



PROJET E4 - ARCHITECTURE CLOUD AWS

Réalisé par :

- Matteo HEIB
- Mohamed CHAOUAY TISSIR
- Idriss QARQABI

Formation : 4AWSORC

Date : 04/02/2026

Année : 2025-2026

Contents

Introduction	3
1. Contexte du projet	3
2. Objectifs du projet	3
3. Méthodologie de travail	3
Partie 1 - Déploiement de l'environnement de développement MVP.....	4
1. Architecture globale.....	4
2. Configuration du VPC (Virtual Private Cloud).....	4
3. Configuration des Subnets	5
4. Configuration de l'Internet Gateway	6
5. Configuration de la Table de Routage.....	7
6. Configuration des groupes de sécurité (Security Groups)	8
7. Création de la paire de clés SSH et permissions	9
8. Base de données RDS MySQL	10
9. Déploiement de l'application E-commerce Django	11
10. Déploiement de WordPress	13
11. Système de sauvegarde.....	14
12. Conclusion de la Partie 1	16
Partie 2 - Déploiement de l'environnement de Cybersécurité.....	17
1. Architecture VPC Cybersécurité.....	17
2. Configuration du VPC Cybersécurité.....	18
3. Configuration des Subnets	18
4. Configuration du routage	19
5. Interconnexion VPC via VPC Peering	20
6. Déploiement de l'instance de cybersécurité	22
7. Configuration des Security Groups pour la communication inter-VPC	23
8. Déploiement des outils de cybersécurité et monitoring	26
Partie 3 : Évolution vers une Infrastructure Haute Disponibilité.....	30
1. Constat de l'existant (Le POC)	30
2. Nouvelle Infrastructure Théorique (Scalable & Redondante)	31
3. Justification des choix (Le "Pourquoi")	31

Introduction

1. Contexte du projet

Dans le cadre de notre intégration au sein d'une équipe DevSecOps d'une grande entreprise française, nous avons été missionnés pour déployer une infrastructure cloud complète sur Amazon Web Services (AWS) pour un nouveau client opérant dans le secteur de l'e-commerce.

2. Objectifs du projet

Le projet se décompose en trois phases principales :

Phase 1 - MVP (Minimum Viable Product) :

- Mise en place d'une application e-commerce avec passerelle de paiement Stripe
- Déploiement d'une stack WordPress pour une future migration
- Configuration de sauvegardes automatisées

Phase 2 - Isolation et sécurisation :

- Création d'environnements isolés pour trois équipes (Développement, IA, Cybersécurité)
- Mise en place de connexions sécurisées inter-environnements
- Déploiement d'une stack GitLab et d'outils de monitoring

Phase 3 - Optimisation et scalabilité :

- Refonte de l'architecture pour une meilleure scalabilité
- Optimisation des coûts et de la redondance
- Proposition d'une infrastructure évolutive

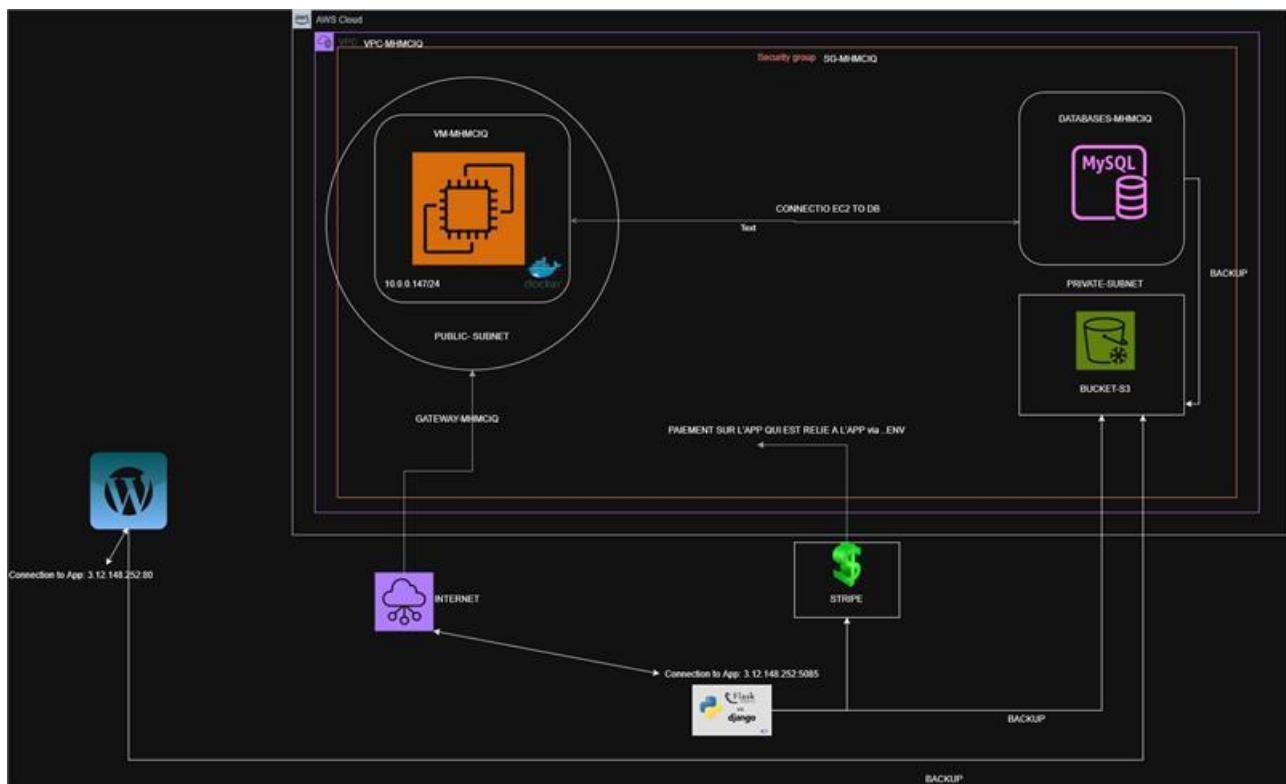
3. Méthodologie de travail

Notre équipe a adopté une approche collaborative avec :

- Utilisation de Git pour le versioning du code et de la documentation
- Déploiement via AWS CLI pour l'automatisation maximale
- Documentation continue de chaque étape du déploiement
- Respect des bonnes pratiques DevSecOps

Partie 1 - Déploiement de l'environnement de développement MVP

1. Architecture globale



Explication : Présentation de l'architecture globale déployée pour le MVP, incluant tous les composants et leurs relations.

2. Configuration du VPC (Virtual Private Cloud)

Qu'est-ce qu'un VPC ?

Un VPC (Virtual Private Cloud) est un réseau virtuel isolé dans le cloud AWS. Il permet de:

- Créer un réseau privé logiquement isolé
- Contrôler totalement l'environnement réseau (adressage IP, sous-réseaux, tables de routage)
- Sécuriser les ressources via des groupes de sécurité et des ACL réseau

Création du VPC

```
~ $ aws ec2 create-vpc --cidr-block 10.0.0.0/16 --tag-specifications 'ResourceType=vpc,Tags=[{Key=Name,Value=VPC-MHMCIQ}]'
{
  "Vpc": {
    "OwnerId": "336749236319",
    "InstanceTenancy": "default",
    "Ipv6CidrBlockAssociationSet": [],
    "CidrBlockAssociationSet": [
      {
        "AssociationId": "vpc-cidr-assoc-044ca53b0a16691aa",
        "CidrBlock": "10.0.0.0/16",
        "CidrBlockState": {
          "State": "associated"
        }
      }
    ],
    "IsDefault": false,
    "Tags": [
      {
        "Key": "Name",
        "Value": "VPC-MHMCIQ"
      }
    ],
    "VpcId": "vpc-07a27d73f88cd80aa",
    "State": "pending",
    "CidrBlock": "10.0.0.0/16",
    "DhcpOptionsId": "dopt-054151dea65bf735d"
  }
}
~ $
```

Résultat :

- **VPC ID** : vpc-07a27d73f88cd80aa
- **CIDR Block** : 10.0.0.0/16

Explication : Nous avons choisi la plage d'adresses 10.0.0.0/16 qui offre 65 536 adresses IP, largement suffisant pour notre infrastructure MVP et son évolution future. Cette plage privée permet d'isoler complètement nos ressources du réseau public.

