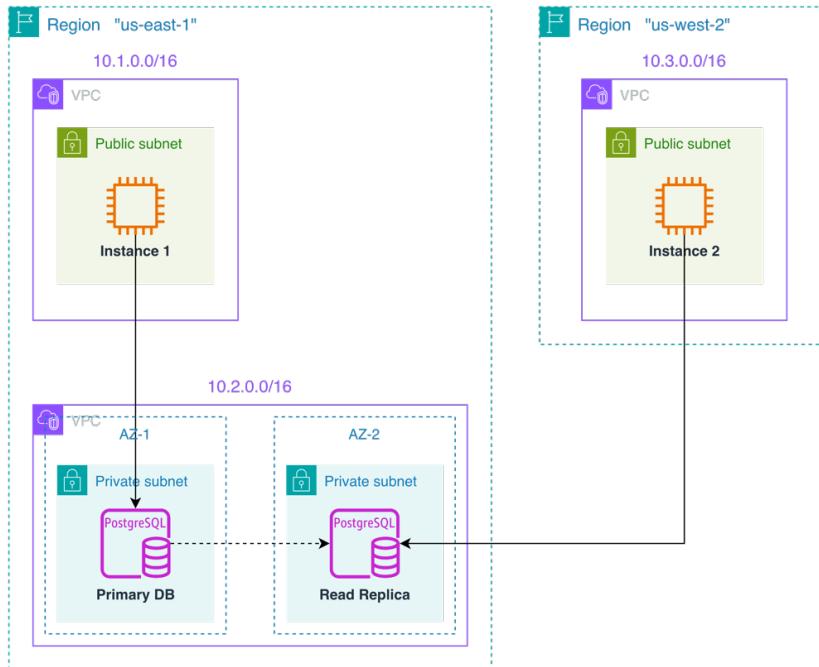


# Évaluation

L'objectif de ce travail est d'implémenter l'architecture ci-dessous sur deux régions différentes :



- Créez votre base de données principale selon la configuration suivante :
  - Engine options : PostgreSQL
  - Templates : Sandbox
  - Donnez des valeurs de votre choix aux paramètres : DB instance identifier, Master username et Master password
  - DB instance class : Burstable classes (db.t3.micro)
  - Sous "Connectivity" sélectionnez votre VPC et votre groupe de sous-réseaux et configurer votre groupe de sécurité.
- Poussez (push) tous les fichiers de l'API qui va communiquer avec la base de données (aws-rds.zip) sur un dépôt distant dans votre compte GitLab.
- Spécifier les configurations suivantes pour les instances EC2 :
  - Amazon machine image (AMI): Amazon Linux 2023
  - Instance type : t3.micro
  - Key pair (login) : une paire de clés existante ou créez une nouvelle paire de clés
  - VPC et sous-réseau : voir figure ci-dessus
  - User data:

```

#!/bin/bash
yum update -y
yum -y install python3-pip
pip3 install fastapi[all] uvicorn sqlalchemy psycopg2-binary pydantic typing-extensions
yum -y install git
git clone https://gitlab_ci_token:<access_token>@gitlab.com/<user_name>/<project_name>.git
/usr/local/bin/uvicorn <project_name>.main:app --host 0.0.0.0 --port 8000

```

- Accédez en SSH à vos instances EC2 et modifiez le fichier "database.py" avec vos paramètres de connexion (à la base de données principale pour "instance 1" et à la copie en lecture pour "instance 2"). Une fois que les différents composants de l'architecture sont créés, collez l'adresse IP de chaque instance "http://adresse\_IP\_de\_votre\_instance:8000" dans la barre d'adresse de navigateur pour afficher le message "Hello World !" Ajouter /docs à votre URL pour afficher et tester les fonctions proposées par l'API de chaque instance. Quel constat faites-vous ?