

2024

Cloud Computing - CAPSTONE PROJECT



Scénario :

Vous êtes embauché en tant qu'ingénieur cloud junior. La société X dispose d'un site web que vous pouvez trouver dans le dossier "sample-app".

Déploiement dans le Cloud

Vous êtes chargé de déployer le site web sur le Cloud AWS en suivant les instructions ci-dessous :

- Tout d'abord, vous devez dessiner un diagramme architectural de la solution et vous assurer d'expliquer vos choix en utilisant <https://app.diagrams.net/>.
- Votre solution doit être à l'intérieur de votre propre Cloud privé virtuel avec vos propres sous-réseaux et routages définis. Utilisez une passerelle Internet pour permettre l'accès Internet entrant et sortant.
- Le site web doit être hébergé à l'intérieur d'une instance EC2.

Création de la base de données

- Créez une base de données PostgreSQL en utilisant RDS d'AWS et utilisez <https://github.com/Paxa/postbird> pour visualiser les informations de la base de données. Veuillez créer quelques tables d'exemple et quelques enregistrements.
- SSH dans le serveur web ci-dessus et installez un client PostgreSQL. Assurez-vous de pouvoir vous connecter à la base de données depuis la ligne de commande.

Soumission

- Assurez-vous de soumettre votre URL GitHub public qui contient tout votre travail dans le projet CAPSTONE de Moodle.
- Assurez-vous que votre PDF est bien rédigé et décrit avec une table des matières et des sections, etc.

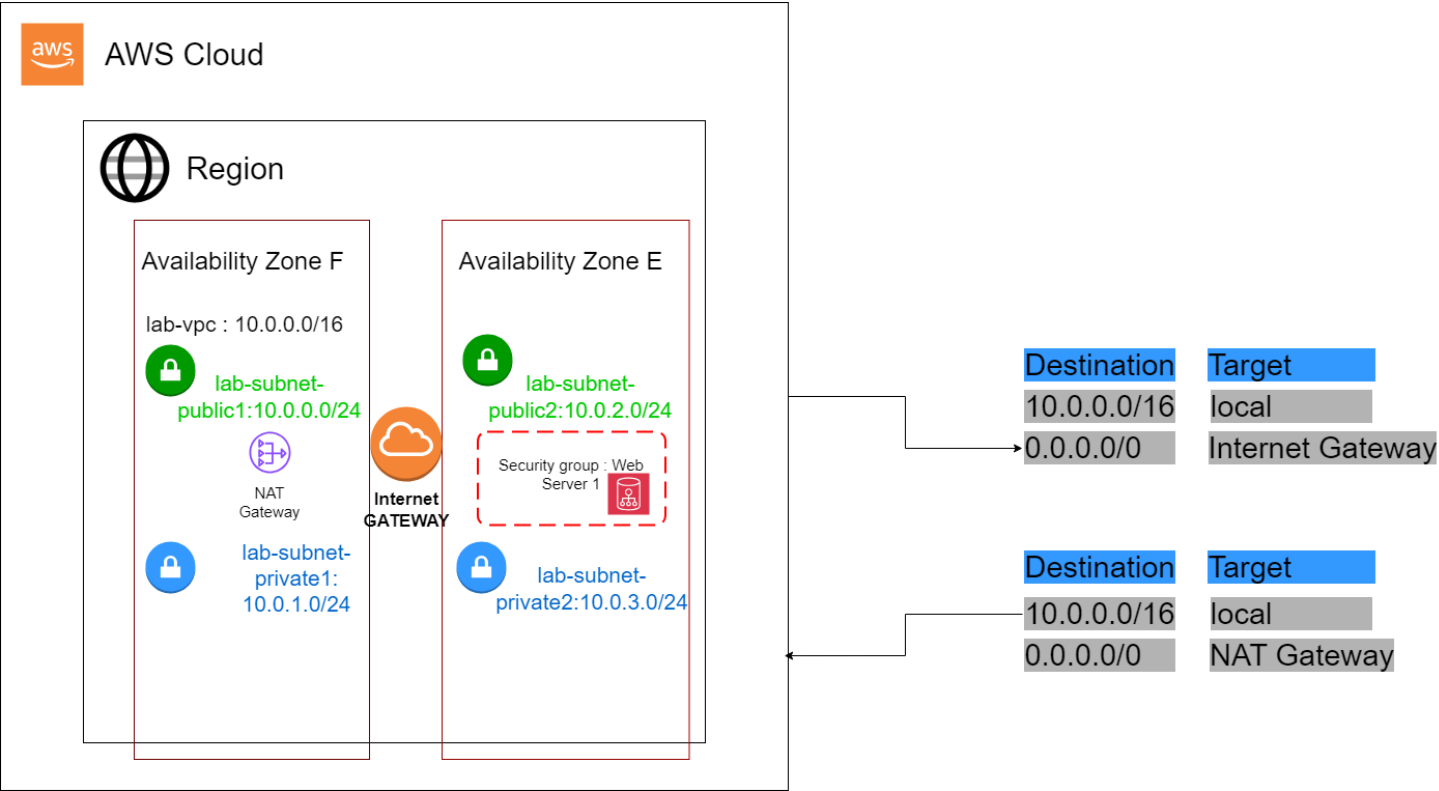
Références :