3. Configuration des Subnets

Rôle des subnets :

Les subnets permettent de segmenter le VPC en sous-réseaux pour :

- Séparer les ressources publiques (accessibles depuis Internet) et privées
- Améliorer la sécurité et l'organisation
- Répartir les ressources sur plusieurs zones de disponibilité (haute disponibilité)

Création des subnets publics :

```
~ $ aws ec2 create-subnet --vpc-id $VPC_ID --cidr-block 10.0.1.0/24 --tag-specifications 'ResourceType=subnet,Tags=[{Key=Name,Value=Subnet-Public-MHMCIQ}]'
{
  "Subnet": {
    "AvailabilityZone": "use2-az1",
    "MapCustomerOwnedIpOnLaunch": false,
    "OwnerId": "336749236319",
    "AssignIpv6AddressOnCreation": false,
    "Ipv6CidrBlockAssociationSet": [],
    "Tags": [
      {
        "Key": "Name",
        "Value": "Subnet-Public-MHMCIQ"
      }
    ],
    "SubnetArn": "arn:aws:ec2:us-east-2:336749236319:subnet/subnet-0d2811ba0ecd84d5",
    "EnableDns64": false,
    "Ipv6Native": false,
    "PrivateDnsNameOptionsOnLaunch": {
      "HostedZoneType": "ip-name",
      "EnableResourceNameDnsARecord": false,
      "EnableResourceNameDnsAAAARecord": false
    },
    "SubnetId": "subnet-0d2811ba0ecd84d5",
    "Status": "available",
    "VpcId": "vpc-07a27d73f88cd80aa",
    "CidrBlock": "10.0.1.0/24",
    "AvailableIpAddressCount": 251,
    "AvailabilityZone": "us-east-2a",
    "DefaultForAz": false,
    "MapPublicIpOnLaunch": false
  }
}
~ $
```

Résultat :

- **Subnet ID** : subnet-0d2811ba0ecdb34d5
- **CIDR Block** : 10.0.1.0/24
- **Availability Zone** : us-east-2a

Explication : Ce subnet public hébergera nos instances EC2 pour l'application e-commerce et WordPress. Il sera connecté à Internet via l'Internet Gateway, permettant aux utilisateurs d'accéder aux applications web.

Création des subnets privés :

```
~ $ aws ec2 create-subnet --vpc-id $VPC_ID --cidr-block 10.0.2.0/24 --availability-zone us-east-2b --tag-specifications 'ResourceType=subnet,Tags=[{"Key=Name,Value=Subnet-Private-MHMCIQ}]'
{
  "Subnet": {
    "AvailabilityZoneId": "use2-az2",
    "MapCustomerOwnedIpOnLaunch": false,
    "OwnerId": "336749236319",
    "AssignIpv6AddressOnCreation": false,
    "Ipv4CidrBlockAssociationSet": [],
    "Tags": [
      {
        "Key": "Name",
        "Value": "Subnet-Private-MHMCIQ"
      }
    ],
    "SubnetArn": "arn:aws:ec2:us-east-2:336749236319:subnet/subnet-0dbc978ca92f84c9d",
    "EnableDns64": false,
    "Ipv6Native": false,
    "PrivateDnsNameOptionsOnLaunch": {
      "HostnameType": "ip-name",
      "EnableResourceNameDnsARecord": false,
      "EnableResourceNameDnsAAAARecord": false
    },
    "SubnetId": "subnet-0dbc978ca92f84c9d",
    "State": "available",
    "VpcId": "vpc-07a27d73f88cd80aa",
    "CidrBlock": "10.0.2.0/24",
    "AvailableIpAddressCount": 251,
    "AvailabilityZone": "us-east-2b",
    "DefaultForAz": false,
    "MapPublicIpOnLaunch": false
  }
}
~ $
```

Résultat :

- **Subnet ID** : subnet-0dbc978ca92f84c9d
- **CIDR Block** : 10.0.2.0/24
- **Availability Zone** : us-east-2b

Explication : Ce subnet privé, situé dans une zone de disponibilité différente (us-east-2b), hébergera notre base de données RDS. L'utilisation de deux AZ différentes est essentielle pour la configuration Multi-AZ de RDS, garantissant la haute disponibilité.

4. Configuration de l'Internet Gateway

Internet Gateway (IGW)

Rôle : Permet aux ressources du VPC de communiquer avec Internet.

```
~ $ aws ec2 create-internet-gateway --tag-specifications 'ResourceType=internet-gateway,Tags=[{"Key=Name,Value=IGW-MHMCIQ}]'
{
  "InternetGateway": {
    "Attachments": [],
    "InternetGatewayId": "igw-04303edd609c296ea",
    "OwnerId": "336749236319",
    "Tags": [
      {
        "Key": "Name",
        "Value": "IGW-MHMCIQ"
      }
    ]
  }
}
~ $
```

Résultat :

- IGW ID : igw-04303edd609c296ea

Attachement de l'IGW au VPC :

```
~ $ aws ec2 attach-internet-gateway --internet-gateway-id igw-04303edd609c296ea --vpc-id $VPC_ID
~ $
```

Explication : L'IGW est attaché au VPC pour permettre la communication bidirectionnelle entre les ressources du subnet public et Internet.

5. Configuration de la Table de Routage

```
~ $ aws ec2 create-route-table --vpc-id $VPC_ID --tag-specifications 'ResourceType=route-table,Tags=[{Key=Name,Value=RT-Public-MHMCIQ}]'
{
  "RouteTable": {
    "Associations": [],
    "PropagatingVgws": [],
    "RouteTableId": "rtb-0210ea72116197bf6",
    "Routes": [
      {
        "DestinationCidrBlock": "10.0.0.0/16",
        "GatewayId": "local",
        "Origin": "CreateRouteTable",
        "State": "active"
      }
    ],
    "Tags": [
      {
        "Key": "Name",
        "Value": "RT-Public-MHMCIQ"
      }
    ],
    "VpcId": "vpc-07a27d73f88cd80aa",
    "OwnerId": "36749236319"
  },
  "ClientToken": "8864324a-1910-4a55-ac0d-f6604863a906"
}
~ $
```

Résultat :

- Route Table ID : rtb-0210ea72116197bf6

Ajout d'une route vers Internet :

```
~ $ aws ec2 create-route --route-table-id rtb-0210ea72116197bf6 --destination-cidr-block 0.0.0.0/0 --gateway-id igw-04303edd609c296ea
{
  "Return": true
}
~ $
```

Association de la table au subnet public :

```
~ $ aws ec2 associate-route-table --route-table-id rtb-0210ea72116197bf6 --subnet-id subnet-0d2811ba0ecdb34d5
{
  "AssociationId": "rtbassoc-0addc0655368729a7",
  "AssociationState": {
    "State": "associated"
  }
}
~ $
```

Explication : La route 0.0.0.0/0 vers l'IGW permet à toutes les instances du subnet public d'accéder à Internet et d'être accessibles depuis Internet (selon les règles des Security Groups).

6. Configuration des groupes de sécurité (Security Groups)

Qu'est-ce qu'un Security Group ?

Les Security Groups agissent comme un firewall virtuel au niveau de l'instance pour contrôler le trafic entrant (inbound) et sortant (outbound). Ils fonctionnent selon le principe de liste blanche : tout est bloqué par défaut, seul le trafic explicitement autorisé peut passer.

Security Group pour l'application e-commerce et WordPress

```
~ $ aws ec2 create-security-group --group-name SG-Web-MHMCIQ --description "SG pour WordPress et Ecommerce" --vpc-id $VPC_ID
{
    "GroupId": "sg-04c1675f2b26cea7e",
    "SecurityGroupArn": "arn:aws:ec2:us-east-2:336749236319:security-group/sg-04c1675f2b26cea7e"
}
~ $
```

Résultat :

- Security Group ID** : sg-04c1675f2b26cea7e
- Nom** : SG-Web-MHMCIQ
- Description** : SG pour WordPress et Ecommerce

Autoriser SSH (port 22) :

```
]}
~ $ aws ec2 authorize-security-group-ingress --group-id sg-04c1675f2b26cea7e --protocol tcp --port 22 --cidr 0.0.0.0/0
{
    "Return": true,
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-07059eac66e4305a0",
            "GroupId": "sg-04c1675f2b26cea7e",
            "GroupOwnerId": "336749236319",
            "IsEgress": false,
            "IpProtocol": "tcp",
            "FromPort": 22,
            "ToPort": 22,
            "CidrIpv4": "0.0.0.0/0",
            "SecurityGroupRuleArn": "arn:aws:ec2:us-east-2:336749236319:security-group-rule/sgr-07059eac66e4305a0"
        }
    ]
}
~ $
```

Autoriser l'application Django (port 5085) :

The screenshot shows a CloudWatch log entry indicating that security group rules have been modified successfully:

Les règles de groupe de sécurité entrantes ont été modifiées avec succès sur le groupe de sécurité. (sg-04c1675f2b26cea7e | SG-Web-MHMCIQ)

Below the log, the AWS Management Console displays the security group configuration for 'sg-04c1675f2b26cea7e - SG-Web-MHMCIQ'. It shows the group name, owner, and VPC association. The 'Règles entrantes' tab is selected, showing two inbound rules:

Name	ID de règle de groupe	Type	Protocole	Plage de ports
-	sgr-07059eac66e4305a0	SSH	TCP	22
-	sgr-015d639687afa9e37	TCP personnalisé	TCP	5085

