Contents

1 Créer le projet Terraform 1
2 Définir le provider AWS 1
3 Créer le Security Group pour MySQL 2
4 Créer la base MySQL RDS 2
5 Initialiser et appliquer Terraform 3
6 Tests – Vérifications 3
6.1 Remarques 3
6.2 dans la console AWS 3
6.3 lister toutes les instances RDS 4
6.4 Vérifier le status d'une instance RDS spécifique 4
6.5 Vérifier l'endpoint pour se connecter 5
6.5.1 Test de connexion sur l'endpoint 5
6.6 Vérifier le port 5
6.7 Vérifier le moteur et la version 5

1 Créer le projet Terraform

Dans CMD :

```
mkdir terraform-rds-mysql
cd terraform-rds-mysql
notepad main.tf
```

Cela va créer ton fichier `main.tf` où tu écriras toute la configuration.

---

2 Définir le provider AWS

Dans `main.tf` :

```
terraform {
  required_version = ">= 1.5"
```

```
  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = "~> 5.0"
    }
  }
}

provider "aws" {
  region = "ca-central-1"    # Changer selon la région
}
```

---

3 Créer le Security Group pour MySQL

Ajouter dans le fichier : main.tf, la section ci-dessous :

```
resource "aws_security_group" "rds_sg" {
  name        = "rds-mysql-sg"
  description = "RDS MySQL security group"

  # Pas d'ingress public

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

---

4 Créer la base MySQL RDS

Ajouter dans le fichier : main.tf, la section ci-dessous :

```
resource "aws_db_instance" "mysql" {
  identifier             = "mysql-lab"
  engine                 = "mysql"
  instance_class         = "db.t3.micro"
  allocated_storage      = 20
  username               = "admin"
  password               = "password123"
  publicly_accessible    = false
  vpc_security_group_ids = [aws_security_group.rds_sg.id]
  skip_final_snapshot    = true
}
```

5 Initialiser et appliquer Terraform

Dans CMD, depuis le dossier du projet :

terraform init
terraform plan
terraform apply

- terraform init : initialise le projet et télécharge le provider AWS.
- terraform plan : te montre ce que Terraform va créer.
- terraform apply : applique la configuration. Confirme par yes.

6 Tests – Vérifications

6.1 Remarques

On fait les vérifications avec des commandes CLI au niveau de la console Cloudshell ,(incluse dans la console AWS), sauf pour la section 6.2.
On a crée la base dans un subnet privé, pour cet exemple et on n'a pas configuré le nécessaire pour qu'elle soit accessible en CLI a partir du laptop. On pourrait dans un second temps mettre la base dans un subnet public et configurer le nécessaire pour cela. (mettre dans main.tf au niveau de la création de la base : publicly_accessible = true, et autoriser mon IP (celle publique de mon laptop) ou CloudShell IP sur le port 3306. Pour cette dernière modification il faudrait ajouter la section suivante dans main.tf au niveau de la section : resource "aws_security_group"

