# Capstone project

PERIN LEA

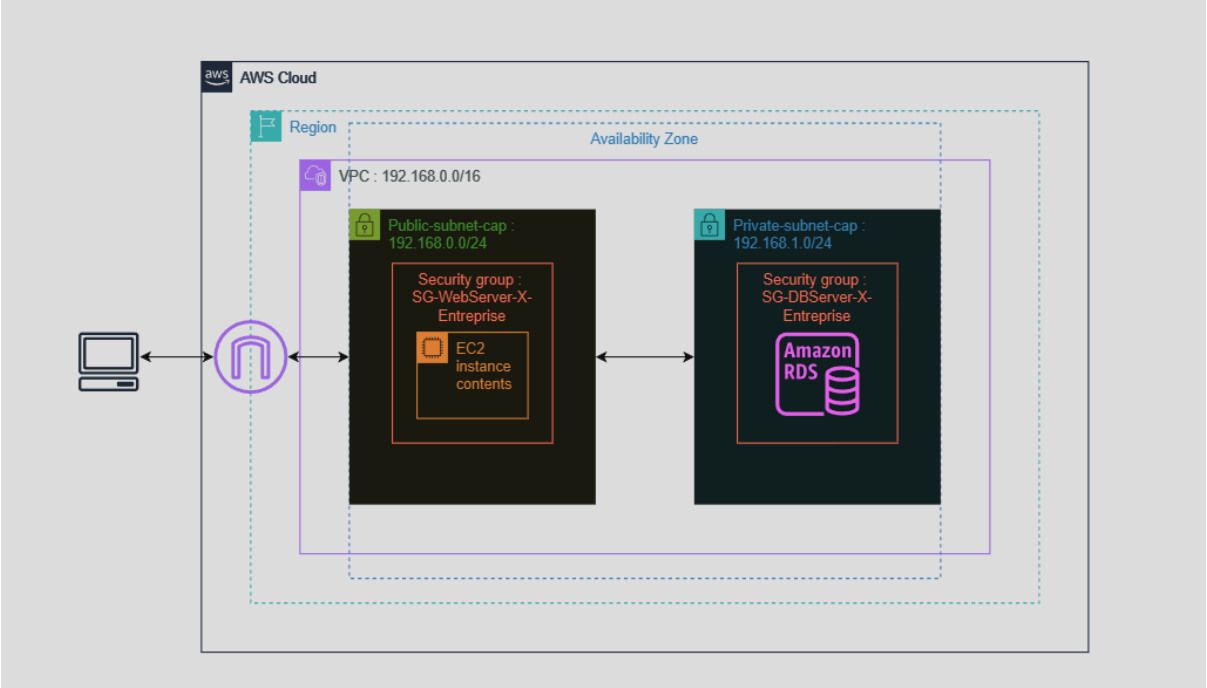
## Sommaire

## 1. Introduction

This document outlines the architectural design and deployment strategy for hosting the X Company's website on the AWS cloud infrastructure.

## 2. Architectural Diagram

The diagram below illustrates the cloud deployment architecture :

