**Documentation Complète du Déploiement Multi-Régional avec Terraform**

**1. Introduction**

Ce document décrit la mise en place d'une infrastructure cloud multi-régionale à l'aide de Terraform dans le cadre du projet *Déploiement Multi-régional d'une Application Web*. Il inclut la gestion des instances EC2, des bases de données, du réseau et du load balancer, ainsi que l'intégration avec un pipeline CI/CD et des outils de monitoring.

**2. Contexte du Projet**

Le projet consiste à déployer une application web open-source en respectant les meilleures pratiques DevOps et en assurant une haute disponibilité grâce à un déploiement multi-régional (France-Allemagne).

**Objectifs pédagogiques**

1. **Infrastructure cloud moderne** : gestion des régions, haute disponibilité, sauvegardes.
2. **Automatisation** : CI/CD, tests automatisés.
3. **Sécurité et monitoring** : gestion des accès, métriques et logs.
4. **Documentation et présentation des résultats**.

**3. Fichiers Terraform et Leur Rôle**

**3.1 main.tf - Fichier Principal**

Ce fichier contient la déclaration du provider AWS et la création des ressources essentielles (instances EC2, bases de données, réseau, etc.).

**3.2 compute.tf - Gestion des Instances EC2**

* Provisionne les serveurs de l'application.
* Définit les types d'instances et les configurations spécifiques.

**3.3 database.tf - Gestion des Bases de Données**

* Déploie un RDS multi-régional pour assurer la redondance des données.
* Définit les paramètres de connexion et de chiffrement.

**3.4 networking.tf - Gestion du Réseau**

* Création de VPCs, sous-réseaux, et groupes de sécurité.
* Configuration du routage pour permettre la communication entre régions.

**3.5 loadbalancer.tf - Gestion du Load Balancer**

* Déploie un Load Balancer pour distribuer le trafic entre les instances EC2.
* Assure la scalabilité et la haute disponibilité.

**3.6 outputs.tf - Définition des Outputs**

* Affiche les informations essentielles après l'exécution de Terraform (IP publique des instances, ARN des services, etc.).

**3.7 terraform.tfvars - Fichier de Variables**

* Contient les valeurs des variables utilisées pour personnaliser le déploiement.

**3.8 variables.tf - Définition des Variables**

* Définit les variables globales utilisées dans tous les fichiers Terraform.

**4. Déploiement de l'Infrastructure**

**4.1 Installation des outils nécessaires**

sudo apt install -y curl gnupg software-properties-common

curl -fsSL https://apt.releases.hashicorp.com/gpg | sudo tee /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-get update && sudo apt-get install terraform awscli

**4.2 Initialisation de Terraform**

terraform init

terraform plan

terraform apply

**4.3 Vérification des Instances Déployées**

aws ec2 describe-instances --query 'Reservations[\*].Instances[\*].[InstanceId,PublicIpAddress]' --output table

**5. Intégration CI/CD**

**5.1 Développement du pipeline CI/CD**

* Automatisation des tests unitaires avec GitLab CI.
* Déploiement continu avec AWS CodePipeline.

**5.2 Mise en place des sauvegardes automatiques**

* Planification des snapshots RDS et sauvegarde des logs applicatifs.

**6. Monitoring et Sécurité**

**6.1 Outils de Monitoring**

* **Prometheus & Grafana** : collecte des métriques et affichage des dashboards.
* **ELK (Elasticsearch, Logstash, Kibana)** : centralisation et analyse des logs.

**6.2 Sécurité**

* **IAM et politiques d'accès restreintes**.
* **Chiffrement des bases de données et stockage sécurisé**.

**7. Résumé et Livrables**

* **Infrastructure cloud multi-régionale déployée**.
* **Pipeline CI/CD fonctionnel**.
* **Système de monitoring et sécurité renforcée**.
* **Documentation technique détaillée et présentation du projet**.