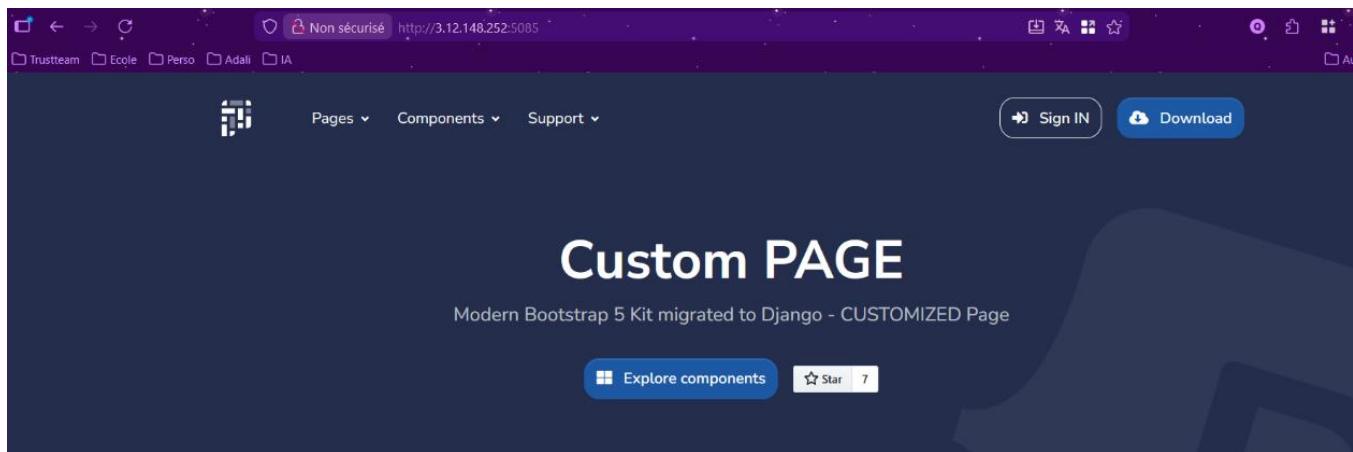


Tableau récapitulatif des règles :

Type	Protocole	Plage de ports	Source	Description
SSH	TCP	22	0.0.0.0/0	Accès administrateur
TCP personnalisé	TCP	5085	0.0.0.0/0	Application Django e-commerce

Explication :

- **Port 22 (SSH)** : Permet l'accès administrateur aux instances pour la configuration et la maintenance
- **Port 5085** : Port personnalisé choisi pour l'application Django afin de limiter les risques de scan automatisé (au lieu du port 80/443 standard). Ce choix améliore la sécurité par obscurité tout en permettant l'accès à l'application.

Note de sécurité : En production, il serait recommandé de :

- Restreindre l'accès SSH à des IP spécifiques (adresses IP de l'équipe)
- Utiliser un bastion host pour l'accès SSH
- Mettre en place un Load Balancer avec certificat SSL/TLS sur les ports 80/443

7. Crédation de la paire de clés SSH et permissions

```
{
~ $ aws ec2 create-key-pair --key-name Key-MHMCIQ --query 'KeyMaterial' --output text > Key-MHMCIQ.pem
~ $ chmod 400 Key-MHMCIQ.pem
~ $ }
```

Explication : La paire de clés SSH est essentielle pour se connecter de manière sécurisée aux instances EC2. La commande génère une clé privée qui est sauvegardée localement avec des permissions restrictives (400) pour garantir que seul le propriétaire peut la lire. Cette clé sera utilisée pour toutes les connexions SSH aux instances EC2.

8. Base de données RDS MySQL

Création du Subnet Group :

```
~ $ aws rds create-db-instance \
>   --db-instance-identifier db-mhmciq \
>   --db-instance-class db.t3.micro \
>   --engine mysql \
>   --allocated-storage 20 \
>   --db-subnet-group-name db-subnet-group-mhmciq \
>   --multi-az \
>   --master-username admin \
>   --master-user-password "Azerty123!" \
>   --tags Key=Name,Value=DB-RDS-MHMCIQ
{
  "DBInstance": {
    "DBInstanceIdentifier": "db-mhmciq",
    "DBInstanceClass": "db.t3.micro",
    "Engine": "mysql",
    "DBInstanceStatus": "creating",
    "MasterUsername": "admin",
    "AllocatedStorage": 20,
    "PreferredBackupWindow": "05:14-05:44",
    "BackupRetentionPeriod": 1,
    "DBSecurityGroups": [],
    "VpcSecurityGroups": [
      {
        "VpcSecurityGroupId": "sg-04894108eda4b3b4e",
        "Status": "active"
      }
    ],
    "DBParameterGroups": [
      {
        "DBParameterGroupName": "default.mysql8.4",
        "ParameterApplyStatus": "in-sync"
      }
    ]
  }
}
```

Explication : Le Subnet Group permet à RDS d'être déployé sur plusieurs zones de disponibilité pour la haute disponibilité.

Déploiement de la base de données :

```
mysql> CREATE DATABASE `DB-MHMCIQ`;
Query OK, 1 row affected (0.03 sec)

my@my-mysql:~$ mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| DB-MHMCIQ |
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
5 rows in set (0.00 sec)

mysql>
```

Configuration :

Engine : MySQL

Multi-AZ : Activé

Endpoint : db-mhmcjq-ctykc0acmtio.us-east-2.rds.amazonaws.com

Port : 3306

Explication : RDS MySQL Multi-AZ assure une haute disponibilité avec basculement automatique en cas de défaillance.

Connexion et initialisation :

Explication : Connexion réussie à la base de données RDS via le client MySQL pour l'initialisation.

9. Déploiement de l'application E-commerce Django

Installation de l'application :

Application choisie : Django Pixel (<https://github.com/app-generator/django-pixel>)

Explication : Application Django moderne avec interface utilisateur responsive, idéale pour un e-commerce.

Configuration de la base de données :

```
ubuntu@ip-10-0-1-208:~/app/ecommerce$ cat .env
DEBUG=True
DB_ENGINE=mysql
DB_NAME=DB-MHMCIQ
DB_HOST=db-mhmciq.ctykc0gcmtio.us-east-2.rds.amazonaws.com
DB_PORT=3306
DB_USERNAME=admin
DB_PASS=Azerty123!
```

```
# Database configuration from environment variables
DB_ENGINE = os.environ.get('DB_ENGINE', 'sqlite3')
DB_NAME = os.environ.get('DB_NAME', 'db.sqlite3')
DB_USERNAME = os.environ.get('DB_USERNAME', '')
DB_PASS = os.environ.get('DB_PASS', '')
DB_HOST = os.environ.get('DB_HOST', 'localhost')
DB_PORT = os.environ.get('DB_PORT', '3306')
```

Explication : Configuration du fichier settings.py pour connecter l'application Django à la base de données RDS MySQL.

Intégration de Stripe :

The left side shows terminal output for navigating to a 'payments' directory, listing files (total 40), and displaying the contents of 'urls.py'. The right side shows a browser window with a Stripe payment form and a footer for 'Pixel'.

```
ubuntu@ip-10-0-1-208:~/app/ecommerce$ cd payments/
ubuntu@ip-10-0-1-208:~/app/ecommerce/payments$ ls -la
total 40
drwxrwxr-x 4 ubuntu ubuntu 4096 Feb 4 12:35 .
drwxrwxr-x 10 ubuntu ubuntu 4096 Feb 4 12:22 ..
-rw-rw-r-- 1 ubuntu ubuntu 0 Feb 4 12:14 __init__.py
-rw-rw-r-- 1 ubuntu ubuntu 389 Feb 4 12:18 admin.py
-rw-rw-r-- 1 ubuntu ubuntu 148 Feb 4 12:14 apps.py
drwxrwxr-x 2 ubuntu ubuntu 4096 Feb 4 12:14 migrations
-rw-rw-r-- 1 ubuntu ubuntu 915 Feb 4 12:16 models.py
drwxrwxr-x 3 ubuntu ubuntu 4096 Feb 4 12:17 templates
-rw-rw-r-- 1 ubuntu ubuntu 60 Feb 4 12:14 tests.py
-rw-rw-r-- 1 ubuntu ubuntu 341 Feb 4 12:17 urls.py
-rw-rw-r-- 1 ubuntu ubuntu 2958 Feb 4 12:35 views.py
ubuntu@ip-10-0-1-208:~/app/ecommerce/payments$ cat urls.py
from django.urls import path
from . import views

app_name = 'payments'

urlpatterns = [
    path('checkout/', views.create_checkout_session, name='checkout'),
    path('success/', views.payment_success, name='success'),
    path('cancel/', views.payment_cancel, name='cancel'),
    path('webhook/', views.stripe_webhook, name='webhook'),
]
```

The screenshot shows a Stripe payment form for a test environment. It includes fields for email, payment method (card number, expiration, CVC), cardholder name, country, and a checkbox for saving payment info. A large blue 'Payer' button is at the bottom.