i. <https://aws.amazon.com/fr/vpc/>

ii. <https://aws.amazon.com/fr/ec2/>

Diagramme

Voici le diagramme architectural de la solution que j’ai choisies :



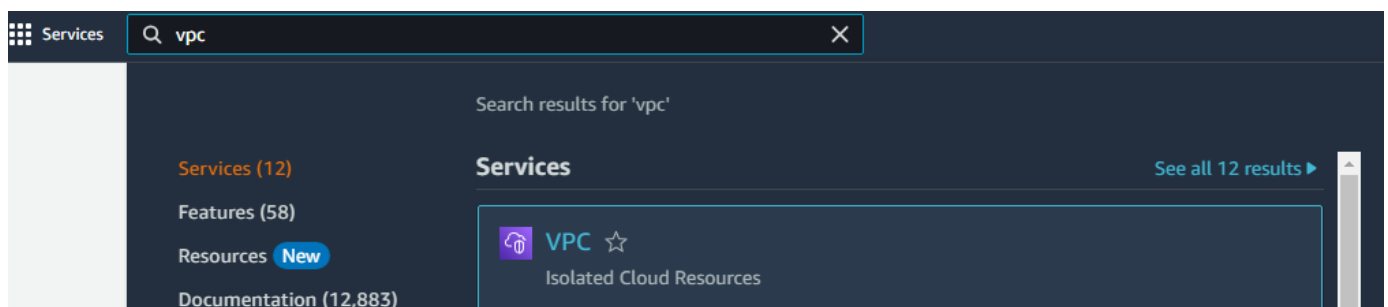
Dans cette architecture, j'ai choisi de déployer notre solution dans deux zones de disponibilité différentes pour garantir la haute disponibilité et la résilience de notre application. En répartissant nos ressources entre les zones de disponibilité F et E, nous nous assurons que notre application peut continuer à fonctionner sans interruption même en cas de défaillance dans l'une des zones. Cela minimise les risques de temps d'arrêt et assure une expérience utilisateur fiable.

En utilisant un Virtual Private Cloud (VPC), nous créons un environnement isolé et sécurisé pour notre application. Les sous-réseaux publics et privés nous permettent de séparer les composants accessibles depuis Internet de ceux qui ne le sont pas, renforçant ainsi la sécurité de notre système. De plus, l'utilisation d'une passerelle NAT dans le sous-réseau public assure une connectivité sécurisée pour les instances situées dans le sous-réseau privé, tout en masquant leurs adresses IP privées.