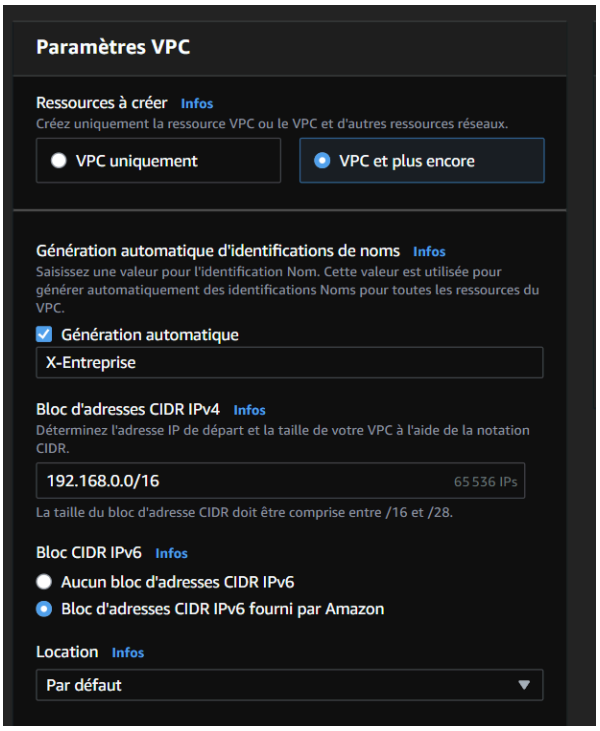



## 3. Deploying Cloud Architecture

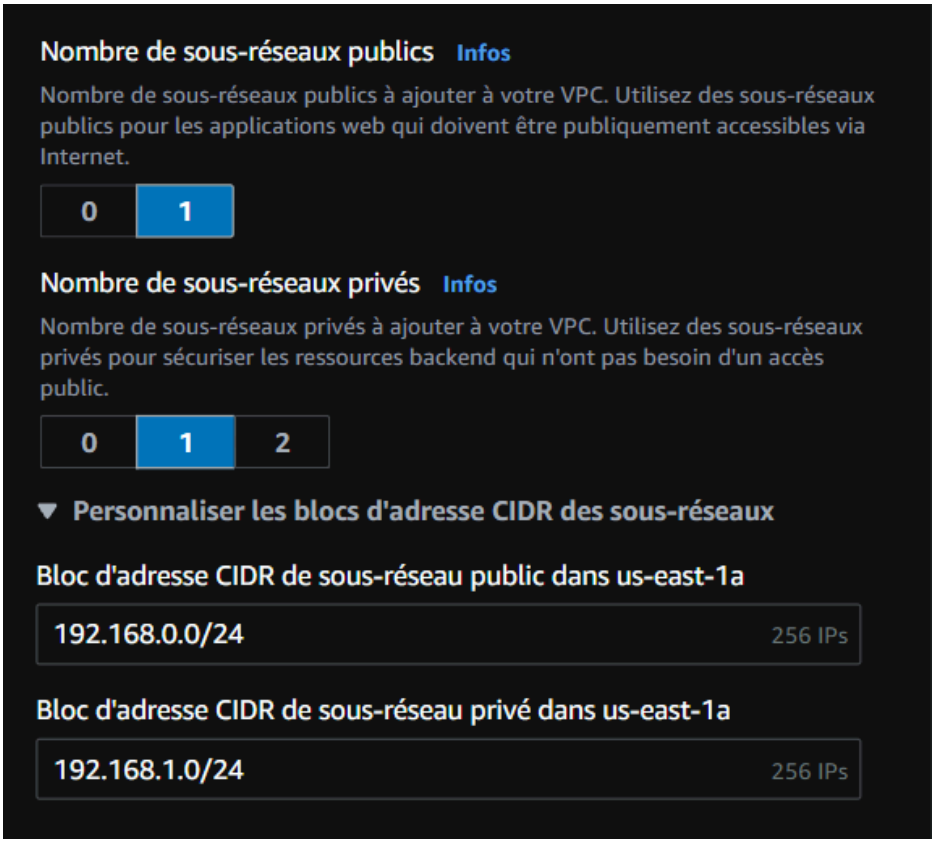
## 3.1 Creating Virtual Private Cloud (VPC)

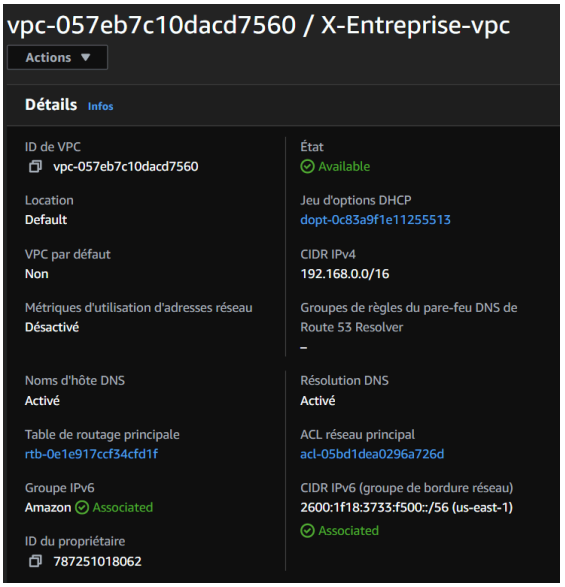
- ➔ Log in to the AWS console.
- ➔ Go to the VPC service and click on "Create VPC".

➔ Specify VPC details such as name, CIDR block




➔ Next, create two subnets: one public subnet and one private subnet, specifying appropriate CIDR ranges.

