Introduction

Ce projet consiste à déployer WordPress et MySQL en utilisant un volume persistant
L'environnement d'execution est un WSL UBUNTU windows sur une section Mobaxtrem

Provisionnent des ressources avec Terraform

J'ai installé d'abord terraform

```
sudo apt update && sudo apt upgrade -y
sudo apt install -y gnupg software-properties-common curl

curl -fsSL https://apt.releases.hashicorp.com/gpg | sudo gpg --dearmor -o
/usr/share/keyrings/hashicorp-archive-keyring.gpg

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 install terraform -y
terraform -v
```

Sur le portail GCP j'ai créé un projet "Cluster Kubernetes WordPress"

J'ai créé un service account sur Cloud Shell de GCP

```
gcloud iam service-accounts create terraform --display-name="Terraform admin account"
```

```
amineamine7897@cloudshell:~ (cluster-kubernetes-wordpress)$ gcloud iam service-accounts create terrafo
rm --display-name="Terraform admin account"
Created service account [terraform].
```

```
gcloud projects add-iam-policy-binding cluster-kubernetes-wordpress \
  --member="serviceAccount:terraform@cluster-kubernetes-wordpress.iam.gserviceaccount.com" \
  --role="roles/editor"

gcloud iam service-accounts keys create ~/terraform-key.json \
  --iam-account=terraform@cluster-kubernetes-wordpress.iam.gserviceaccount.com


cloudshell download ~/terraform-key.json
```

J'obtiens un fichier JSON et je l'ai mis au dans mon dossier personnel `cd ~`

terraform-key.json

J’ai activé l’API Compute Engine

```
gcloud services enable compute.googleapis.com \
  --project=cluster-kubernetes-wordpress
```

J’ai activé l’API Kubernetes Engine

```
 gcloud services enable container.googleapis.com  --project=cluster-kubernetes-wordpress
```

J’ai préparé les fichiers Terraform

```
mkdir -p projet_final/terraform
vim main.tf
```

```
# ============================
# PROVIDER GCP
# ============================
provider "google" {
  project = "cluster-kubernetes-wordpress"        # Remplace par ton ID de projet GCP
  region  = "us-central1"           # Région du cluster
  zone    = "us-central1-a"         # Zone pour les VM
  credentials = file("~/terraform-key.json")
}

# ============================
# RESEAU VPC
# ============================
resource "google_compute_network" "k8s_network" {
  name                    = "k8s-network"
  auto_create_subnetworks = true  # GCP crée automatiquement un subnet
}

# ============================
# FIREWALL pour SSH et Kubernetes
# ============================
resource "google_compute_firewall" "k8s_fw" {
  name    = "k8s-firewall"
  network = google_compute_network.k8s_network.name

  allow {
    protocol = "tcp"
    ports    = ["22", "6443", "30000-32767"] # SSH + API Kubernetes + NodePorts
  }

  source_ranges = ["0.0.0.0/0"]
}

# Génération de la clé RSA pour Ansible
resource "tls_private_key" "ansible_key" {
  algorithm = "RSA"
  rsa_bits  = 4096
}

# Sauvegarder la clé privée localement pour Ansible
resource "local_file" "ansible_private_key" {
  content  = tls_private_key.ansible_key.private_key_pem
  # filename = "${path.module}/id_rsa"
  filename = "${path.module}/id_rsa_ansible"
  file_permission = "0600"
}

# ============================
# VM Ansible
# ============================
```

```
resource "google_compute_instance" "ansible" {
  name         = "ansible"
  machine_type = "e2-small"  # 1 vCPU, 2GB RAM
  zone         = "us-central1-a"

  boot_disk {
    initialize_params {
      image = "debian-cloud/debian-12"
      size  = 20
    }
  }

  network_interface {
    network    = google_compute_network.k8s_network.name
    access_config {}  # Attribution IP publique
  }

  metadata = {
    # ssh-keys = "ansible:${tls_private_key.ansible_key.public_key_openssh}"
    ssh-keys = "ansible:${file("~/.ssh/id_rsa.pub")}"
  }

  tags = ["ansible"]
}

# ============================
# CLUSTER GKE
# ============================
resource "google_container_cluster" "wordpress-cluster" {
  name       = "wordpress-cluster"
  location   = "us-central1-a"  # zone unique
  networking_mode = "VPC_NATIVE"

  deletion_protection = false

  # Crée directement un node pool minimal intégré
  initial_node_count = 1

  node_config {
    machine_type = "e2-small"
    disk_size_gb = 20
    disk_type    = "pd-standard"
    oauth_scopes = ["https://www.googleapis.com/auth/cloud-platform"]
  }
}

# ============================
# NODE POOL GKE
# ============================
# # le vrai
resource "google_container_node_pool" "primary_nodes" {
  name       = "primary-pool"
  cluster    = google_container_cluster.wordpress-cluster.name
  location   = google_container_cluster.wordpress-cluster.location
  node_count = 1

  node_config {
    machine_type = "e2-medium"
    disk_size_gb = 20
    disk_type    = "pd-standard"

    oauth_scopes = [
      "https://www.googleapis.com/auth/cloud-platform"
    ]
  }
}

# ============================
# PERSISTENT DISK pour PV (optionnel)
# ============================
```

```
resource "google_compute_disk" "mysql_pv" {
  name  = "mysql-pv-disk"
  type  = "pd-standard"
  zone  = "us-central1-a"
  size  = 10  # 50 GB
}
```