Environnement de test New... Environnement de test

Achat sur votre site
850 000,00 €

Coordonnées

E-mail
email@example.com

Moyen de paiement

Informations de la carte
1234 1234 1234 1234 VISA
MM / AA CVC

Nom du titulaire de la carte
Nom complet

Pays ou région
France

Enregistrer mes informations pour régler plus rapidement
Payez en toute sécurité chez Environnement de test New business et partout où Link est accepté.

Payer

The screenshot shows the Stripe test environment dashboard. The left sidebar includes sections for Environment de test (with New business selected), Accueil, Soldes, Transactions (selected), Clients, Catalogue de produits, Raccourcis, Connect, Profils, and Payments (with Analyses, Litiges, Radar, Payment Links, and Terminal). The main content area displays a summary of transactions: 2 Réussis, 0 Remboursés, 0 Contestés, 0 En échec, and 0 Non capturés. Below this is a table of transaction details:

Montant	Moyen de paiement	Description	Client	Date	Date du remboursement	Motif du refus de paie...
850000,00 €	VISA **** 4242	pi_3Sx5PjHdJTFc3Zje1CxvpIMU	mohamed@settat.fr	4 févr. à 12:42	—	—
20,00 €	VISA **** 4242	pi_3Sx5GnHdJTFc3Zje0s5Fvr3d	fremendy97@gmail.com	4 févr. à 12:33	—	—

Explication : Création d'une page de paiement fonctionnelle permettant aux utilisateurs de réaliser des transactions sécurisées via Stripe.

10. Déploiement de WordPress

The screenshot shows the WordPress dashboard. The left sidebar includes Accueil, Articles, Médias, Pages, Commentaires, Apparence, Extensions, Comptes, Outils, et RégLAGES. The main content area displays the "Bienvenue sur WordPress !" screen with the message "En savoir plus sur la version 6.9.1". It features three cards: "Créez des contenus riches avec les blocs et les compositions", "Personnalisez l'ensemble de votre site avec les thèmes basés sur des blocs", and "Modifiez l'apparence de votre site avec les styles globaux". At the bottom, there are sections for "État de santé du site" (Aucune information) and "Brouillon rapide".

Configuration :

- Port : 80 (HTTP)
- Base de données : RDS MySQL

Explication : Installation et configuration de WordPress pour une future migration du contenu client.

11. Système de sauvegarde

Création du bucket S3 :

```
UNKNOWN OPTIONS: 2
~ $ aws s3 mb s3://bucket-mhmciq --region us-east-2
make_bucket: bucket-mhmciq
~ $ █
```

Explication : Bucket S3 dédié au stockage sécurisé des sauvegardes de l'application et de la base de données.

Script de sauvegarde automatisé :

```
ubuntu@ip-10-0-1-208:~$ nano ~/backup_django.sh
ubuntu@ip-10-0-1-208:~$ chmod +x ~/backup_django.sh
ubuntu@ip-10-0-1-208:~$ █
```

```
GNU nano 7.2
#!/bin/bash
# Indique au système que c'est un script à exécuter avec Bash.

DATE=$(date +%Y-%m-%d)
# Crée une variable avec la date du jour (ex: 2026-02-04) pour nommer tes fichiers.

# --- SAUVEGARDE DE LA BASE DE DONNÉES ---
cp /home/ubuntu/app/e-commerce/db.sqlite3 ~/db_${DATE}.sqlite3
# Copie ton fichier de base de données vers ton dossier personnel avec la date.

# --- SAUVEGARDE DE L'APPLICATION ---
tar -czf ~/app_${DATE}.tar.gz -C /home/ubuntu/app/e-commerce --exclude='venv' .
# tar -czf : Compresse l'application en un fichier .tar.gz.
# -C /home/... : Se déplace dans le dossier avant de compresser.
# --exclude='venv' : Ignore le dossier 'venv' (trop lourd et facile à recréer).

# --- ENVOI VERS AWS S3 ---
aws s3 cp ~/db_${DATE}.sqlite3 s3://bucket-mhmciq/backups/
aws s3 cp ~/app_${DATE}.tar.gz s3://bucket-mhmciq/backups/
# Utilise l'AWS CLI pour envoyer tes deux fichiers vers ton bucket S3.

# --- NETTOYAGE ---
rm ~/db_${DATE}.sqlite3 ~/app_${DATE}.tar.gz
# Supprime les copies locales pour ne pas gaspiller l'espace disque du serveur.
```

Fonctionnement du script :

- Génération d'un nom de fichier avec la date du jour
- Copie de la base de données SQLite locale
- Compression de l'application (exclusion du dossier venv)
- Upload automatique vers S3
- Nettoyage des fichiers temporaires

Explication : Script Bash automatisant la sauvegarde quotidienne de l'application Django et de sa base de données vers S3.

```
ubuntu@ip-10-0-1-208:~$ aws configure
AWS Access Key ID [None]: AKIAU4Z6RSBPTE7TARWO
AWS Secret Access Key [None]: 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
Default region name [None]: us-east-2
Default output format [None]:
ubuntu@ip-10-0-1-208:~$ ~/backup_django.sh
upload: ./db_2026-02-04.sqlite3 to s3://bucket-mhmciq/backups/db_2026-02-04.sqlite3
upload: ./app_2026-02-04.tar.gz to s3://bucket-mhmciq/backups/app_2026-02-04.tar.gz
ubuntu@ip-10-0-1-208:~$
```

The screenshot shows the AWS S3 console interface. On the left, there's a sidebar with navigation links like 'Amazon S3', 'Compartiments', 'Gestion des accès et sécurité', 'Gestion du stockage et informations', and 'Paramètres du compte et de l'organisation'. The main area is titled 'backups/' under the 'Compartiments' section. It shows a table of objects with columns for Nom (Name), Type, Dernière modification (Last modified), Taille (Size), and Classe de stockage (Storage class). There are buttons for Actions, Copier l'URI S3, Copier l'URL, Télécharger (Download), Ouvrir (Open), Supprimer (Delete), and Charger (Upload).

Nom	Type	Dernière modification	Taille	Classe de stockage
app_2026-02-04.tar.gz	gz	04 Feb 2026 02:02:36 PM CET	18.0 Mo	Standard
db_2026-02-04.sqlite3	sqlite3	04 Feb 2026 02:02:35 PM CET	128.0 Ko	Standard

Explication : Vérification de la présence des sauvegardes dans le bucket S3 avec horodatage.

Sauvegarde WordPress

```
upload: ../../home/ubuntu/temp_backups/django_db_2026-02-04.sqlite3 to s3://bucket-mhmciq/backups/2026-02-04/django_db_2026-02-04.sqlite3
upload: ../../home/ubuntu/temp_backups/wp_db_2026-02-04.sql to s3://bucket-mhmciq/backups/2026-02-04/wp_db_2026-02-04.sql
upload: ../../home/ubuntu/temp_backups/wp_files_2026-02-04.tar.gz to s3://bucket-mhmciq/backups/2026-02-04/wp_files_2026-02-04.tar.gz
upload: ../../home/ubuntu/temp_backups/django_app_2026-02-04.tar.gz to s3://bucket-mhmciq/backups/2026-02-04/django_app_2026-02-04.tar.gz
ubuntu@ip-10-0-1-208:/var/www/html$
```

This screenshot shows the AWS S3 console with a different folder structure. The main area is titled '2026-02-04' under the 'backups/' compartment. It displays a table of objects with columns for Nom (Name), Type, Dernière modification (Last modified), Taille (Size), and Classe de stockage (Storage class). There are buttons for Actions, Copier l'URI S3, Copier l'URL, Télécharger (Download), Ouvrir (Open), Supprimer (Delete), and Charger (Upload).

Nom	Type	Dernière modification	Taille	Classe de stockage
django_app_2026-02-04.tar.gz	gz	04 Feb 2026 02:20:27 PM CET	18.0 Mo	Standard
django_db_2026-02-04.sqlite3	sqlite3	04 Feb 2026 02:20:27 PM CET	128.0 Ko	Standard
wp_db_2026-02-04.sql	sql	04 Feb 2026 02:20:27 PM CET	192.4 Ko	Standard
wp_files_2026-02-04.tar.gz	gz	04 Feb 2026 02:20:27 PM CET	27.0 Mo	Standard

Explication : Extension du système de sauvegarde pour inclure les fichiers et la base de données WordPress.