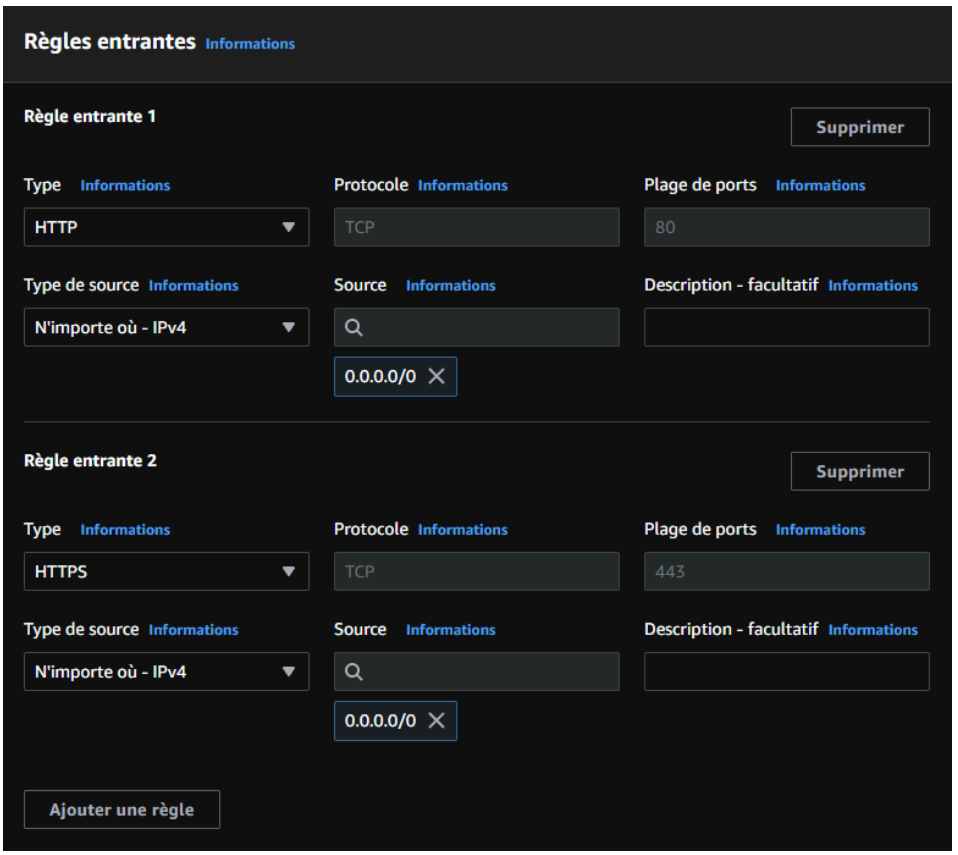


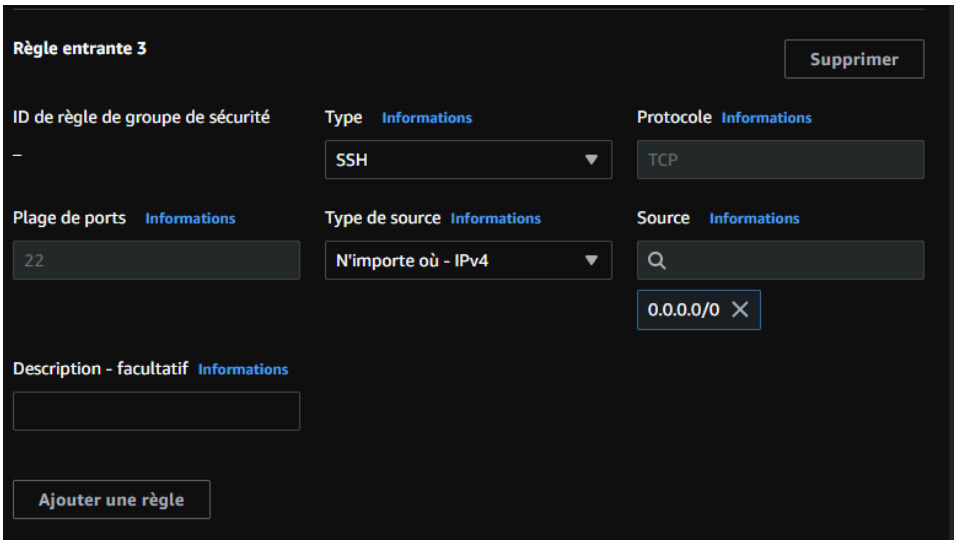



➔ Create security groups to regulate inbound and outbound traffic for EC2 instances. There are two security groups created for this purpose:

➔ SG-WebServer-X-Enterprise (for EC2 instances in the public subnet):

- Inbound Rules :
  Allow HTTP (port 80) traffic from anywhere.
  Allow HTTPS (port 443) traffic from anywhere.
  Allow SSH (port 22) traffic from anywhere.
- Outbound Rules :
  All outbound traffic is allowed by default.




➔ SG-DBServer-X-Enterprise (for RDS database instances in the private subnet) :

- Inbound Rules:
  Allow PostgreSQL (port 5432) traffic from SG-WebServer-X-Enterprise.

- Outbound Rules:
  All outbound traffic is allowed by default

## Détails de base

Nom du groupe de sécurité Informations

SG-DBServer-X-Entreprise

Le nom ne peut pas être modifié après sa création.

Description Informations

*Autorise l'accès SSH aux développeurs*

VPC Informations

vpc-057eb7c10dacd7560 (X-Entreprise-vpc)