J’ai lancé terraform

```
terraform init
terraform plan
terraform apply
```

Affichage des ressources crées

```
amine@PRO:~/projet_final/terraform$ terraform state list
google_compute_disk.mysql_pv
google_compute_firewall.k8s_fw
google_compute_instance.ansible
google_compute_network.k8s_network
google_container_cluster.wordpress-cluster
google_container_node_pool.primary_nodes
```

Configuration avec Ansible

J’ai généré une clé SSH sur ma machine locale

```
ssh-keygen -t rsa -b 4096 -C "WSL-Moba"
```

J’ai inséré la clé publique dans la machine "ansible" au niveau du fichier "main.tf"

J’ai ouvert une session SSH dans Mobaxterm avec la clé privée.

Sur la machine ansible

J’ai installé ansible

```
sudo apt update


sudo apt install -y ansible
sudo apt install -y python3-kubernetes python3-openshift
ansible --version
```

J’ai Installé la collection Kubernetes et le client Python Kubernetes

```
ansible-galaxy collection install kubernetes.core
# pip3 install kubernetes
python3 -c "import kubernetes; print(kubernetes.__version__)"
```

J’ai insallé kubectl et gcloud

```
sudo apt update && sudo apt install -y google-cloud-sdk-gke-gcloud-auth-plugin kubectl
```

J’ai authentifié à mon compte gcp

```
gcloud auth login

# ou dans wsl local
gcloud auth application-default login
```

Pour s'authentifier à mon compte gcp avec les credentiels sur ma machine locale

```
amine@PRO:~$ export GOOGLE_APPLICATION_CREDENTIALS="/home/amine/terraform-key.json"
amine@PRO:~$ gcloud auth activate-service-account --key-file=$GOOGLE_APPLICATION_CREDENTIALS

Activated service account credentials for: [terraform@cluster-kubernetes-wordpress.iam.gserviceaccount.com]

# installation de kubectl (et de gke-gcloud-auth-plugin s’il faut)

gcloud components install kubectl

# gcloud components install gke-gcloud-auth-plugin



gcloud container clusters get-credentials wordpress-cluster --zone us-central1-a --project
cluster-kubernetes-wordpress
```

J’ai configuré kubeconfig pour accéder au cluster GKE :

```
gcloud container clusters get-credentials wordpress-cluster --zone us-central1-a --project
cluster-kubernetes-wordpress
```

```
kubectl get nodes
```

```
ansible@ansible:~$ kubectl get nodes
NAME                                                STATUS   ROLES    AGE     VERSION
gke-wordpress-cluster-default-pool-165ab838-8dgq    Ready    <none>   4h53m   v1.33.3-gke.1136000
gke-wordpress-cluster-primary-pool-622319d0-9sq4    Ready    <none>   4h47m   v1.33.3-gke.1136000
```