12. Conclusion de la Partie 1

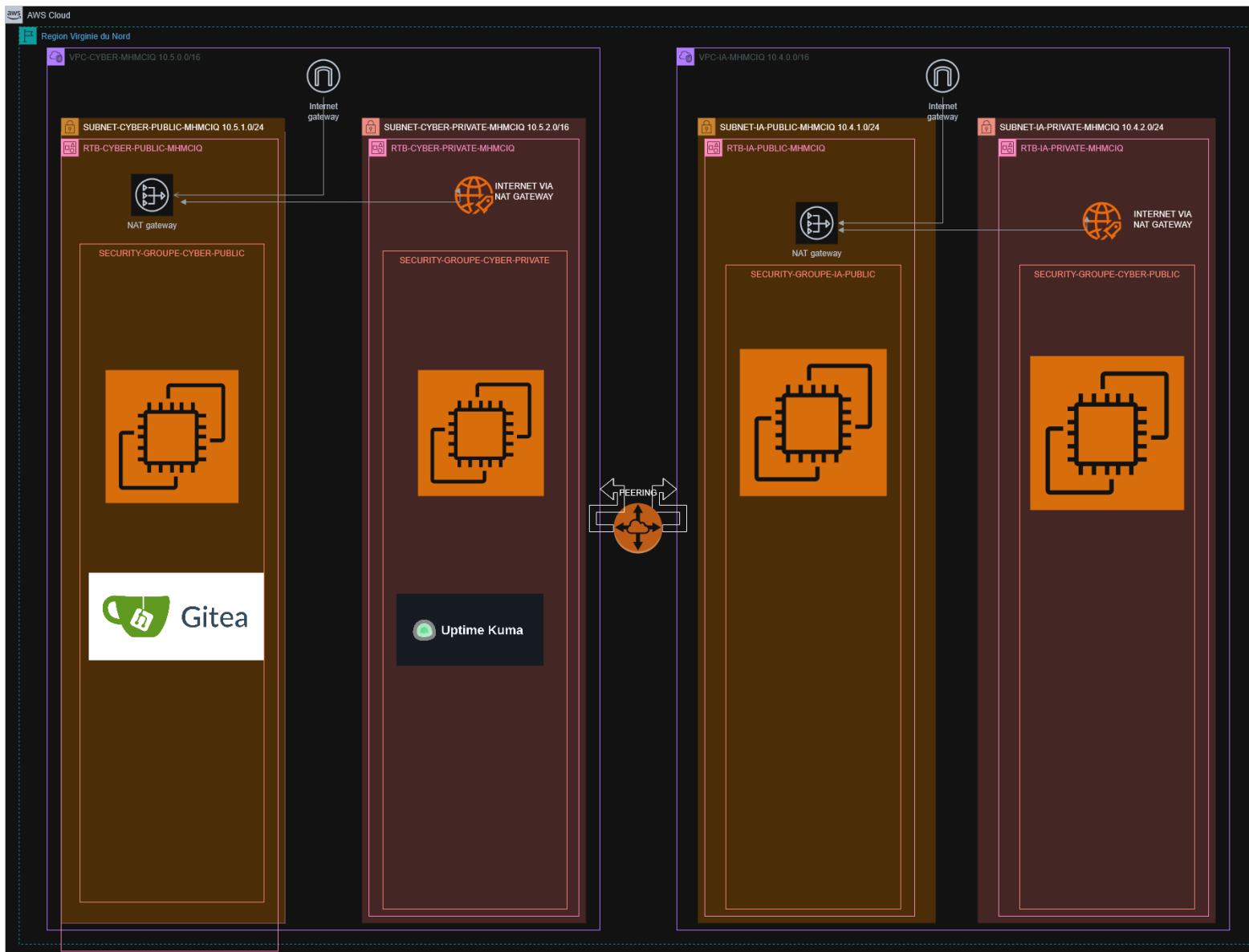
L'infrastructure MVP a été déployée avec succès en respectant les contraintes du client :

- **Application e-commerce** : Django avec paiement Stripe fonctionnel
- **WordPress** : Déployé et opérationnel
- **Base de données** : RDS MySQL Multi-AZ pour haute disponibilité
- **Sauvegardes** : Automatisées quotidiennement vers S3
- **Sécurité** : Security Groups et clés SSH configurés
- **Automatisation** : Déploiement via AWS CLI

L'architecture est fonctionnelle, sécurisée et prête pour la phase 2 du projet.

Partie 2 - Déploiement de l'environnement de Cybersécurité

1. Architecture VPC Cybersécurité



Explication : Architecture du VPC dédié à la cybersécurité avec segmentation réseau public/privé et interconnexion sécurisée avec le VPC IA via VPC Peering.

2. Configuration du VPC Cybersécurité

Création du VPC

The screenshot shows the AWS VPC Details page for the VPC named "vpc-002907cac531b310a / VPC-CYBER-MHMCIQ". Key details include:

- ID de VPC:** vpc-002907cac531b310a
- Résolution DNS:** Active
- CIDR IPv6 (groupe de bordure réseau):** -
- ID de contrôle de chiffrement:** -
- État:** Available
- Location:** default
- VPC par défaut:** Non
- Métriques d'utilisation d'adresses réseau:** Désactivé
- Mode de contrôle de chiffrement:** -
- Bloquer l'accès public:** Désactivé
- Jeu d'options DHCP:** dopt-0737764753af0e9a3
- CIDR IPv4:** 10.5.0.0/16
- Noms d'hôte DNS:** Désactivé
- Table de routage principale:** rtb-0871a1066d9ff9db4 / RTB-CYBER-MHMCIQ
- Groupe IPv6:** -
- ID du propriétaire:** 336749236319

Résultat :

- **VPC ID :** vpc-002907cac531b310a
- **Nom :** VPC-CYBER-MHMCIQ
- **CIDR Block :** 10.5.0.0/16

Explication : VPC dédié à l'environnement de cybersécurité avec une plage d'adresses distincte du VPC principal pour une isolation réseau optimale.

3. Configuration des Subnets

Création des Subnets

<input type="checkbox"/>	SUBNET-CYBER-PRIVATE-MHMCIQ	subnet-061046981cabb7e14	Available	vpc-002907cac531b310a VPC...	Désactivé	10.5.2.0/24
<input type="checkbox"/>	SUBNET-CYBER-PUBLIC-MHMCIQ	subnet-0d98bfc702b90e6a0	Available	vpc-002907cac531b310a VPC...	Désactivé	10.5.1.0/24

Subnet Public :

- **Nom :** SUBNET-CYBER-PUBLIC-MHMCIQ
- **Subnet ID :** subnet-0d98bfc702b90e6a0
- **CIDR :** 10.5.1.0/24
- **AZ :** us-east-1a

Subnet Privé :

- **Nom :** SUBNET-CYBER-PRIVATE-MHMCIQ
- **Subnet ID :** subnet-061046981cabb7e14
- **CIDR :** 10.5.2.0/24
- **AZ :** us-east-1a

Explication : Segmentation réseau permettant d'isoler les ressources publiques (outils de monitoring accessibles) des ressources privées (bases de données, outils d'analyse sensibles).

4. Configuration du routage

Création des Tables de Routage :

The screenshot shows the AWS VPC Tables of Routing interface. On the left, there's a sidebar with navigation links like 'Tableau de bord du VPC', 'AWS Global View', 'Cloud privé virtuel', 'Tables de routage' (which is selected), and 'Sécurité'. The main area displays 'Tables de routage (1/4)'. It lists four route tables, with 'RTB-CYBER-PUBLIC-MHMCIQ' selected. Below it, a detailed view for 'rtb-0871a1066d9ff9db4 / RTB-CYBER-PUBLIC-MHMCIQ' is shown, with tabs for 'Routes' (selected), 'Détails', 'Associations de sous-réseau', 'Associations de périphérie', 'Propagation de routage', and 'Balises'. The 'Routes' tab shows two routes: one to 'igw-09b5a40b20ba4fb34' and another to 'local'.

Table de routage Public :

- Nom :** RTB-CYBER-PUBLIC-MHMCIQ
- Route Table ID :** rtb-0871a1066d9ff9db4

Table de routage Privée :

- Nom :** RTB-CYBER-PRIVATE-MHMCIQ
- Route Table ID :** rtb-0de7ade1be05bde85
- Explication :** Deux tables de routage distinctes pour gérer différemment le trafic des subnets public et privé.

Configuration de la route publique

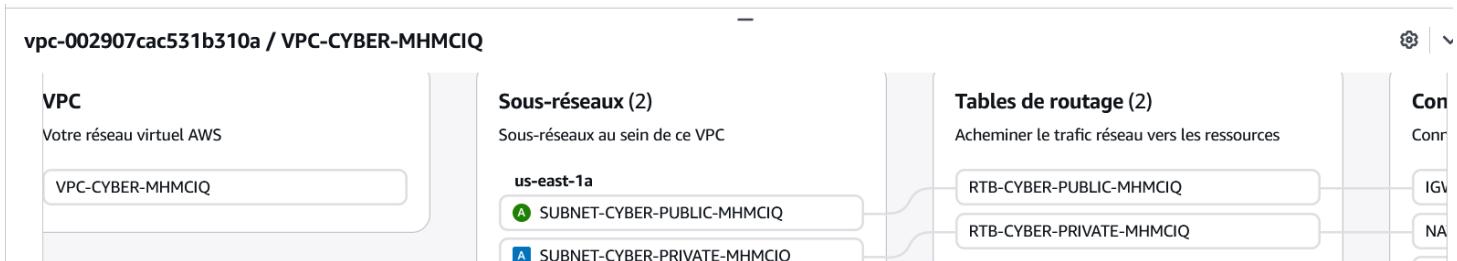
This screenshot shows the configuration of the RTB-CYBER-PRIVATE-MHMCIQ route table. The top part shows the list of route tables again, with 'RTB-CYBER-PRIVATE-MHMCIQ' selected. The bottom part shows its detailed configuration, with the 'Routes' tab selected. It lists two routes: one to 'nat-0f5ba70ffcaa2ca51' and another to 'local'.