## Règles entrantes Informations

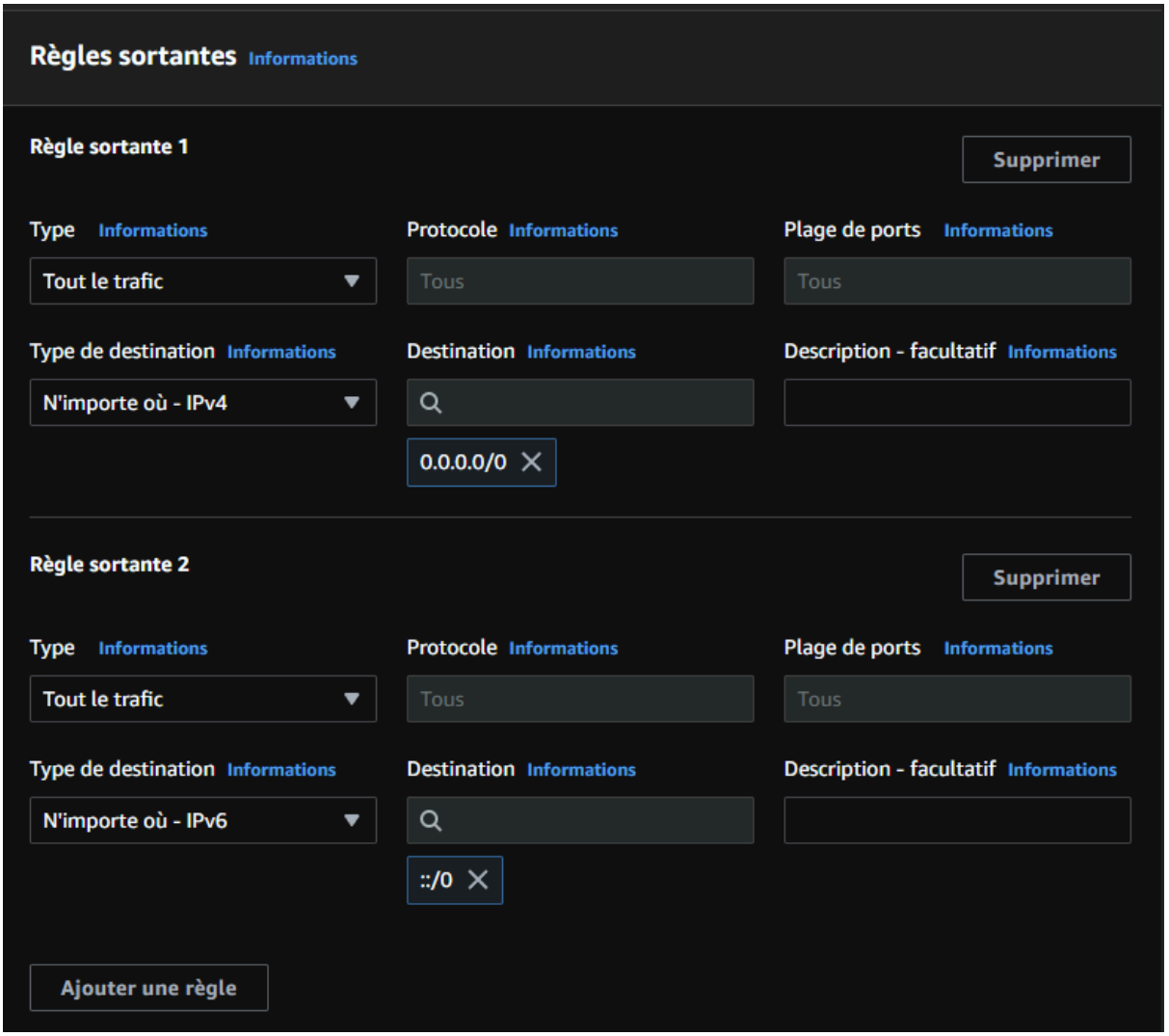
**Règle entrante 1**

Supprimer

| Type Informations | Protocole Informations | Plage de ports Informations |
|---|---|---|
| PostgreSQL | TCP | 5432 |

| Type de source Informations | Source Informations | Description - facultatif Informations |
|---|---|---|
| Personnalisé(e) | sg-0e3ee78c3d9692771 | |
| | sg-0e3ee78c3d9692771 | |

Ajouter une règle




Result :

| Name | ID du groupe de sécurité | Nom du groupe de sécurité | ID de VPC | Description |
|---|---|---|---|---|
| – | sg-0e3ee78c3d9692771 | SG-WebServer-X-Entreprise | vpc-057eb7c10dacd7560 | SG-webserver |
| – | sg-09645734465d5e4ed | default | vpc-057eb7c10dacd7560 | default VPC security group |
| – | sg-09bce223e7fecb8c9 | default | vpc-0ece8e9b175e27be5 | default VPC security group |
| – | sg-06c140100f0409a42 | SG-DBServer-X-Entreprise | vpc-057eb7c10dacd7560 | SG-DBServer |
| – | sg-0e33f6425e9a844f3 | default | vpc-0518753564195588c | default VPC security group |
| – | sg-03065471518afdf03 | Ec2SecurityGroup | vpc-0ece8e9b175e27be5 | VPC Security Group |

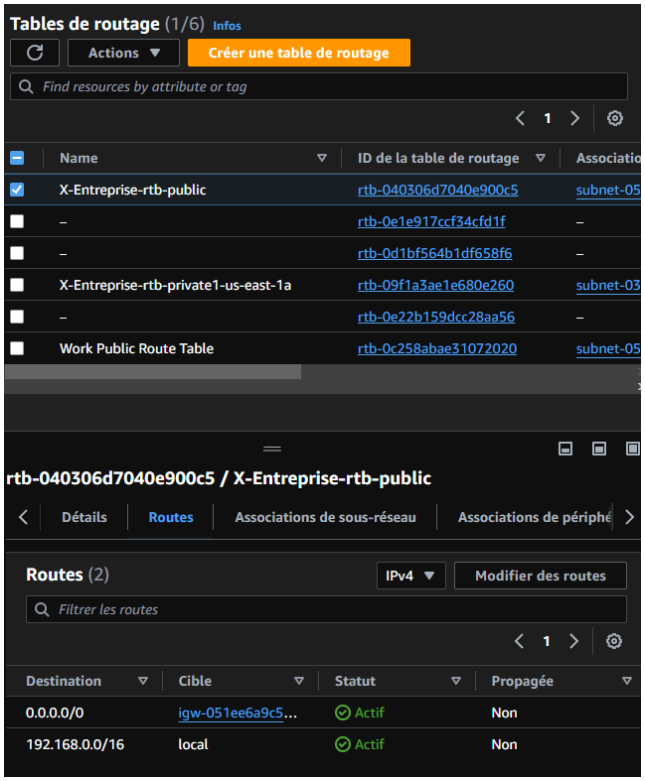
## 3.2 Adding Internet Gateway

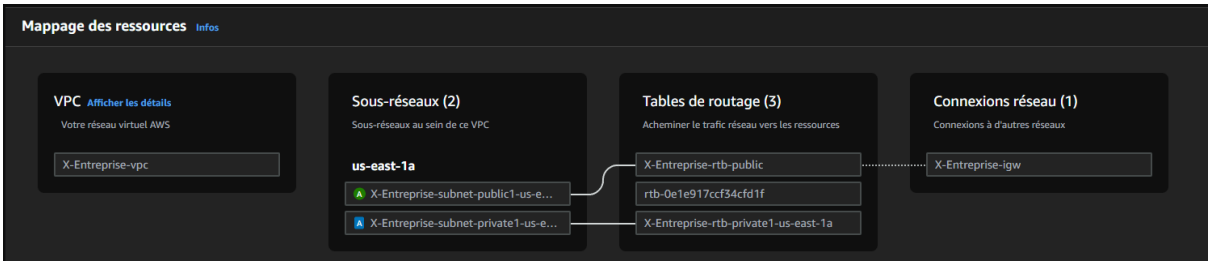
- In the VPC console, select "Internet Gateways" and click on "Create Internet Gateway".
- Attach the Internet Gateway to the VPC created earlier.

