TD2 – Sécurisation réseau AWS (VPC, Subnets, Security Groups, flux applicatifs)

Objectifs

- Comprendre l'architecture réseau AWS (VPC, subnets, routage)
- Identifier des failles réseau dans une infrastructure existante
- Mettre en place une segmentation réseau (public / privé)
- Contrôler les flux entre services (web, backend, base de données)
- Appliquer le principe du moindre accès réseau avec Terraform

Contexte

Vous intervenez dans une startup.

Une application a été déployée rapidement pour un MVP.

Architecture actuelle :

- un serveur web accessible depuis Internet
- un backend applicatif
- une base de données

Tout fonctionne, mais aucun design réseau n'a été pensé.

Un audit de sécurité impose :

- isolation des composants
- suppression des accès inutiles
- contrôle strict des flux réseau

Objectif technique

Reconcevoir l'architecture réseau pour :

- exposer uniquement le serveur web
- isoler le backend
- sécuriser la base de données
- contrôler précisément les flux entre les composants

Contraintes

- Le serveur web doit rester accessible en HTTP
- Le backend ne doit pas être accessible depuis Internet
- La base de données ne doit jamais être exposée
- Chaque flux doit être justifié
- Toute la configuration doit être faite avec Terraform

Infrastructure existante

Fichier : `network/main.tf`

```
provider "aws" {
  region = "eu-west-1"
}

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

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

resource "aws_instance" "web" {
  ami           = "ami-123456"
  instance_type = "t2.micro"
  subnet_id     = aws_subnet.main.id
}

resource "aws_instance" "backend" {
  ami           = "ami-123456"
  instance_type = "t2.micro"
  subnet_id     = aws_subnet.main.id
}

resource "aws_instance" "db" {
  ami           = "ami-123456"
  instance_type = "t2.micro"
  subnet_id     = aws_subnet.main.id
}

resource "aws_security_group" "all_open" {
  name   = "all-open"
  vpc_id = aws_vpc.main.id

  ingress {
    from_port   = 0
    to_port     = 0
    protocol    = "-1"
    cidr_blocks = ["0.0.0.0/0"]
  }

  egress {
    from_port   = 0
    to_port     = 0
    protocol    = "-1"
    cidr_blocks = ["0.0.0.0/0"]
  }
}
```

Ressources

Commandes utiles :

Fichier : `commandes/terraform.txt`

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

1. Analyse de l’existant

Identifier les problèmes de sécurité présents dans cette infrastructure.

Indiquer :

- les problèmes critiques
- les problèmes de conception

2. Compréhension des flux

Quels sont les flux nécessaires au fonctionnement ?

Lister précisément :

- Internet → Web
- Web → Backend (port ?)
- Backend → DB (port ?)

Quels flux doivent être interdits ?

3. Mise en pratique – segmentation réseau

Modifier l’infrastructure pour créer :

- 1 subnet public (web)
- 1 subnet privé (backend + db)

4. Mise en pratique – accès Internet

Permettre :

- accès Internet au serveur web
- accès sortant pour le backend (updates)

Ajouter les composants nécessaires.

Pourquoi le backend ne doit-il pas avoir d'IP publique ?

5. Mise en pratique – Security Groups

Créer 3 Security Groups :

- web-sg
- backend-sg
- db-sg

Définir les règles suivantes :

- web :
    1. HTTP depuis Internet
- backend :
    1. accessible uniquement depuis web
- db :
    1. accessible uniquement depuis backend

6. Erreur volontaire

Un développeur propose la règle suivante pour le backend :

- autoriser le port 8080 depuis 0.0.0.0/0

Pourquoi cette règle est dangereuse ?

Dans quel cas peut-elle sembler fonctionner correctement ?

7. Apparition d'un problème

Après mise en place des règles :

- le site web fonctionne
- mais l'application ne répond plus correctement

Que vérifier en priorité ?

- Security Groups ?
- ports ?
- flux autorisés ?

8. Correction

Corriger les règles pour :

- restaurer la communication web → backend
- restaurer backend → DB

9. Extension

Ajouter :

- un NAT Gateway
- permettre au backend de faire des mises à jour sans être exposé

Quelle différence entre :

- Internet Gateway
- NAT Gateway ?

10. Bonnes pratiques

Pourquoi utiliser :

- un Security Group par service ?
- des règles basées sur des Security Groups plutôt que des IP ?

Challenge final

Objectif :

- Web accessible depuis Internet (port 80)
- Backend inaccessible depuis Internet
- DB inaccessible depuis Internet
- Backend accessible uniquement depuis Web
- DB accessible uniquement depuis Backend

Implémenter l’architecture complète.

Vérifier que :

- aucun flux inutile n’est autorisé
- chaque règle est justifiée

Bonus

Ajouter :

- un second serveur web
- un Load Balancer

Pourquoi un Load Balancer améliore aussi la sécurité ?