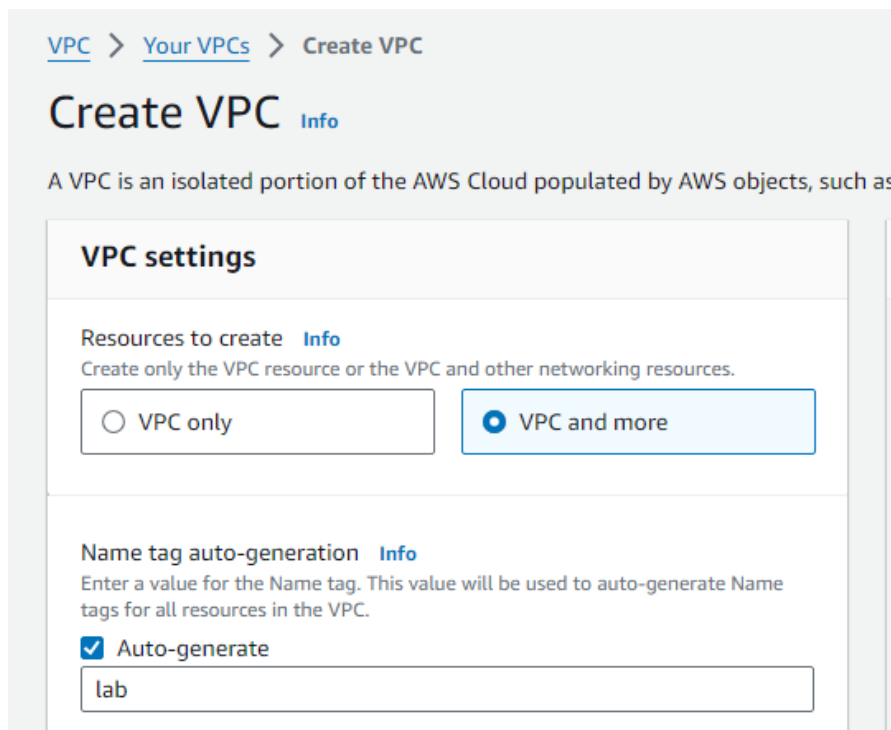
La mise en place d'une passerelle Internet au centre de notre architecture permet aux instances EC2 dans les sous-réseaux publics d'accéder à Internet et de répondre aux requêtes des utilisateurs externes. En associant des groupes de sécurité à nos sous-réseaux et instances, nous pouvons contrôler finement le trafic entrant et sortant, renforçant ainsi la sécurité de notre système. Cette architecture garantit une connectivité Internet fiable et sécurisée tout en préservant la confidentialité et l'intégrité de nos données.

Une fois le diagramme fait, on va mettre en place la solution.

Il faut tout d'abord aller dans l'environnement de test (sandbox) :



On appuie ensuite sur " Create VPC "



On met les différentes informations, VPC and more, lab en nom, le CIDR Block ipv4 et les zones de disponibilité ainsi que les options pour le public subnet et pour le private subnet.

Create VPC workflow

Wait for NAT Gateways to activate

72%

▼ Details

- ✓ Create VPC: [vpc-00a03d0da2b715f94](#)
- ✓ Enable DNS hostnames
- ✓ Enable DNS resolution
- ✓ Verifying VPC creation: [vpc-00a03d0da2b715f94](#)
- ✓ Create subnet: [subnet-06245de95852d6be1](#)
- ✓ Create subnet: [subnet-0592e035706c95411](#)
- ✓ Create internet gateway: [igw-0b84f90dfe1ee11e6](#)
- ✓ Attach internet gateway to the VPC
- ✓ Create route table: [rtb-01ffc87322ef70eec](#)
- ✓ Create route
- ✓ Associate route table
- ✓ Allocate elastic IP: [eipalloc-0c0cdc4100ebb7743](#)
- ✓ Create NAT gateway: [nat-09e9da88a4639e6e8](#)
- ⋮ Wait for NAT Gateways to activate
- ⌚ Create route table
- ⌚ Create route
- ⌚ Associate route table
- ⌚ Verifying route table creation

On va ensuite créer les sous réseaux :

Create subnet [Info](#)

VPC

VPC ID

Create subnets in this VPC.

vpc-00a03d0da2b715f94 (lab-vpc) ▼

Associated VPC CIDRs

IPv4 CIDRs

10.0.0.0/16

<input type="checkbox"/>	lab-subnet-public1-us-east-1a	subnet-06245de95852d6be1	Available	vpc-00a03d0da2b715f94 lab-...	10.0.0.0/24	-	250
<input type="checkbox"/>	lab-subnet-private1-us-east-1a	subnet-0592e035706c95411	Available	vpc-00a03d0da2b715f94 lab-...	10.0.1.0/24	-	251
<input type="checkbox"/>	lab-subnet-public2	subnet-04d0d7a8e6f444d78	Available	vpc-00a03d0da2b715f94 lab-...	10.0.2.0/24	-	251
<input type="checkbox"/>	lab-subnet-private2	subnet-010eb0df52860b865	Available	vpc-00a03d0da2b715f94 lab-...	10.0.3.0/24	-	251

On va ensuite faire la configuration des passerelles et des routes :

On va ajouter une passerelle Internet à notre VPC pour permettre l'accès Internet aux instances dans les sous-réseaux publics.

Et configurer les routes pour diriger le trafic vers la passerelle Internet pour les sous-réseaux publics, et vers la passerelle NAT pour les sous-réseaux privés.

