour empêcher l'envoi accidentel du fichier secrets.tfvars et d'autres fichiers d'état Terraform vers le dépôt Git.

**Gestion des Paires de Clés SSH :**

- Le code **ne génère pas** automatiquement de nouvelles paires de clés
- Au lieu de cela, basé sur les noms dans la colonne key_pair du fichier vm.csv, le code **lit le contenu des fichiers de clés publiques locaux existants** (par ex. ~/.ssh/nom_cle.pub) à l'aide de la fonction file().
- Il **importe ensuite ces clés publiques** dans OpenStack en tant que ressources openstack_compute_keypair_v2. Cela élimine le besoin d'importation manuelle mais nécessite toujours que la paire de clés (publique et privée) existe localement *avant* d'exécuter Terraform.

**Validation des Données d'Entrée et des Ressources :**

- Une ressource null_resource avec un bloc lifecycle.precondition a été ajoutée pour vérifier que le fichier vm.csv chargé n'est pas vide avant de tenter de créer des ressources.
- Des sources de données (data) openstack_networking_network_v2 et openstack_networking_secgroup_v2 avec une boucle for_each (définies *en dehors* de la ressource d'instance) ont été utilisées pour **vérifier l'existence** des réseaux et des groupes de sécurité spécifiés dans le CSV *avant* que les instances ne tentent de les utiliser.
- La recherche des groupes de sécurité (data "openstack_networking_secgroup_v2") a été affinée en ajoutant tenant_id pour éviter les erreurs en cas d'existence de groupes portant le même nom dans différents projets (résolution du problème "More than one Security Group found").
- **Gestion des Conflits de Noms de Clés SSH :** L'idée initiale était d'implémenter une validation proactive pour détecter les conflits de noms de clés SSH avant terraform apply. Cela aurait nécessité l'utilisation d'une source de données comme openstack_compute_keypairs_v2 pour lister les clés existantes. Cependant, cette source de données n'était pas disponible dans la version du fournisseur OpenStack utilisée (~> 1.50). La mise à jour vers une version plus récente du fournisseur (par ex., ~> 2.0) a été envisagée, mais abandonnée en raison des risques potentiels de changements incompatibles ("breaking changes") qu'une mise à jour majeure pourrait introduire. Par conséquent, il a été décidé de **ne pas implémenter la validation proactive**. **L'exigence est donc que les noms de clés SSH spécifiés dans la colonne key_pair du fichier vm.csv soient uniques au sein du projet OpenStack cible.** Le code s'appuie sur l'erreur renvoyée par l'API OpenStack lors de terraform apply si un conflit de nom est détecté. Un commentaire détaillé dans main.tf explique comment interpréter et résoudre cette erreur si elle survient.

**Amélioration de la Lisibilité et de la Fiabilité :**

- Un fichier outputs.tf a été ajouté, définissant des sorties (par ex. created_vm_details) qui affichent des informations utiles sur les ressources créées après l'exécution de terraform apply.
- De nombreux commentaires ont été ajoutés dans les fichiers .tf pour expliquer le fonctionnement des différents blocs de code.
- Des dépendances explicites (depends_on) ont été ajoutées dans la ressource openstack_compute_instance_v2 pour garantir l'ordre correct de création des ressources (par ex. dépendance de la validation CSV, de la tentative de création/importation des paires de clés et de la recherche des réseaux/groupes de sécurité).

**Structure des Fichiers :** La structure finale du répertoire du projet Terraform comprend les fichiers : main.tf, variables.tf, outputs.tf, secrets.tfvars, vm.csv, .gitignore.

**5. Exécution des Opérations Terraform**

- Toutes les opérations Terraform (init, plan, apply) ont été exécutées depuis le système Ubuntu.
- Lors de l'exécution de plan et apply, l'indicateur -var-file="secrets.tfvars" a été utilisé pour garantir le chargement des valeurs des variables depuis le fichier dédié et éviter la demande interactive des informations d'identification.

Le résultat de ces actions est un code Terraform plus fiable, sécurisé et facile à gérer pour l'automatisation du déploiement de machines virtuelles dans un environnement OpenStack/DevStack.

05/05/2025 Sannois

Rafal RUTKOWSKI