

TD2 - Sécurisation réseau d'une architecture AWS (ALB, Backend, DB)

Objectifs

- Comprendre la segmentation réseau dans AWS - Mettre en place une architecture sécurisée (ALB + backend + DB) - Manipuler les Security Groups - Identifier et corriger une mauvaise exposition réseau - Appliquer le principe du moindre privilège réseau

Contexte

Vous travaillez dans une startup qui déploie une API backend sur AWS.

L'infrastructure actuelle a été déployée rapidement avec Terraform.

Problème :

- le backend est directement accessible depuis Internet - la base de données est mal isolée - aucune réelle segmentation réseau

Objectif métier :

Sécuriser l'infrastructure sans casser le fonctionnement de l'application.

1. Analyse de l'infrastructure existante



Lire la configuration Terraform ci-dessous Identifier les problèmes de sécurité

Fichier : 'main.tf'

```
provider "aws" {  
  region = "eu-west-1"  
}  
  
resource "aws_security_group" "backend_sg" {  
  name = "backend_sg"  
  
  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"

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



Quels sont les deux problèmes majeurs dans cette configuration ? Pourquoi sont-ils critiques ?

2. Mise en pratique - Déploiement



Appliquer la configuration Terraform Observer les accès possibles

- Accéder au backend via son IP publique - Vérifier que le port 5432 est accessible



Pourquoi est-il dangereux que la base de données soit accessible publiquement ?

3. Introduction d'un ALB

Objectif :

- Introduire un point d'entrée unique - Ne plus exposer directement le backend



Ajouter un Application Load Balancer

Fichier : 'alb.tf'

```
resource "aws_security_group" "alb_sg" {
  name = "alb_sg"

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



Pourquoi l'ALB peut-il être exposé publiquement contrairement au backend ?

4. Erreur volontaire - Faux sentiment de sécurité

L'ALB a été ajouté.

Mais rien n'a été modifié côté backend.



Tester les accès : - via ALB - directement via IP du backend



Peut-on toujours accéder directement au backend ? Pourquoi est-ce un problème ?

5. Correction - Isolation du backend

Objectif :

- autoriser uniquement le trafic venant de l'ALB

Fichier : 'backend_sg.tf'

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



Appliquer la correction Tester les accès



Pourquoi utilise-t-on un Security Group comme source plutôt qu'une IP ?

6. Correction - Isolation de la base de données

Objectif :

- autoriser uniquement le backend à accéder à la DB

Fichier : 'db_sg.tf'

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



Appliquer la correction Tester l'accès à la base de données



Pourquoi ne doit-on jamais exposer une base de données sur Internet ?

7. Réflexion globale



Décrire les flux réseau autorisés dans l'architecture finale Quels flux sont explicitement interdits ?

8. Extension



Ajouter une règle HTTPS (port 443) sur l'ALB



Pourquoi HTTPS est indispensable en production ?

Challenge final

Objectif :

- vérifier que toute l'architecture respecte le principe du moindre privilège



Vérifier :

- aucun accès direct Internet → backend - aucun accès direct Internet → DB - seuls les flux nécessaires sont autorisés



Si un attaquant compromet l'ALB, peut-il accéder directement à la base de données ? Pourquoi ?

Bonus



Ajouter un second backend derrière l'ALB



Quel est l'intérêt du load balancing en plus de la sécurité ?

From:
<http://slamwiki2.kobject.net/> - **SlamWiki 2.1**

Permanent link:
<http://slamwiki2.kobject.net/eadl/bloc4/fm4/td2?rev=1781445966>

Last update: **2026/06/14 16:06**