Je vois ben mes podes

J’ai redémarré la machine

```
sudo reboot
```

Dans ma machine locale, j’ai copié la clé privé dans la machine ansible

```
amine@PRO:~$ scp /home/amine/.ssh/id_rsa ansible@34.136.11.20:/home/ansible/.ssh/id_rsa
```

je copie également le fichier ~/.kube/config

```
scp ~/.kube/config ansible@34.136.11.20:~/.kube/config
```

J'ai créé les fichiers ansible

```
mkdir -p projet_kube_WordPress/ansible
cd projet_kube_WordPress/ansible
```

```
nano hosts.ini

[ansible]
34.136.11.20  ansible_user=ansible ansible_ssh_private_key_file=~/.ssh/id_rsa
```

```
ansible -i hosts.ini all -m ping
```

j'ai déployer mysql avec ansible

```
nano mysql.yml
```

```yaml
---
- name: Déployer MySQL sur GKE
  hosts: ansible
  become: false
  gather_facts: false
  collections:
    - kubernetes.core

  tasks:
    - name: Créer le namespace wordpress
      k8s:
        api_version: v1
        kind: Namespace
        name: wordpress
        state: present

    - name: Déployer le Secret MySQL
      k8s:
        api_version: v1
        kind: Secret
        namespace: wordpress
        name: mysql-secret
        state: present
        definition:
          type: Opaque
          data:
            mysql-root-password: "{{ 'rootpassword' | b64encode }}"
            mysql-user-password: "{{ 'userpassword' | b64encode }}"

    - name: Créer le PersistentVolumeClaim pour MySQL
      k8s:
        api_version: v1
        kind: PersistentVolumeClaim
        namespace: wordpress
        name: mysql-pvc
        state: present
        definition:
          spec:
```

```
        accessModes:
          - ReadWriteOnce
        resources:
          requests:
            storage: 10Gi

- name: Déployer MySQL
  k8s:
    api_version: apps/v1
    kind: Deployment
    namespace: wordpress
    name: mysql
    state: present
    definition:
      spec:
        replicas: 1
        selector:
          matchLabels:
            app: mysql
        template:
          metadata:
            labels:
              app: mysql
          spec:
            containers:
              - name: mysql
                image: mysql:8.0
                env:
                  - name: MYSQL_ROOT_PASSWORD
                    valueFrom:
                      secretKeyRef:
                        name: mysql-secret
                        key: mysql-root-password
                  - name: MYSQL_PASSWORD
                    valueFrom:
                      secretKeyRef:
                        name: mysql-secret
                        key: mysql-user-password
                  - name: MYSQL_USER
                    value: wordpress
                  - name: MYSQL_DATABASE
                    value: wordpress
                ports:
                  - containerPort: 3306
                volumeMounts:
                  - name: mysql-data
                    mountPath: /var/lib/mysql
            volumes:
              - name: mysql-data
                persistentVolumeClaim:
                  claimName: mysql-pvc

- name: Créer un Service ClusterIP pour MySQL
  k8s:
    api_version: v1
    kind: Service
    namespace: wordpress
    name: mysql
    state: present
    definition:
      spec:
        selector:
          app: mysql
        ports:
          - port: 3306
            targetPort: 3306
```

J’ai deployer WordPress

nano wordpress.yml

```
---
- name: Déployer WordPress sur GKE
  hosts: ansible
  become: false
  gather_facts: false
  collections:
    - kubernetes.core

  tasks:
    - name: Créer le namespace wordpress (si pas déjà créé)
      k8s:
        api_version: v1
        kind: Namespace
        name: wordpress
        state: present

    - name: Déployer WordPress
      k8s:
        api_version: apps/v1
        kind: Deployment
        namespace: wordpress
        name: wordpress
        state: present
        definition:
          spec:
            replicas: 1
            selector:
              matchLabels:
                app: wordpress
            template:
              metadata:
                labels:
                  app: wordpress
              spec:
                containers:
                  - name: wordpress
                    image: wordpress:6.4-php8.2-apache
                    env:
                      - name: WORDPRESS_DB_HOST
                        value: mysql.wordpress.svc.cluster.local
                      - name: WORDPRESS_DB_USER
                        value: wordpress
                      - name: WORDPRESS_DB_PASSWORD
                        valueFrom:
                          secretKeyRef:
                            name: mysql-secret
                            key: mysql-user-password
                      - name: WORDPRESS_DB_NAME
                        value: wordpress
                    ports:
                      - containerPort: 80

    - name: Créer un Service LoadBalancer pour WordPress
      k8s:
        api_version: v1
        kind: Service
        namespace: wordpress
        name: wordpress
        state: present
        definition:
          spec:
            type: LoadBalancer
            selector:
              app: wordpress
            ports:
              - port: 80
```

```
    targetPort: 80
```