Routes configurées :

- 10.5.0.0/16 → local
- 0.0.0.0/0 → Internet Gateway

Explication : Le subnet public accède directement à Internet via l'Internet Gateway pour permettre l'accès aux outils de cybersécurité.

Configuration de la route privée



Routes configurées :

- 10.5.0.0/16 → local
- 0.0.0.0/0 → NAT Gateway (située dans le subnet public)

Explication : Le subnet privé accède à Internet via une NAT Gateway pour les mises à jour de sécurité tout en restant inaccessible depuis l'extérieur.

5. Interconnexion VPC via VPC Peering

Création du VPC Peering CYBER → IA

Nom - facultatif
Créez une balise avec une clé « Name » et une valeur à spécifier.
PEERING-CYBER-TO-IA

Sélectionner un VPC local auquel s'appairer
ID du VPC (demandeur)
vpc-002907cac531b310a (VPC-CYBER-MHMCIQ)

CIDR de VPC pour vpc-002907cac531b310a (VPC-CYBER-MHMCIQ)
CIDR Status Motif du statut
10.5.0.0/16 Associated -

Sélectionner un autre VPC auquel s'appairer
Compte
 Mon compte
 Un autre compte
Région
 Cette région (us-east-1)
 Une autre région
ID de VPC (accepteur)
vpc-021acf691af32c592 (VPC-IA-MHMCIQ)

CIDR de VPC pour vpc-021acf691af32c592 (VPC-IA-MHMCIQ)
CIDR Status Motif du statut
10.4.0.0/16 Associated -

Résultat :

- Nom : PEERING-CYBER-TO-IA
- Peering Connection ID : pcx-0d1d19121edcbbadd
- VPC Demandeur : vpc-002907cac531b310a (VPC-CYBER-MHMCIQ - 10.5.0.0/16)
- VPC Accepteur : vpc-031acf691cf32c692 (VPC-IA-MHMCIQ - 10.4.0.0/16)
- Statut : Associé

Explication : VPC Peering permettant une communication privée et sécurisée entre l'environnement de cybersécurité et l'environnement IA sans passer par Internet.

Mise à jour des routes - VPC CYBER

Destination	Cible	Statut	Propagée	Origine du routage
10.5.0.0/16	local	Actif	Non	CreateRouteTable
0.0.0.0/0	Passerelle Internet	Actif	Non	CreateRoute
10.4.0.0/16	Connexion d'appairage	Inactif	Non	CreateRoute

Ajouter une route

Annuler Aperçu Enregistrer les modifications

Route ajoutée dans RTB-CYBER-PUBLIC-MHMCIQ :

- Destination : 10.4.0.0/16
- Cible : pcx-0d1d19121edcbbadd (Connexion d'appairage)

Explication : Permet au subnet public du VPC Cyber de communiquer avec le VPC IA.

Destination	Cible	Statut	Propagée	Origine du routage
10.5.0.0/16	local	Actif	Non	CreateRouteTable
0.0.0.0/0	Passerelle NAT	Actif	Non	CreateRoute
10.4.0.0/16	Connexion d'appairage	Inactif	Non	CreateRoute

Ajouter une route Utiliser : « pcx-0d1d19121edcbbadd »
pcx-0d1d19121edcbbadd (PEERING-CYBER-TO-IA)

Annuler Aperçu Enregistrer les modifications

Route ajoutée dans RTB-CYBER-PRIVATE-MHMCIQ :

- Destination : 10.4.0.0/16
- Cible : pcx-0d1d19121edcbbadd (Connexion d'appairage)

Explication : Permet au subnet privé du VPC Cyber de communiquer avec le VPC IA pour l'analyse de données.

Mise à jour des routes - VPC IA

Destination	Cible	Statut	Propagée	Origine du routage
0.0.0.0/0	igw-092c3fffc0bf2c30a3	Actif	Non	Créer une routage
10.4.0.0/16	local	Actif	Non	Créer une table de routage
10.5.0.0/16	pcx-0d1d19121edcbbadd	Actif	Non	Créer une routage

Explication : Confirmation de l'établissement bidirectionnel du VPC Peering.

Destination	Cible	Statut	Propagée	Origine du routage
0.0.0.0/0	nat-0f06dc1adb160882	Actif	Non	Créer une routage
10.4.0.0/16	local	Actif	Non	Créer une table de routage
10.5.0.0/16	pcx-0d1d19121edcbbadd	Actif	Non	Créer une routage

Routes ajoutées dans les tables de routage du VPC IA :

- **Destination :** 10.5.0.0/16
- **Cible :** pcx-0d1d19121edcbbadd

Explication : Permet au VPC IA de répondre aux requêtes provenant du VPC Cyber, établissant une communication bidirectionnelle.

6. Déploiement de l'instance de cybersécurité

Création de la VM dans le subnet public

Instances (1/1) Informations	
Name	i-0308db379b185e661
ID d'instance	i-0308db379b185e661
État de l'instance	En cours d...
Type d'instance	t3.micro
Contrôle des statu...	Initialisation en cc
Statut d'alarme	Afficher les alarm
Zone de dispon...	us-east-1a

Résumé de l'instance Informations		
ID d'instance	Adresse IPv4 publique	Adresses IPv4 privées
i-0308db379b185e661	-	10.5.1.35
Adresse IPv6	État de l'instance	DNS public
-	En cours d'exécution	-
Type de nom d'hôte	Nom DNS de l'IP privé (IPv4 uniquement)	Adresses IP élastiques
Nom de l'adresse IP: ip-10-5-1-35.ec2.internal	ip-10-5-1-35.ec2.internal	
Réponse à un nom DNS de resource privée	Type d'instance	

Configuration de l'instance :

- Nom : VM1-CYBER-PUBLIC-MHMCIQ
- Instance ID : i-0308d0b379b185e661
- Type : t3.micro
- Subnet : SUBNET-CYBER-PUBLIC-MHMCIQ
- Adresse IP privée : 10.5.1.35
- Statut : En cours d'exécution

Explication : Instance EC2 déployée dans le subnet public pour héberger les outils de cybersécurité accessibles par l'équipe.

7. Configuration des Security Groups pour la communication inter-VPC

Communication Subnet Public Cyber → Subnet Privé Cyber

Modifier les règles entrantes Informations

Les règles entrantes contrôlent le trafic entrant qui est autorisé à atteindre l'instance.

ID de règle de groupe de sécurité	Type	Protocole	Plage de ports	Source	Description - facultatif
sg-0a3c97e013ca069a7	SSH	TCP	22	Personne...	10.5.0.0/16

[Ajouter une règle](#) [Supprimer](#)

[Annuler](#) [Aperçu des modifications](#) [Enregistrer les règles](#)

Security Group : sg-08fbc11bf2a6149c2 - SECURITY-GROUP-CYBER-PRIVATE

Règle d'entrée configurée :

- **Type :** SSH
- **Protocole :** TCP
- **Port :** 22
- **Source :** 10.5.0.0/16 (tout le VPC Cyber)

Explication : Cette règle permet aux instances du subnet public Cyber d'accéder en SSH aux instances du subnet privé Cyber pour l'administration et la maintenance, tout en maintenant l'isolation du subnet privé vis-à-vis d'Internet.

```
ubuntu@ip-10-5-1-137:~$ ssh -i KeySSH.pem ubuntu@10.5.2.238
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-1018-aws x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/pro

System information as of Wed Feb  4 15:46:48 UTC 2026

System load: 0.02      Temperature:          -273.1 C
Usage of /: 26.6% of 6.71GB  Processes:           115
Memory usage: 23%        Users logged in:       0
Swap usage:  0%          IPv4 address for ens5: 10.5.2.238

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Wed Feb  4 15:46:26 2026 from 10.5.1.137
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-10-5-2-238:~$
```

Résultat :

- Connexion SSH depuis 10.5.1.137 (subnet public) vers 10.5.2.238 (subnet privé)
- IP privée de destination : 10.5.2.238
- Système : Ubuntu 24.04.3 LTS

Explication : Test de connectivité validant la configuration du Security Group. L'instance du subnet privé est accessible uniquement depuis le subnet public du même VPC, garantissant une architecture sécurisée en couches.

