

Terraform × AWS – High Availability Web Infrastructure

❖ Objectif du projet

Ce projet vise à déployer automatiquement, à l'aide de **Terraform**, une infrastructure web **hautement disponible** sur **AWS**, conforme aux bonnes pratiques cloud, et compatible avec les contraintes **AWS Academy**.

L'architecture déployée comprend :

- un **VPC dédié**
- des **subnets publics et privés** répartis sur **2 Availability Zones**
- un **Application Load Balancer (ALB)**
- un **Auto Scaling Group (ASG)** d'instances EC2
- un **NAT Gateway** pour l'accès sortant des instances privées
- une **base de données RDS MySQL**

☒ Architecture déployée

Composants principaux

- **VPC** : **10.0.0.0/16**
- **Subnets publics** : hébergent l'ALB et le NAT Gateway
- **Subnets privés** : hébergent les instances EC2 et la base de données
- **ALB (HTTP 80)** : point d'entrée public
- **ASG** : minimum 2 instances EC2 réparties sur 2 AZ
- **RDS MySQL** : base privée, non accessible publiquement

Sécurité

- **Security Groups stricts** :
 - **ALB** → autorise HTTP depuis Internet
 - **Web** → autorise HTTP uniquement depuis l'ALB
 - **DB** → autorise MySQL uniquement depuis les instances Web
- Aucune instance privée ne possède d'IP publique

📁 Structure du projet

```
terraform_project/
|
├── providers.tf
├── versions.tf
├── variables.tf
├── locals.tf
├── vpc.tf
└── security_groups.tf
```

```
└── alb.tf
└── asg.tf
└── rds.tf
└── outputs.tf
└── terraform.tfvars
└── README.md
```

⚙️ Prérequis

- Compte AWS Academy actif
 - Région obligatoirement : us-east-1
 - Terraform ≥ 1.5
 - AWS CLI configuré avec les credentials du lab
 - PowerShell (Windows) ou terminal équivalent
-

🔒 Configuration AWS Academy

Récupération des credentials du Lab

Depuis AWS Academy → AWS Details → AWS CLI credentials

Configurer le profil :

```
aws configure
```

Renseigner :

- Access Key
 - Secret Key
 - Région : us-east-1
 - token session
-

🔧 Déploiement de l'infrastructure

Initialisation :

```
terraform init
```

Vérification du plan :

```
terraform plan
```

Déploiement :

```
terraform apply
```

Confirmer avec :

```
yes
```

🌐 Vérification du fonctionnement

Récupérer l'URL du site :

```
terraform output alb_dns_name
```

Tester dans un navigateur ou en CLI :

```
curl http://<alb_dns_name>
```

Résultat attendu :

- Page web personnalisée
 - Message de bienvenue
 - Load balancing fonctionnel via l'ASG
-

💻 User Data (configuration des instances)

Les instances EC2 utilisent Amazon Linux 2023 et installent automatiquement nginx :

```
dnf -y install nginx
systemctl enable --now nginx
```

Une page HTML personnalisée est générée automatiquement au démarrage.

🗄 Base de données (RDS)

- MySQL 8.0
- Déployée en Single AZ (MultiAZ non disponible pour student aws)
- Subnets privés uniquement
- Accès restreint via Security Group

Endpoint disponible via :

```
terraform output db_endpoint
```

✍ Nettoyage des ressources

Pour supprimer toute l'infrastructure :

```
terraform destroy
```

💻 Technologies utilisées

- Terraform
 - AWS (EC2, VPC, ALB, ASG, RDS)
 - Amazon Linux 2023
 - Nginx
-

👤 Auteur

Tom Deneyer

Projet académique – HEH – Infrastructure Programmable