Diary :

17/01/2025

Je vais continuer de resoudre les pbs avec chatgpt en lui disant a chaque fois quelle solution a fonctionné pour qu'il me liste bien a la fin les difficultés rencontrées pour le rapport et continuer de taffer.
Je supprimerai pour le rapport les mentions de chatgpt etc.

Path vers la cle ssh et adresses ip des vms update dans inventory par update_inventory.sh

Pour l'instant, run : `./setup_and_run.sh` pour lancer tout le programme, on pourrait rendre le debug plus facile en divisant en différents .sh les différentes commandes dans setup_and_run.sh pour debugb plus facilement mais ya pas le temps !

Reste a faire :

1. Faire en sorte que les adresses ip soient gérée dynamiquement ou statiquement mais qu'on ait pas besoin de les copier coller ou alors que les vms soient gérées via leurs noms :

   Il y a peut etre des solutions utilisant juste ansible ou d'autres utilisant juste le shell

2. Clé SSH ou autre solution pour que le setup_and_run.exe fonctionne sans sudo à compléter

3. Diviser les fichiers de commandes .sh en plusieurs sous-fichiers dans un directory (ex : commands) pour run des commandes séparéments en débugs et pas tout run du début à chaque fois.

   exemple :
   commands
   |
   --- setup_terraform.sh
   --- setup_venv.sh
   --- setup_ansible.sh
   --- run_all_commands.sh

PROF

Commandes actuelles pour lancer le projet :

1. Vérifier et lancer Terraform
   Initialiser Terraform (si ce n'est pas déjà fait) :

   ```
   terraform init
   ```

   Appliquer la configuration pour créer les VMs (master + workers) :

   ```
   terraform apply
   ```

2. Vérifier les ressources créées par Terraform
   Afficher les adresses IP du master et des workers (si configuré dans output) :

terraform output

Tester l'accès SSH à la VM master :

```
ssh -i ~/.ssh/id_rsa_terraform ubuntu@<ip_master>
```

3. Entrer en environnement virtuel avant la partie Ansible :

4. Configurer les VMs avec Ansible
   Tester la connectivité Ansible avec le fichier inventory :

   ```
   ansible -i inventory all -m ping
   ```

   Configurer les workers (installer Python, copier le script, etc.) :

   ```
   ansible-playbook -i inventory worker.yml
   ```

   Configurer le master et exécuter le script principal (distribuer les tâches aux workers) :

   ```
   ansible-playbook -i inventory master.yml
   ```

5. Résultats attendus

   Les VMs (master + workers) sont créées et accessibles via SSH.
   Les workers sont configurés pour exécuter le WordCount.
   Le master distribue les partitions du texte aux workers, collecte les résultats, et affiche les WordCounts.

Probleme et solutions :

terraform apply retourne : `failed to dial libvirt: permission denied`

Cause :

L'utilisateur n'a pas les permissions pour accéder au socket /var/run/libvirt/libvirt-sock.

PROF

Solution :

Ajoutez l'utilisateur au groupe libvirt :
`sudo usermod -aG libvirt $(whoami) && newgrp libvirt`

Vérifiez les permissions du socket :
`ls -l /var/run/libvirt/libvirt-sock`

Si besoin, corrigez-les :
`sudo chown root:libvirt /var/run/libvirt/libvirt-sock && sudo chmod 660 /var/run/libvirt/libvirt-sock`

Redémarrez le service libvirtd :
`sudo systemctl restart libvirtd`

Testez avec virsh :
`virsh list --all`

Après ces étapes, relancez terraform apply

14/01/2025

On a commencé le projet avec terraform/docker pour contourner le pb de kvm qui ne fonctionne pas sur mac, mais Boris nous a dit qu'il veut qu'on utilise kvm, même si c'est sur une seule machine.