  Select the Internet Gateway you just created and click on "Actions" -> "Attach to VPC".

  Select your VPC-X-Enterprise and click on "Attach Internet Gateway".

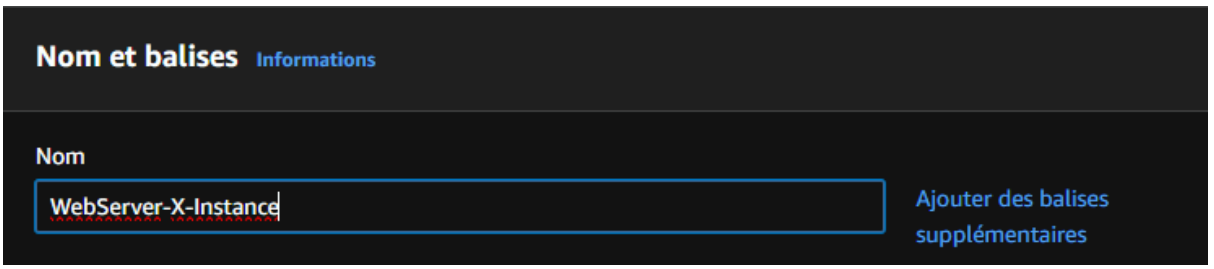
- Modify the public route table to include a route to the Internet Gateway. Click on "Add route" and enter "0.0.0.0/0" as the destination (which means all traffic) and select the Internet Gateway you attached to your VPC as the target.



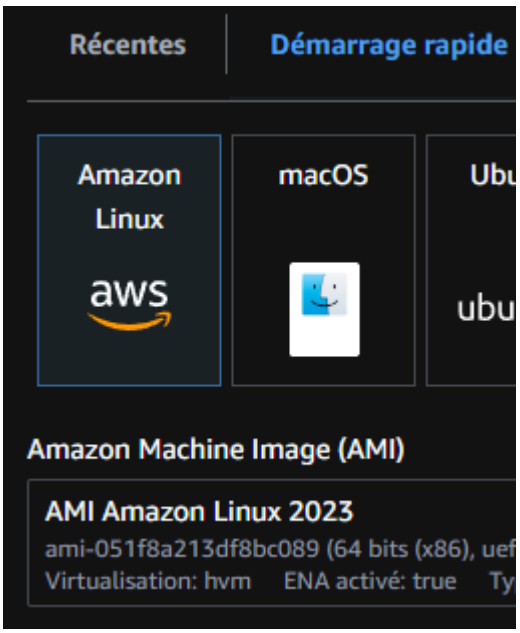




## 3.3 Creating EC2 Instances

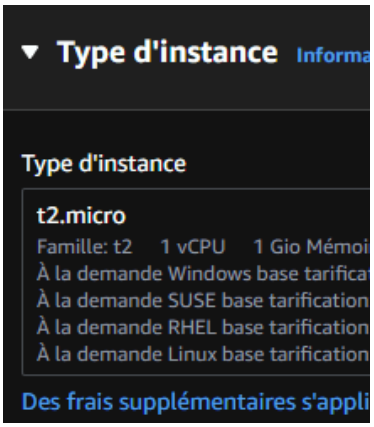
➔ Go to the EC2 console and click on "Launch Instance".



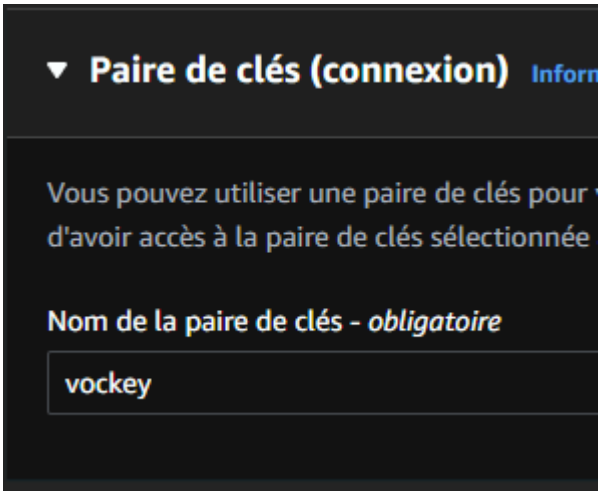

➔ Select the Amazon Linux 2 (version 2023) image.



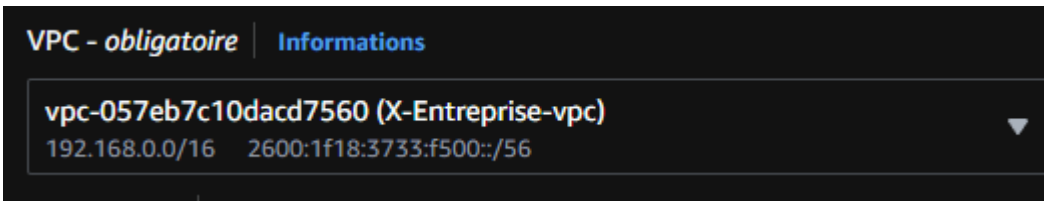

➔ Choose the instance type "t2.micro".



