**Rapport de Projet**

**Déploiement d'une Application Node.js sur Azure avec Terraform et Ansible**

**1. Objectif du Projet**

L’objectif est de déployer automatiquement une application Node.js sur une machine virtuelle Ubuntu hébergée sur Microsoft Azure.  
Ce déploiement est réalisé en deux étapes :

* Provisionnement de l’infrastructure avec **Terraform**
* Configuration logicielle et déploiement de l’application avec **Ansible**

**2. Étapes du Projet**

**2.1 Définition de l’Infrastructure avec Terraform**

**a. Provider Azure (azurerm)**

Permet à Terraform d’interagir avec les ressources Microsoft Azure.

**b. Resource Group**

Création d’un conteneur logique pour regrouper toutes les ressources du projet.

**c. Réseau Virtuel et Sous-Réseaux**

Un réseau virtuel Azure est créé avec deux sous-réseaux :

* Un pour la machine virtuelle (vm-subnet)
* Un pour une base de données MySQL (mysql-subnet), avec délégation de service.

**d. Adresse IP Publique**

Permet d’accéder à la machine virtuelle depuis Internet (SSH + Application).

**e. Interface Réseau (NIC)**

Connecte la machine virtuelle au réseau, à son sous-réseau et à l’IP publique.

**f. Groupe de Sécurité Réseau (NSG)**

Définit des règles pour autoriser :

* Le port **22** pour SSH
* Le port **3000** pour l’application Node.js

**g. Machine Virtuelle Ubuntu**

Création d'une VM Ubuntu avec les caractéristiques suivantes :

* Taille : Standard\_B1s
* OS : Ubuntu Server 22.04 LTS
* Authentification : par mot de passe (ou par clé SSH)
* Rattachée à l’interface réseau, à l’IP publique, et au NSG

**2.2 Génération Dynamique de l’inventory Ansible**

Après le déploiement de l’infrastructure, on récupère l’adresse IP publique de la machine et on génère dynamiquement le fichier inventory.ini pour Ansible :

bash

CopyEdit

IP=$(curl -s ifconfig.me)

echo "[deploy-vm]" > inventory.ini

echo "$IP ansible\_user=azureuser ansible\_password=rootroot123! ansible\_connection=ssh" >> inventory.ini

**2.3 Déploiement de l’Application avec Ansible**

Le playbook Ansible effectue les opérations suivantes sur la machine :

1. **Installation des outils de base**
   * Installe git, nodejs, npm via le gestionnaire de paquets APT.
2. **Installation de PM2**
   * PM2 est installé globalement avec npm pour gérer le processus Node.js.
3. **Clonage du Dépôt GitHub**
   * Récupération de l’application via git clone.
4. **Installation des Dépendances Node.js**
   * npm install dans le dossier du projet.
5. **Démarrage de l’Application avec PM2**
   * L’application est démarrée en tant qu’utilisateur azureuser.
6. **Configuration de PM2 pour le redémarrage automatique**
   * pm2 startup et pm2 save permettent de redémarrer automatiquement l'application après un reboot.

**3. Tests et Résultat Attendu**

Une fois toutes les étapes complétées, l’application Node.js est accessible depuis un navigateur web à l’adresse :

http://<IP\_PUBLIQUE>:3000

On peut aussi vérifier le fonctionnement de l’application avec la commande suivante sur la VM :

pm2 list