Pour inverser les rôles de tes instances dans un cluster **Amazon Aurora** (i.e., faire de l'instance en eu-west-3a la **writer** et celle en eu-west-3b la **reader**), tu dois **modifier le cluster Aurora** pour promouvoir une instance de lecture en instance d'écriture, et éventuellement **mettre à jour ton code Terraform pour refléter ce changement.**

## 🧠 Concepts clés Aurora

Dans un cluster Aurora :

* Il y a **un writer (instance principale)**.
* Il peut y avoir plusieurs **read replicas (instances secondaires)**.
* Le writer peut être changé manuellement ou via une **failover**.
* Tu ne spécifies pas directement "cette instance est writer" via Terraform ; tu définis les instances et leur **priorité de failover**.

## ✅ Étapes pour changer le writer

### 1. **Forcer un failover vers l'instance en eu-west-3a**

Tu dois provoquer un **failover** du writer vers l’instance de lecture que tu veux promouvoir (celle dans eu-west-3a).

**Option 1 : via AWS CLI (hors Terraform)**

aws rds failover-db-cluster \

--db-cluster-identifier <ton-cluster> \

--target-db-instance-identifier <instance-eu-west-3a>

Cela promeut l’instance spécifiée comme nouvelle writer.

### 2. **Mettre à jour ton code Terraform**

Dans Terraform, tu ne marques **pas** une instance comme writer. Par contre, tu peux définir leur **availability\_zone** et leur **promotion\_tier** pour indiquer leur priorité lors d’un failover :

resource "aws\_rds\_cluster\_instance" "writer" { identifier = "db-writer-eu-west-3a" cluster\_identifier = aws\_rds\_cluster.aurora\_cluster.id instance\_class = "db.r6g.large" engine = "aurora-mysql" availability\_zone = "eu-west-3a" promotion\_tier = 0 # Priorité la plus haute pour devenir writer }

resource "aws\_rds\_cluster\_instance" "reader" { identifier = "db-reader-eu-west-3b" cluster\_identifier = aws\_rds\_cluster.aurora\_cluster.id instance\_class = "db.r6g.large" engine = "aurora-mysql" availability\_zone = "eu-west-3b" promotion\_tier = 1 # Moins prioritaire }

## 🚨 Important

* Terraform **ne déclenche pas le failover automatiquement**. Il ne fait que décrire l’infrastructure. Tu dois déclencher le failover manuellement **ou via un script externe** si tu veux que cela arrive.
* Après le failover, Terraform ne va **pas "revenir en arrière"** et réaffecter le writer précédent, sauf si tu recrées les ressources.

## 💡 Recommandation

1. Utilise le failover-db-cluster manuellement (ou via script).
2. Ajuste tes ressources aws\_rds\_cluster\_instance dans Terraform :
   1. promotion\_tier = 0 pour la nouvelle writer.
   2. promotion\_tier = 1 ou plus pour les lecteurs.
3. Applique via terraform apply.