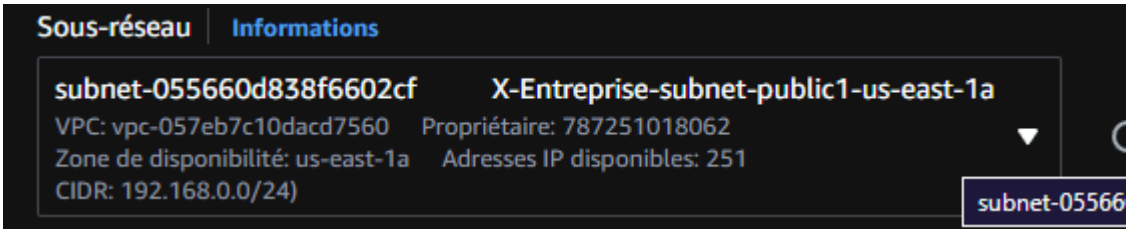

➔ Select the key pair "vockey".



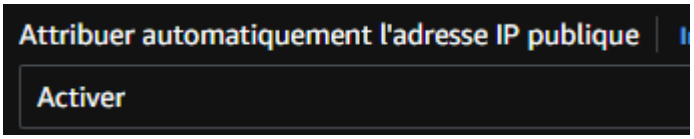

➔ In the network settings, choose your VPC "x-enterprise-vpc".



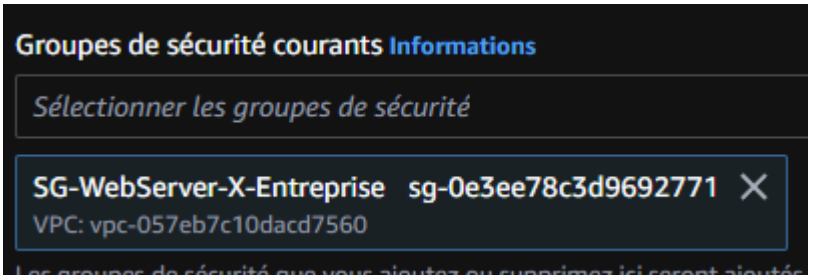

➔ Select your public subnet: "x-enterprise-subnet-public-us-east-1a".




➔ Check the box to enable "Auto-assign Public IP".



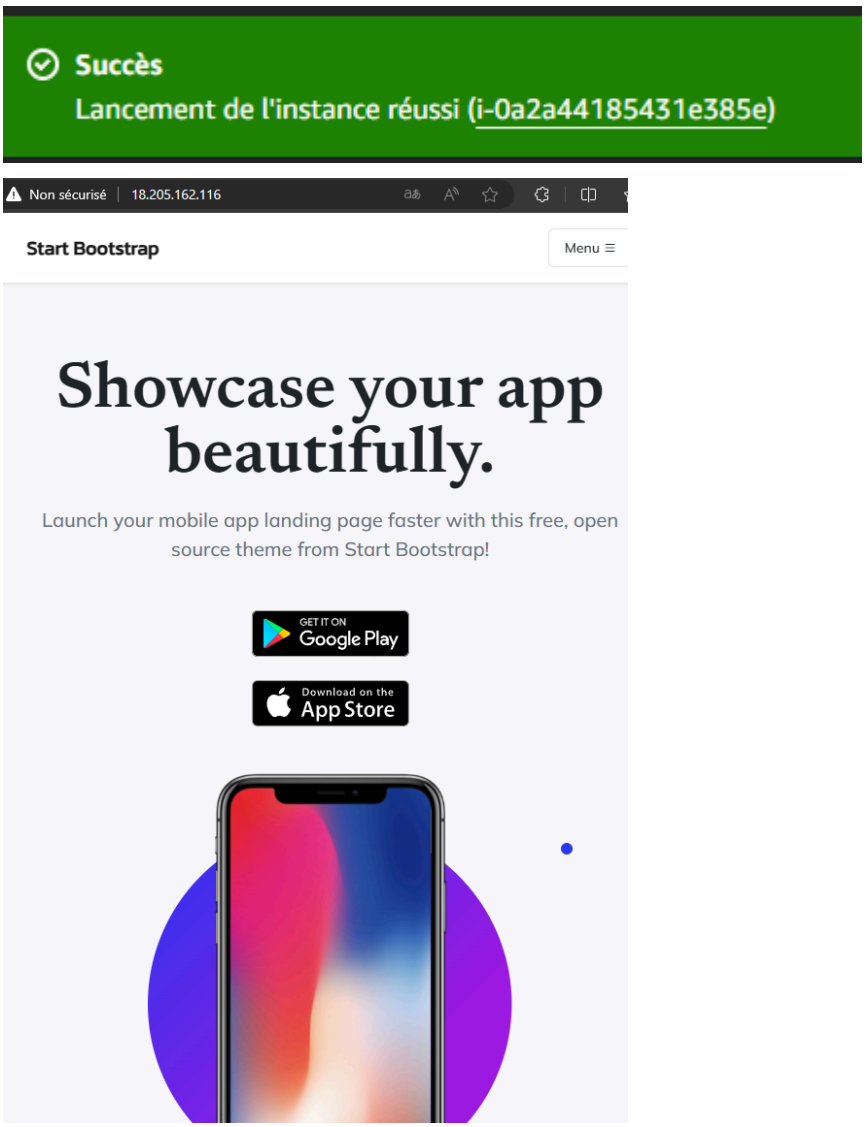

➔ Select the existing security group "sg-webserver-x-enterprise" to allow HTTP and HTTPS traffic.