Communication Subnet Privé Cyber → Subnet Public IA

```
ubuntu@ip-10-5-2-238:~$ ssh -i KeySSH.pem ubuntu@10.4.1.165
The authenticity of host '10.4.1.165 (10.4.1.165)' can't be established.
ED25519 key fingerprint is SHA256:o0UY4XCrDaBEMeeSuun4qqZThn8WlDQlhiu/OQRuLpo.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.4.1.165' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-1018-aws x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/pro

System information as of Wed Feb  4 15:53:18 UTC 2026

System load: 0.0      Temperature:          -273.1 C
Usage of /: 25.9% of 6.71GB  Processes:           111
Memory usage: 23%        Users logged in:       0
Swap usage:  0%          IPv4 address for ens5: 10.4.1.165

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*-/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-10-4-1-165:~$
```

Security Group : sg-0a076e74743a198d4 - SECURITY-GROUP-IA-PUBLIC

Règle d'entrée configurée :

- **Type** : SSH
- **Protocole** : TCP
- **Port** : 22
- **Source** : 10.5.2.0/24 (subnet privé Cyber)

Explication : Cette règle autorise les instances du subnet privé Cyber à communiquer avec les instances du subnet public IA via le VPC Peering, permettant aux outils de cybersécurité d'analyser les services IA.

The screenshot shows the AWS Management Console interface for modifying security group rules. The URL is [VPC > Groupes de sécurité > sg-0a076e74743a198d4 - SECURITY-GROUP-IA-PUBLIC > Modifier les règles entrantes](#). The page title is "Modifier les règles entrantes". A note says "Les règles entrantes contrôlent le trafic entrant qui est autorisé à atteindre l'instance." Below is a table with one row:

ID de règle de groupe de sécurité	Type	Informations	Protocole	Informations	Plage de ports	Source	Informations	Description - facultatif
sgr-0ad6ecf6f94c531a0	SSH		TCP		22	Person...		

Below the table, there's a search bar with "10.5.2.0/24" and a "Supprimer" button. At the bottom are "Annuler", "Aperçu des modifications", and "Enregistrer les règles" buttons.

Résultat :

- Connexion SSH depuis 10.5.2.238 (subnet privé Cyber) vers 10.4.1.165 (subnet public IA)
- IP privée de destination : 10.4.1.165
- Système : Ubuntu 24.04.3 LTS

Explication : Validation de la communication inter-VPC via VPC Peering. Les outils de monitoring du VPC Cyber peuvent désormais surveiller les services déployés dans le VPC IA sans exposition publique.

Communication Subnet Privé Cyber → Subnet Privé IA

```
ubuntu@ip-10-4-2-229:~$ ssh -i KeySSH.pem ubuntu@10.4.2.229
The authenticity of host '10.4.2.229 (10.4.2.229)' can't be established.
ED25519 key fingerprint is SHA256:ToolNar4RsNg0JUmDey/eu7djuPC0y6RDVLUe16TVMM.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.4.2.229' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-1018-aws x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/pro

System information as of Wed Feb  4 15:55:11 UTC 2026
System Load: 0.0 Temperature: -273.1 C
Usage of /: 26.1% of 6.71GB Processes: 112
Memory usage: 24%
Swap usage: 0% IPv4 address for ens5: 10.4.2.229

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-10-4-2-229:~$
```

Security Group : sg-094322fb3b735c890 - SECURITY-GROUP-IA-PRIVATE

Règle d'entrée configurée :

- Type : SSH
- Protocole : TCP
- Port : 22
- Source : 10.5.2.0/24 (subnet privé Cyber)

Explication : Cette règle permet aux outils de cybersécurité du subnet privé Cyber d'accéder aux ressources sensibles du subnet privé IA pour l'audit de sécurité et la surveillance des bases de données.

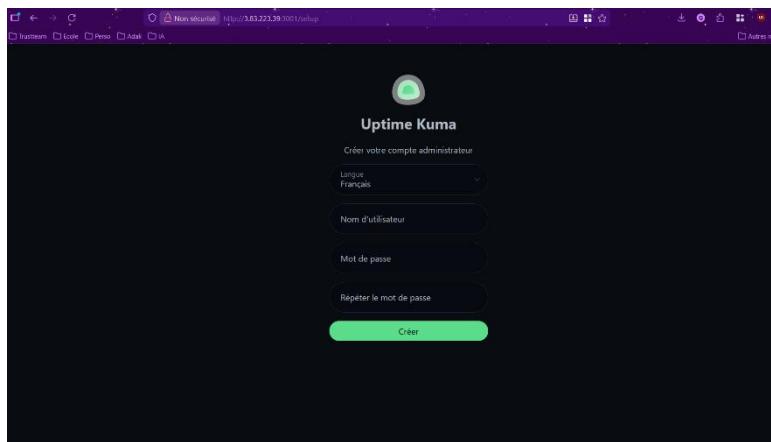
Résultat :

- Connexion SSH depuis 10.5.2.238 (subnet privé Cyber) vers 10.4.2.229 (subnet privé IA)
- IP privée de destination : 10.4.2.229
- Système : Ubuntu 24.04.3 LTS

Explication : Communication sécurisée établie entre les deux subnets privés via VPC Peering. Cette configuration permet l'audit de sécurité des ressources critiques du VPC IA tout en maintenant leur isolation complète d'Internet.

8. Déploiement des outils de cybersécurité et monitoring

Installation de Gitea (Gestion de versions)



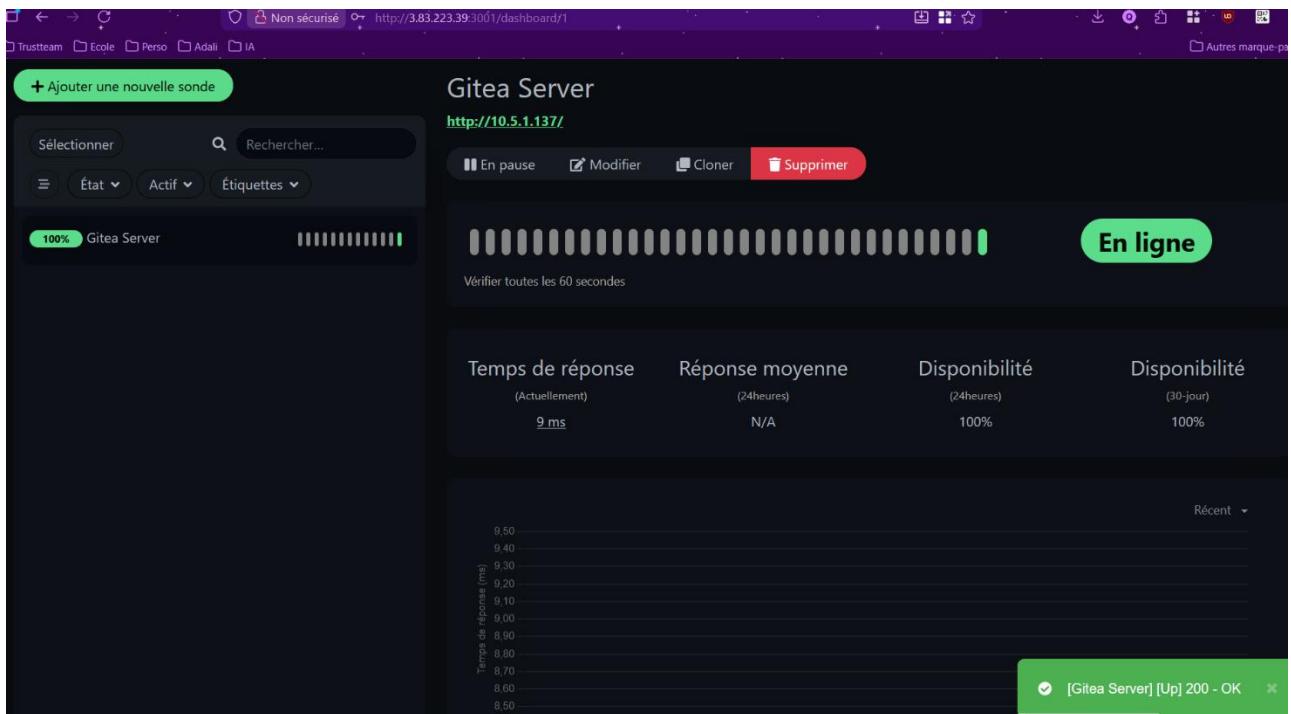
Configuration initiale :

- **URL d'accès** : <http://3.83.223.39:3001/setup>
- **Interface** : Uptime Kuma - Création du compte administrateur
- **Langue** : Français

Explication : Installation d'Uptime Kuma, un outil de monitoring léger et open-source, pour surveiller la disponibilité et les performances de nos applications. Cette interface permet de créer le compte administrateur qui gérera l'ensemble des sondes de surveillance.

Choix technique : Gitea a été choisi à la place de GitLab pour sa légèreté et sa faible consommation de ressources, tout en offrant les fonctionnalités essentielles de gestion de versions (Git). Ce choix a été validé par l'équipe pédagogique.