Les services WordPress et MySQL sont bien déployés.

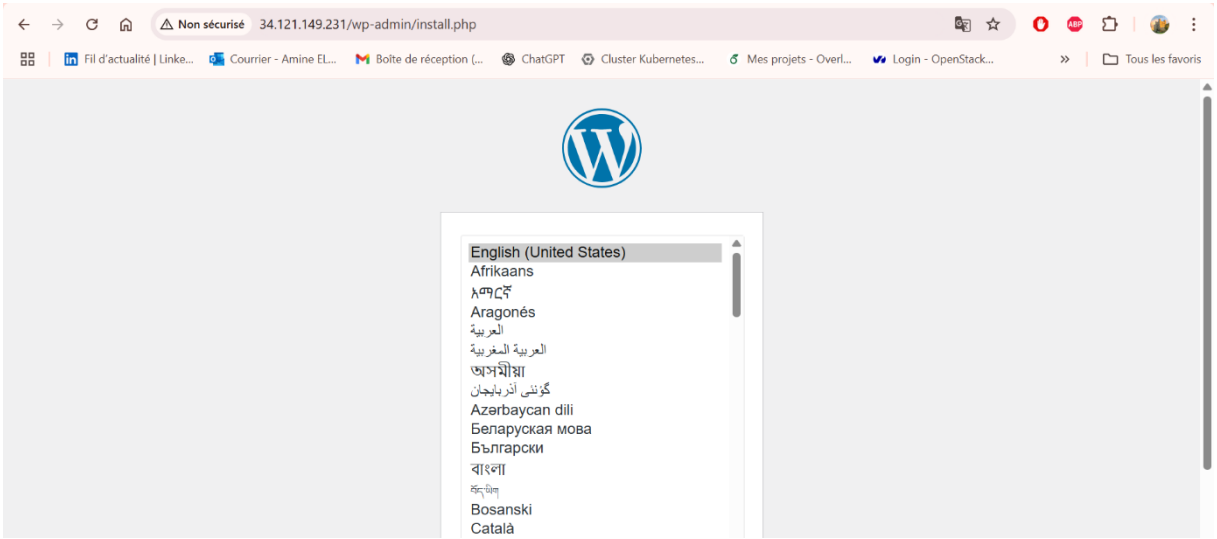
```
kubectl get svc -n wordpress
```

```
ansible@ansible:~$ kubectl get svc -n wordpress
NAME         TYPE           CLUSTER-IP       EXTERNAL-IP      PORT(S)        AGE
mysql        ClusterIP      34.118.238.149   <none>           3306/TCP       6h50m
wordpress    LoadBalancer   34.118.231.28    34.121.149.231   80:32409/TCP   6h48m
```

```
kubectl get pods -n wordpress
```

```
ansible@ansible:~/projet_kube_WordPress/ansible$ kubectl get pods -n wordpress
NAME                         READY   STATUS    RESTARTS   AGE
mysql-7b94c76c57-czjzk       1/1     Running   0          20m
wordpress-676bdb7d59-krtl9   1/1     Running   0          18m
```

Dans un navigateur, j'affiche la page de WordPress avec l'IP publique :



Les couts

	Service	Coût d'utilisation	Programmes de remises	Autres remises	Sous-total
◆	Compute Engine	0,47 €	–	-0,47 €	0,00 €
■	Kubernetes Engine	0,71 €	–	-0,71 €	0,00 €
●	Networking	0,18 €	–	-0,18 €	0,00 €
▼	Cloud Monitoring	0,08 €	–	-0,08 €	0,00 €