➔ Copy this script to my EC2 instance, and then execute it to install and start Apache.

```
#!/bin/bash

# Installation d'Apache
echo "Installation d'Apache..."
sudo yum update -y
sudo yum install -y httpd
sudo systemctl start httpd
sudo systemctl enable httpd

# Clonage du dépôt Git
echo "Clonage du dépôt Git..."
git clone https://github.com/Nerlyss1/CAPSTONE-PROJECT.git

# Copie des fichiers vers /var/www/html
echo "Copie des fichiers vers /var/www/html..."
cd CAPSTONE-PROJECT/2024/b3/cloud/sample-app
sudo cp -r * /var/www/html/

# Attribution des permissions
echo "Attribution des permissions..."
sudo chown -R apache:apache /var/www/html
sudo chmod -R 755 /var/www/html

# Redémarrage d'Apache
echo "Redémarrage d'Apache..."
sudo systemctl restart httpd

echo "Installation et déploiement terminés avec succès."
```

➔ Finally, launch the instance.

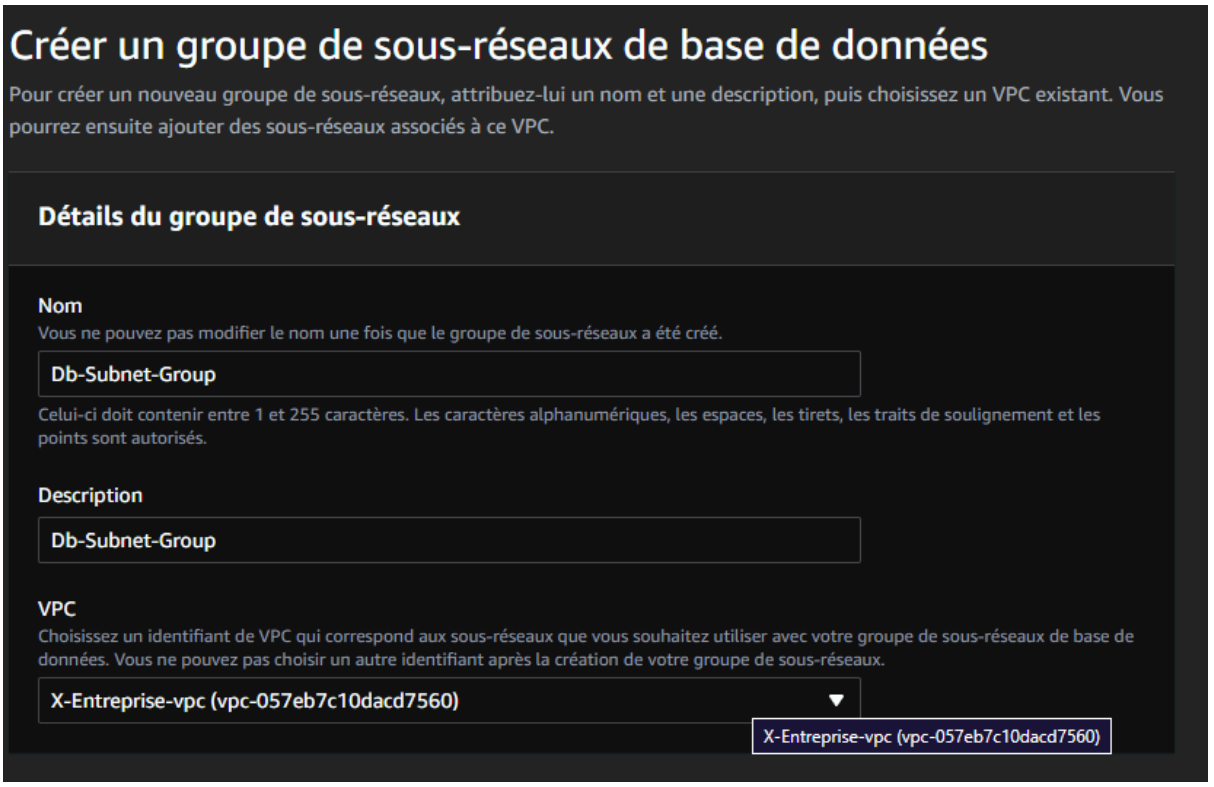



## 4. Database Creation

### 4.1. Creating PostgreSQL Database on AWS RDS

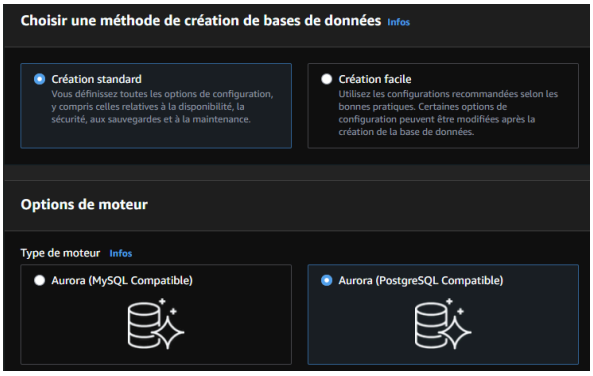
➔ Go to the RDS service.

➔ Create a database subnet group



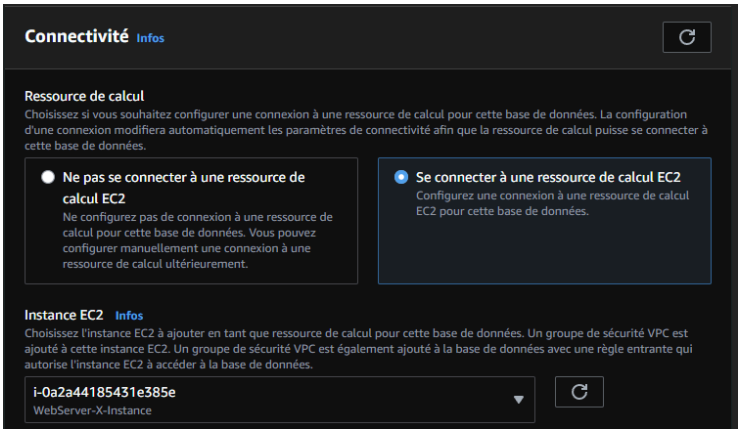

➔ Click on "Create database".