Configuration du framework de monitoring (Uptime Kuma)



Dashboard de monitoring :

- **Outil** : Uptime Kuma
- **URL** : <http://10.5.1.137/>
- **Service surveillé** : Gitea Server (<http://10.5.1.137/>)
- **Statut** : En ligne (barre verte à 100%)

Métriques affichées :

- **Temps de réponse** : 9 ms (Acheminement)
- **Réponse moyenne** : N/A (24 heures)
- **Disponibilité (24h)** : 100%
- **Disponibilité (30j)** : 100%

- **Intervalle de vérification** : Toutes les 60 secondes
- **Statut HTTP** : [Gitea Server] [Up] 200 - OK

Explication : Framework de monitoring opérationnel surveillant activement la disponibilité de Gitea. La sonde vérifie l'état du service toutes les 60 secondes et affiche en temps réel les métriques de performance. La barre verte et le statut "200 - OK" confirment que le service est pleinement fonctionnel.

Valeur ajoutée :

- Surveillance en temps réel de la disponibilité des applications
- Alertes automatiques en cas d'indisponibilité
- Historique des performances sur 24h et 30 jours
- Interface intuitive pour le suivi de l'état de santé de l'infrastructure

Test de déploiement avec Gitea

The screenshot shows the Gitea administration interface. At the top, the URL is `http://3.83.223.39/admin/TACOS`. The main navigation bar includes links for Tickets, Demandes d'ajout, Jalons, and Explorateur. Below this, the 'Code' tab is selected, showing the repository details for 'admin / TACOS'. A section titled 'Introduction rapide' provides instructions to clone the repository. Below it, a 'Création d'un nouveau dépôt en ligne de commande' section contains a terminal window with the following git commands:

```
touch README.md
git init
git checkout -b main
git add README.md
git commit -m "first commit"
git remote add origin http://3.83.223.39/admin/TACOS.git
git push -u origin main
```

Further down, a 'Soumission d'un dépôt existant par ligne de commande' section also contains a terminal window with similar git commands:

```
git remote add origin http://3.83.223.39/admin/TACOS.git
git push -u origin main
```

Configuration Gitea :

- URL d'accès : `http://3.83.223.39/admin/TACOS`
- Utilisateur : admin
- Dépôt de test : TACOS

Fonctionnalités testées :

- **Code** : Gestion des fichiers sources
- **Tickets** : Suivi des problèmes
- **Paquets** : Gestion des packages
- **Projets** : Organisation des tâches
- **Wiki** : Documentation

Commandes Git disponibles :

Clonage du dépôt :

```
git clone http://3.83.223.39/admin/TACOS.git
```

Création d'un nouveau dépôt en ligne de commande :

```
touch README.md
```

```
git init
```

```
git checkout -b main
```

```
git add README.md
```

```
git commit -m "first commit"
```

```
git remote add origin http://3.83.223.39/admin/TACOS.git
```

```
git push -u origin main
```

Soumission d'un dépôt existant :

```
git remote add origin http://3.83.223.39/admin/TACOS.git
```

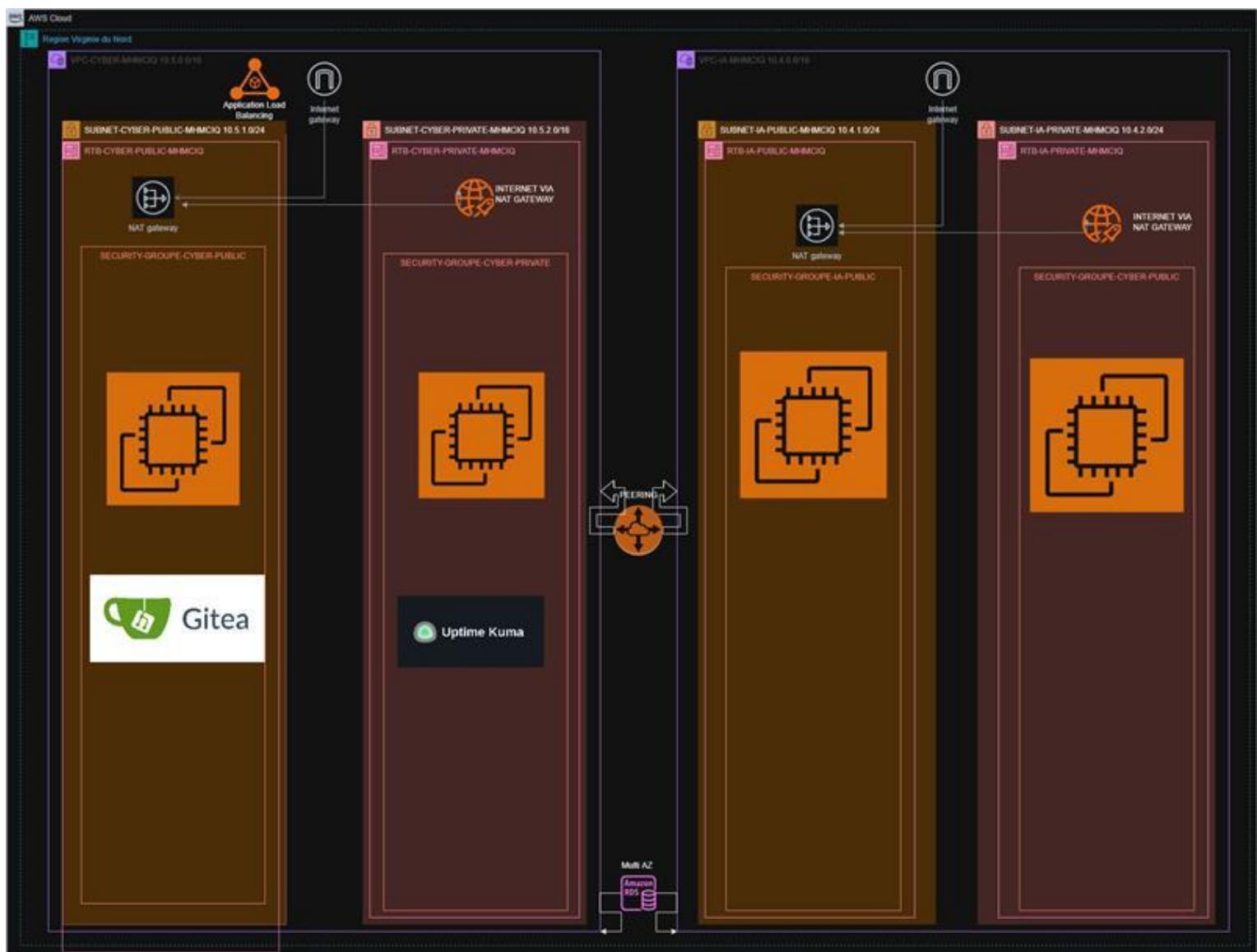
```
git push -u origin main
```

Explication : Gitea est pleinement opérationnel et prêt à héberger le code source de l'équipe DevSecOps. Le dépôt de test "TACOS" valide le bon fonctionnement de toutes les fonctionnalités Git (clone, push, pull, gestion des branches). L'équipe peut désormais versionner le code de l'infrastructure as code (scripts Terraform, Ansible, etc.).

Partie 3 : Évolution vers une Infrastructure Haute Disponibilité

1. Constat de l'existant (Le POC)

L'infrastructure actuelle a permis de valider le fonctionnement des services (Gitea, Uptime Kuma) et l'interconnexion des réseaux (VPC Peering). Cependant, elle repose sur des instances uniques (Single Point of Failure). Si une instance tombe, le service s'arrête.



2. Nouvelle Infrastructure Théorique (Scalable & Redondante)

Pour répondre aux besoins du client (budget moyen, gain de temps, fiabilité), nous proposons de passer d'un modèle **IaaS** (Gestion des serveurs) à un modèle **PaaS/Serverless** (Gestion des services).

Les trois piliers de l'évolution :

- **La Haute Disponibilité (Multi-AZ)** : Au lieu d'un seul centre de données, nous déployons les ressources sur **deux Zones de Disponibilité (AZ)**. Si l'une tombe, l'autre prend le relais instantanément.
- **Le "Serverless" avec AWS Fargate** : On remplace les VMs EC2 par des conteneurs gérés par **AWS ECS Fargate**. L'équipe ne perd plus de temps à mettre à jour l'OS ou à gérer le disque dur ; AWS s'occupe de la puissance de calcul nécessaire.
- **La Persistence des données** :
 - **Amazon RDS (PostgreSQL)** : Une base de données managée qui se réplique toute seule entre les zones.
 - **Amazon EFS** : Un stockage de fichiers partagé qui permet à plusieurs instances de Gitea de lire les mêmes données en même temps.

3. Justification des choix (Le "Pourquoi")

- **Gain de temps (Opérations)** : En utilisant des services "managés", l'équipe Cyber se concentre sur la sécurité et le code, pas sur la maintenance des serveurs.
- **Scalabilité** : Grâce à l'**Application Load Balancer (ALB)**, l'infrastructure peut ajouter ou supprimer des conteneurs Gitea automatiquement selon le nombre d'utilisateurs connectés.
- **Cohérence budgétaire** : Le modèle "Pay-as-you-go" d'AWS Fargate permet de ne payer que pour ce que les conteneurs consomment réellement, évitant ainsi de payer pour des serveurs surdimensionnés qui tournent à vide.