

## **Rapport de Projet - Déploiement d'une Infrastructure avec Terraform sur Azure**

### **Introduction**

Ce rapport détaille le processus de déploiement d'une infrastructure sur Microsoft Azure en utilisant Terraform. L'objectif est d'établir une architecture cloud automatisée et reproductible pour répondre aux besoins d'une application ou d'un service.

### **Étape 1 : Configuration de l'Environnement**

#### 1. Installation de Terraform

Pour ma part, j'ai utilisé Chocolatey sur Windows pour installer Terraform.

```
```bash
```

Installation de Terraform avec Chocolatey (sur Windows)

```
choco install terraform
```

```
```
```

#### 2. Configuration de l'accès à Azure

Connectez-vous à votre compte Azure en utilisant l'Azure CLI.

```
```bash
```

```
az login
```

```
```
```

### **Étape 2 : Création des Fichiers de Configuration Terraform**

#### 1. Fichier provider.tf :

- Le fichier `provider.tf` contient la configuration du fournisseur Azure, spécifiant la région et les informations d'authentification.

#### 2. Fichier main.tf :

- Le fichier `main.tf` définit les ressources Azure telles que les réseaux, les machines virtuelles, les groupes de ressources, etc.

- Utilisation des ressources de données Azure pour récupérer des informations existantes.

### 3. Fichier output.tf :

- Le fichier `output.tf` définit les sorties pour afficher les informations utiles après le déploiement.

### 4. Fichier data.tf :

- Utilisation des ressources de données pour récupérer des informations existantes sur Azure, telles que les sous-réseaux, les groupes de sécurité réseau, etc.

## **Étape 3 : Exécution de Terraform**

### 1. Initialisation du Projet Terraform:

```
```bash
terraform init
```
```

### 2. Validation des Changements :

```
```bash
terraform plan
```
```

### 3. Application des Changements :

```
```bash
terraform apply
```
```

## **Étape 4 : Configuration supplémentaire avec Chocolatey**

### 1. Installation de Docker avec Chocolatey:

- Créez un script ``install_docker.ps1`` pour installer Docker à l'aide de Chocolatey.

### 2. Intégration dans la Configuration Terraform:

- Ajoutez une ressource de provisionnement dans le fichier ``main.tf`` pour exécuter le script ``install_docker.ps1`` après le déploiement de la machine virtuelle.

Fichier `terraform.tfstate` :

Le fichier ``terraform.tfstate`` est un fichier de stockage local qui contient l'état actuel des ressources gérées par Terraform. Il inclut des informations sur les ressources créées, telles que leur ID, leur type, leur nom et leurs attributs. Ce fichier est utilisé par Terraform pour suivre l'état de l'infrastructure et pour effectuer des opérations de gestion telles que la mise à jour, la destruction ou la récupération des ressources. Il est important de ne pas modifier manuellement ce fichier, car il est généralement géré par Terraform lui-même.

## **Étape 5 : Finalisation du projet Terraform**

### 1. Exécution de Terraform `fmt` pour formater le code :

```
```bash
terraform fmt
```
```

### 2. Validation et application des changements :

- Exécution de ``terraform plan`` pour valider les changements proposés et ``terraform apply`` pour les appliquer.

### 3. Génération des clés SSH et autorisation d'accès :

- Génération de la Clé SSH Privée : ``terraform output -raw tls_private_key > id_rsa``
- Création des clés SSH avec ``ssh-keygen`` et modification des permissions avec ``chmod 600 id_rsa`` pour sécuriser l'accès.

### 4. Connexion à la machine virtuelle via SSH :

- Récupération de l'Adresse IP Publique : ``terraform output public_ip_address``
- Utilisation de la clé SSH pour se connecter à la machine virtuelle avec ``ssh -i id_rsa devops@public_ip_address``

## **Conclusion**

Ce rapport a décrit en détail le processus de déploiement d'une infrastructure sur Microsoft Azure en utilisant Terraform. Chaque étape, depuis la configuration de l'environnement jusqu'à l'intégration de Docker avec Chocolatey, a été abordée. L'utilisation de Terraform offre une approche efficace et reproductible pour la gestion automatisée des ressources cloud, offrant ainsi un meilleur contrôle et une meilleure maintenabilité par rapport à l'utilisation de la CLI ou de l'interface utilisateur d'Azure.