TD2 – Sécurisation réseau AWS (ALB, Backend, DB)

Objectifs

- Comprendre la segmentation réseau dans AWS
- Identifier les mauvaises pratiques réseau
- Introduire un point d'entrée sécurisé (ALB)
- Contrôler les flux avec les Security Groups
- Appliquer le principe du moindre privilège

Contexte

Vous intervenez dans la même startup.

Une application backend est déployée sur AWS avec une base de données.

Le déploiement a été fait rapidement via Terraform.

Résultat :

- l'application fonctionne
- mais toute l'infrastructure est exposée

Un audit de sécurité impose :

- suppression des accès directs aux composants internes
- mise en place d'une architecture sécurisée
- contrôle strict des flux réseau

Objectif technique

Transformer l'architecture actuelle :

- Internet → Backend (actuel, non sécurisé)

en :

- Internet → ALB → Backend → DB

Contraintes

- L'application doit rester accessible
- Le backend ne doit plus être exposé directement
- La base de données doit être isolée
- Utiliser Terraform uniquement

Infrastructure existante

Fichier : `network/main.tf`

```
provider "aws" {
  region = "eu-west-3"
}

resource "aws_vpc" "main" {
  cidr_block = "10.0.0.0/16"
}

resource "aws_subnet" "public" {
  vpc_id     = aws_vpc.main.id
  cidr_block = "10.0.1.0/24"
}

resource "aws_subnet" "private" {
  vpc_id     = aws_vpc.main.id
  cidr_block = "10.0.2.0/24"
}

resource "aws_security_group" "backend_sg" {
  name   = "backend-sg"
  vpc_id = aws_vpc.main.id

  ingress {
    from_port   = 80
    to_port     = 80
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }
}

resource "aws_security_group" "db_sg" {
  name   = "db-sg"
  vpc_id = aws_vpc.main.id

  ingress {
    from_port   = 5432
    to_port     = 5432
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }
}

resource "aws_instance" "backend" {
  ami           = "ami-123456"
  instance_type = "t2.micro"
  subnet_id     = aws_subnet.public.id
  vpc_security_group_ids = [aws_security_group.backend_sg.id]
}

resource "aws_instance" "db" {
```

```
  ami           = "ami-123456"
  instance_type = "t2.micro"
  subnet_id     = aws_subnet.private.id
  vpc_security_group_ids = [aws_security_group.db_sg.id]
}
```

Ressources

Fichier : `commandes/terraform.txt`

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

1. Compréhension de l’existant

Quels composants sont présents ?

Quelle est l’architecture actuelle ?

Pourquoi cette architecture fonctionne-t-elle malgré ses défauts ?

2. Analyse des problèmes

Identifier les problèmes :

- exposition du backend
- exposition de la base de données
- placement des ressources

Quelle est la règle la plus dangereuse dans ce code ?

Pourquoi ?

3. Mise en pratique – introduction d’un ALB

Objectif :

- créer un point d’entrée unique

Créer :

- un Application Load Balancer
- un Security Group associé
- autoriser HTTP depuis Internet

Fichier : `network/alb.tf`

```
resource "aws_security_group" "alb_sg" {
  name   = "alb-sg"
  vpc_id = aws_vpc.main.id

  ingress {
    from_port   = 80
    to_port     = 80
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }
}
```

4. Erreur volontaire

L'ALB est ajouté.

Mais rien n'a été modifié sur le backend.

Tester :

- accès via ALB
- accès direct au backend

Pourquoi peut-on toujours accéder directement au backend ?

Quel est le problème ?

5. Correction – sécurisation du backend

Modifier le Security Group du backend :

- autoriser uniquement le trafic depuis l'ALB

Fichier : `network/backend_sg.tf`

```
resource "aws_security_group" "backend_sg" {
  name   = "backend-sg"
  vpc_id = aws_vpc.main.id

  ingress {
    from_port       = 80
    to_port         = 80
    protocol        = "tcp"
    security_groups = [aws_security_group.alb_sg.id]
```

```
  }
}
```

6. Correction – sécurisation de la base

Modifier le Security Group de la base :

- autoriser uniquement le backend

Fichier : `network/db_sg.tf`

```
resource "aws_security_group" "db_sg" {
  name   = "db-sg"
  vpc_id = aws_vpc.main.id

  ingress {
    from_port       = 5432
    to_port         = 5432
    protocol        = "tcp"
    security_groups = [aws_security_group.backend_sg.id]
  }
}
```

7. Apparition d’un problème

Après modification :

- l’application peut ne plus répondre

Pourquoi ?

Quels flux ont pu être bloqués ?

8. Analyse

Vérifier :

- règles de Security Groups
- ports utilisés
- configuration du backend

9. Correction

Corriger les règles pour :

- restaurer le fonctionnement

- conserver la sécurité

10. Extension

Ajouter :

- HTTPS sur l'ALB
- redirection HTTP → HTTPS

Challenge final

Objectif :

- seul l'ALB est accessible depuis Internet
- le backend est privé
- la base est isolée

Mettre en place :

- 3 Security Groups
- flux stricts entre chaque couche

Dessiner les flux réseau autorisés

Bonus

Ajouter :

- une seconde instance backend
- équilibrage de charge

Quel est l'intérêt du load balancing en termes de sécurité et de disponibilité ?