```
ingress {
  from_port   = 3306
  to_port     = 3306
  protocol    = "tcp"
  cidr_blocks = ["TON_IP/32"]
}
```

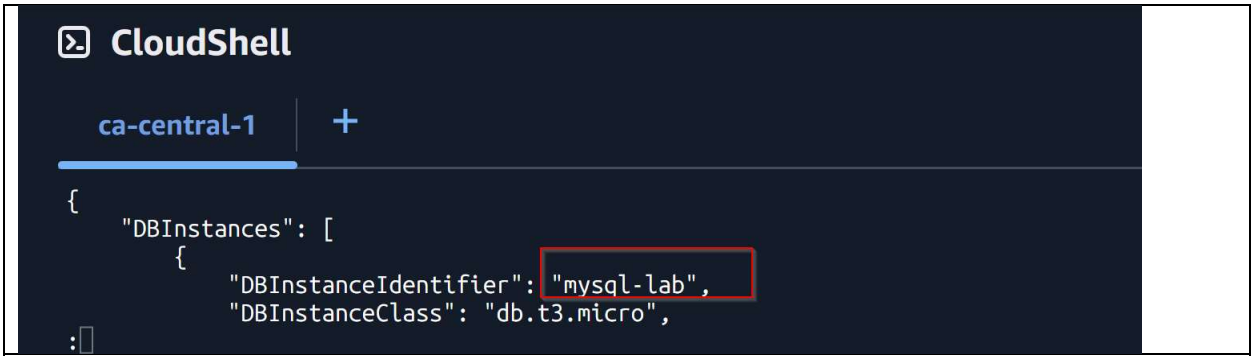
Les vérifications peuvent être faite soit en mode CLI, à partir de la console Cloudshell, soit directement en visuel dans la console AWS , au niveau de la section 'Aurora and RDS – Databases'.
Elles sont faites avec le subnet , en mode privé.

6.2 dans la console AWS

On vérifie la présence de la base, au niveau de la console AWS

6.3 lister toutes les instances RDS

aws rds describe-db-instances



ou plus restrictif :

aws rds describe-db-instances --query 'DBInstances[*].[DBInstanceIdentifier,DBInstanceStatus]' --output table

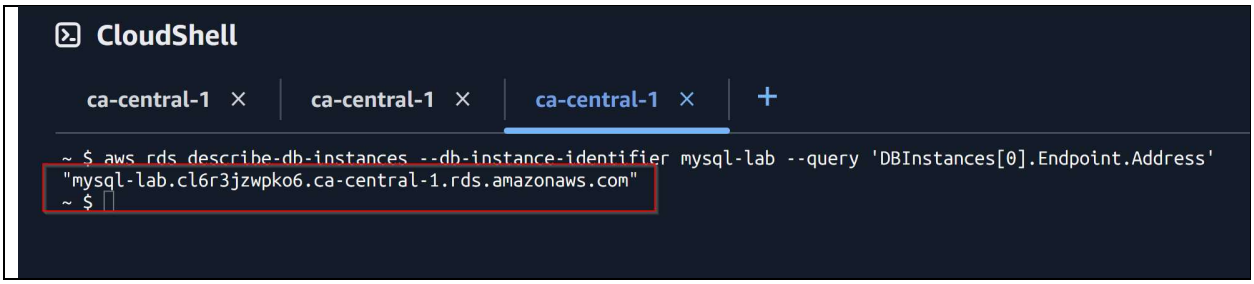
6.4 Vérifier le status d'une instance RDS spécifique

Dans notre cas, nous avons crée seulement une seule instance et il n'existe pour l'instant qu'une seule instance RDS , qui devrait être : mysql-lab

aws rds describe-db-instances --db-instance-identifier mysql-lab

6.5 Vérifier l'endpoint pour se connecter

aws rds describe-db-instances --db-instance-identifier mysql-lab --query 'DBInstances[0].Endpoint.Address'



Endpoint : mysql-lab.cl6r3jzwpko6.ca-central-1.rds.amazonaws.com

6.5.1 Test de connexion sur l'endpoint

Pour tester le endpoint avec une commande mysql qui permettrait d'entrer dans la base et de faire des requêtes SQL, il faudrait que nous mettions en place les remarques , indiquée au paragraphe 6.1, c'est-à-dire créer un subnet de type public et permettre è l'IP de la console cloudshell ou bien selon le cas à l'IP du laptop l a permission adéquate.

Mysql -h endpoint-adress -p 3306 -u username -p password

```
mysql -h mysql-lab.cl6r3jzwpko6.ca-central-1.rds.amazonaws.com -P 3306 -u admin -p
```

6.6 Vérifier le port

aws rds describe-db-instances --db-instance-identifier mysql-lab --query 'DBInstances[0].Endpoint.Port'

par défaut pour MySLQ : 3306

6.7 Vérifier le moteur et la version

aws rds describe-db-instances --db-instance-identifier mysql-lab --query 'DBInstances[0].[Engine,EngineVersion]'