A terme, on veut aussi une vm qui heberge un site pour lancer le wordcount et voir les résultats.

importer l'image ubuntu :
wget https://cloud-images.ubuntu.com/focal/current/focal-server-cloudimg-amd64.img

Les différents types de fichiers et leurs raison d'être :

main.tf : Définit la configuration principale pour Terraform, y compris les ressources à créer (VMs, volumes, etc.).
cloudinit.yaml : Configure automatiquement les VMs lors de leur démarrage initial (utilisateurs, logiciels, scripts).
terraform.tfvars : Fournit les valeurs spécifiques aux variables définies dans variables.tf.
variables.tf : Définit les variables utilisées dans main.tf pour rendre la configuration modulaire et réutilisable.
playbook.yaml : Automatiser les tâches de configuration et déploiement sur les VMs avec Ansible (installation de logiciels, déploiement de scripts). \

Virsh

Verifier avec terraform les vm crées :
`virsh list`
Voir leurs adresses ip :
`virsh domifaddr <nom_de_la_vm>`

PROF

Ansible

Commande pour lancer le playbook : `ansible-playbook -i inventory wordcount.yml`

Commande pour générer une clé SSH pour ne pas toujours avoir à rentrer le login/password
`ssh-keygen -t rsa -b 2048 -f ~/.ssh/id_rsa_terraform -N ""`
Il faut ensuite mettre le contenu de la clé publique générée dans cloudinit.yaml (ssh-authorized-keys) puis le path de la clé privée dans inventory.sh

Ce qui est tenté :

J'essaie de créer une machine avec terraform/kvm et libvirt comme provider et qu'elle soit configurée avec ansible pour faire un wordcount.

C'est bon, j'ai réussi à mettre une adresse ip à ma vm, je vais maintenant pouvoir tester ansible.

Ansible fonctionne ! J'ai pu faire un wordcount sur une vm.

J'ai commencé à faire un worker et plusieurs vms, les yaml pour configurer les workers et le master ne fonctionnent pas, j'ai l'impression qu'il y a un problème avec le `inventory`. Peut être un problème d'adresse IP ou de clé SSH commune.

J'ai essayé de changer le nom du projet local en "Projet_infra_3ASN" a voir si ca a quitté le git.

Difficultés rencontrées : cf chat-gpt

Pas d'adresse IP attribuée à la VM

```
Problème : Terraform ne pouvait pas récupérer une adresse IP pour la VM.
Causes : Réseau "default" inactif ou mal configuré ; absence de qemu-
guest-agent.
Solution : Activation du réseau "default" dans KVM et installation de
qemu-guest-agent via cloud-init.
```

Demande de login/mot de passe dans la VM

```
Problème : Connexion via virsh console demandait un login et un mot de
passe.
Causes : cloud-init non configuré pour ajouter un utilisateur ou une clé
SSH.
Solution : Configuration d’un fichier cloudinit.yaml pour créer un
utilisateur ubuntu et ajouter une clé SSH.
```

Absence de mkisofs

PROF

```
Problème : Terraform retournait une erreur indiquant que mkisofs n'était
pas installé.
Causes : Le binaire manquait sur le système.
Solution : Installation de genisoimage ou création d’un lien symbolique
vers /usr/bin/mkisofs.
```

Ansible ne trouvait pas l’inventaire

```
Problème : Ansible ne trouvait pas le fichier inventory.
Causes : Mauvais nom du fichier ou chemin incorrect.
Solution : Renommer le fichier en inventory et utiliser son chemin
absolu.
```

Problèmes d’authentification SSH avec Ansible

```
Problème : Erreur Permission denied (publickey,password) lors de la
connexion.
Causes : Clé SSH incorrecte ou absente dans inventory; clé publique non
configurée dans la VM.
Solution : Génération d’une clé SSH, ajout au cloudinit.yaml, et test
manuel de la connexion.
```

Avertissement Ansible : clé SSH non approuvée

```
Problème : Ansible demandait de confirmer les clés SSH des hôtes.
Solution : Pré-approuver manuellement les clés SSH ou désactiver la
vérification avec host_key_checking = False.
```

Déploiement et exécution du WordCount

```
Problème : La configuration manuelle des dépendances et scripts était
fastidieuse.
Solution : Automatisation via Ansible avec un fichier wordcount.yml pour
installer Python, copier les fichiers, et exécuter le script.
```