➔ Choose PostgreSQL as the database engine.

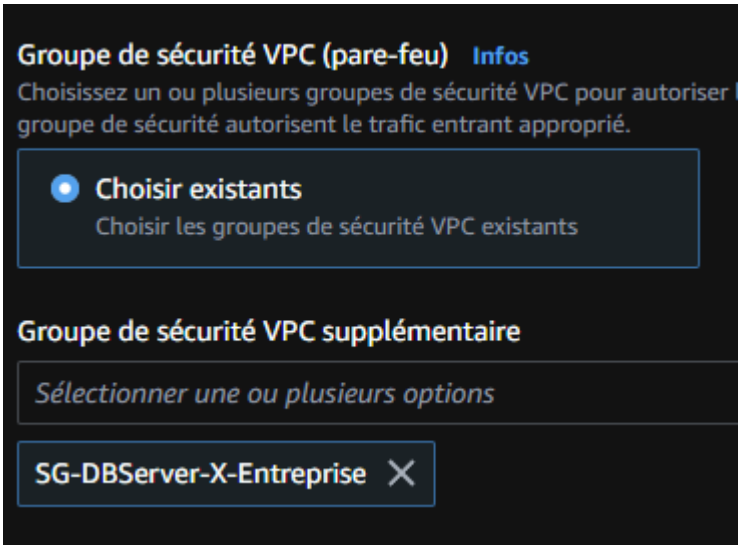



➔ Configure database instance details such as instance type

➔ Connection of instance EC2



➔ Select security group options to allow access to the database from your EC2 instance.

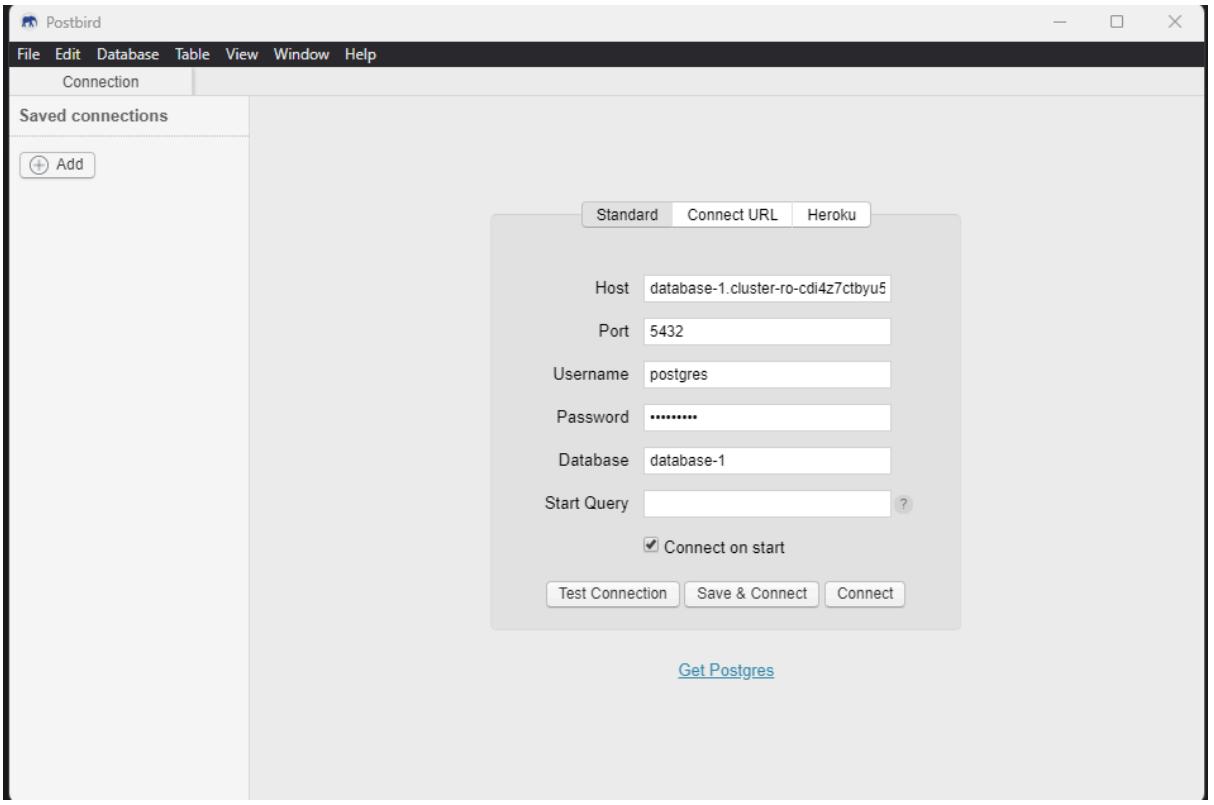



➔ Launch the database creation process.

## 4.2. Installation and Configuration of Postbird

➔ Install Postbird

➔ Once Postbird is installed, launch the application and connect to your RDS database using the appropriate credentials.



## 5. Database Connection

- ➔ An SSH connection is established with the EC2 web server.
- ➔ A PostgreSQL client is installed on the web server.
- ➔ A connection to the database is established from the command line to verify connectivity and functionality.