Il faut ensuite, faire le déploiement de l'instance EC2 pour le site Web :

On va lancer une instance EC2 dans le sous-réseau public, en choisissant une AMI appropriée et en configurant les paramètres de sécurité (groupes de sécurité) pour permettre l'accès HTTP/HTTPS.

[EC2](#) > [Instances](#) > Launch an instance

Launch an instance [Info](#)

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags [Info](#)


Name

Web Server

[Add additional tags](#)

▼ Application and OS Images (Amazon Machine Image) [Info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

 Search our full catalog including 1000s of application and OS images

Recents

Quick Start



[Browse more AMIs](#)

Including AMIs from
AWS, Marketplace and
the Community

Amazon Machine Image (AMI)

Amazon Linux 2023 AMI

ami-051f8a213df8bc089 (64-bit (x86), uefi-preferred) / ami-05adadbbe8cf9fb48 (64-bit (Arm), uefi)

Free tier eligible ▼

On crée un groupe de sécurité :

Firewall (security groups) [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☒ Create security group

☐ Select existing security group

Security group name - *required*

Web Server security group

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and . _ - / () # , @ [] + = & ; ' ! \$ *

Description - *required* [Info](#)

Security group for my web server

Inbound Security Group Rules

▼ Security group rule 1 (TCP, 22, 0.0.0.0/0)

Remove

Instances (2)
[Info](#)

All states

Instance state

Actions

1

<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
<input type="checkbox"/>	Web Server	i-0fc714ade76c0d77b	Pending	t2.micro	-	View alarms	us-east-1a	-	-	-
<input type="checkbox"/>	Bastion Host	i-06911bb2a1eae6d33	Running	t2.micro	-	View alarms	us-east-1a	ec2-54-174-74-60.com...	54.174.74.60	-

Les instances s’afficheront en pending : il faut attendre que ce soit en Running.

[EC2](#)
[Security Groups](#)
[sg-05d3158a7b3e3932f - Web Server security group](#)
[Edit inbound rules](#)

Edit inbound rules
[Info](#)

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules
[Info](#)

Security group rule ID

Type

Protocol

Port range

Source

Description - optional

-

HTTP

TCP

80

Anywhere...

0.0.0.0/0

Delete

Add rule

Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Cancel

Preview changes

Save rules

[EC2](#)
[Security Groups](#)
[Create security group](#)

Create security group
[Info](#)

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

Basic details

Security group name

DB Security Group

Name cannot be edited after creation.

Description

Permit access from Web Security Group

VPC

vpc-00a03d0da2b715f94 (lab-vpc)

Inbound rules
[Info](#)

Type

Protocol

Port range

Source

Description - optional

MYSQL/Aurora

TCP

3306

Custom

sg-05d3158a7b3e3932f

sg-05d3158a7b3e3932f


Delete

Add rule

On va ensuite faire la création de la base de données RDS PostgreSQL :

Services

[See all 12 results ►](#)

 **RDS** ☆
Managed Relational Database Service

[RDS](#) > [Subnet groups](#) > Create DB subnet group

Create DB subnet group

To create a new subnet group, give it a name and a description, and choose an existing VPC. You will then be able to add subnets related to that VPC.

Subnet group details

Name

You won't be able to modify the name after your subnet group has been created.

Must contain from 1 to 255 characters. Alphanumeric characters, spaces, hyphens, underscores, and periods are allowed.

Description

VPC

Choose a VPC identifier that corresponds to the subnets you want to use for your DB subnet group. You won't be able to choose a different VPC identifier after your subnet group has been created.


Subnets

Choose the subnets that you want to add. The list includes the subnets in the selected Availability Zones.

Select subnets ▼

subnet-0592e035706c95411 (10.0.1.0/24) ✕

subnet-010eb0df52860b865 (10.0.3.0/24) ✕



 For Multi-AZ DB clusters, you must select 3 subnets in 3 different Availability Zones.

Subnets selected (2)		
Availability zone	Subnet ID	CIDR block
us-east-1a	subnet-0592e035706c95411	10.0.1.0/24
us-east-1b	subnet-010eb0df52860b865	10.0.3.0/24

Create database

Choose a database creation method [Info](#)



Standard create

You set all of the configuration options, including ones for availability, security, backups, and maintenance.



Easy create

Use recommended best-practice configurations. Some configuration options can be changed after the database is created.

Engine options

Engine type [Info](#)



Aurora (MySQL Compatible)



Aurora (PostgreSQL Compatible)



MySQL



MariaDB



On va accéder à la console RDS et créez une instance de base de données PostgreSQL dans le sous-réseau privé.

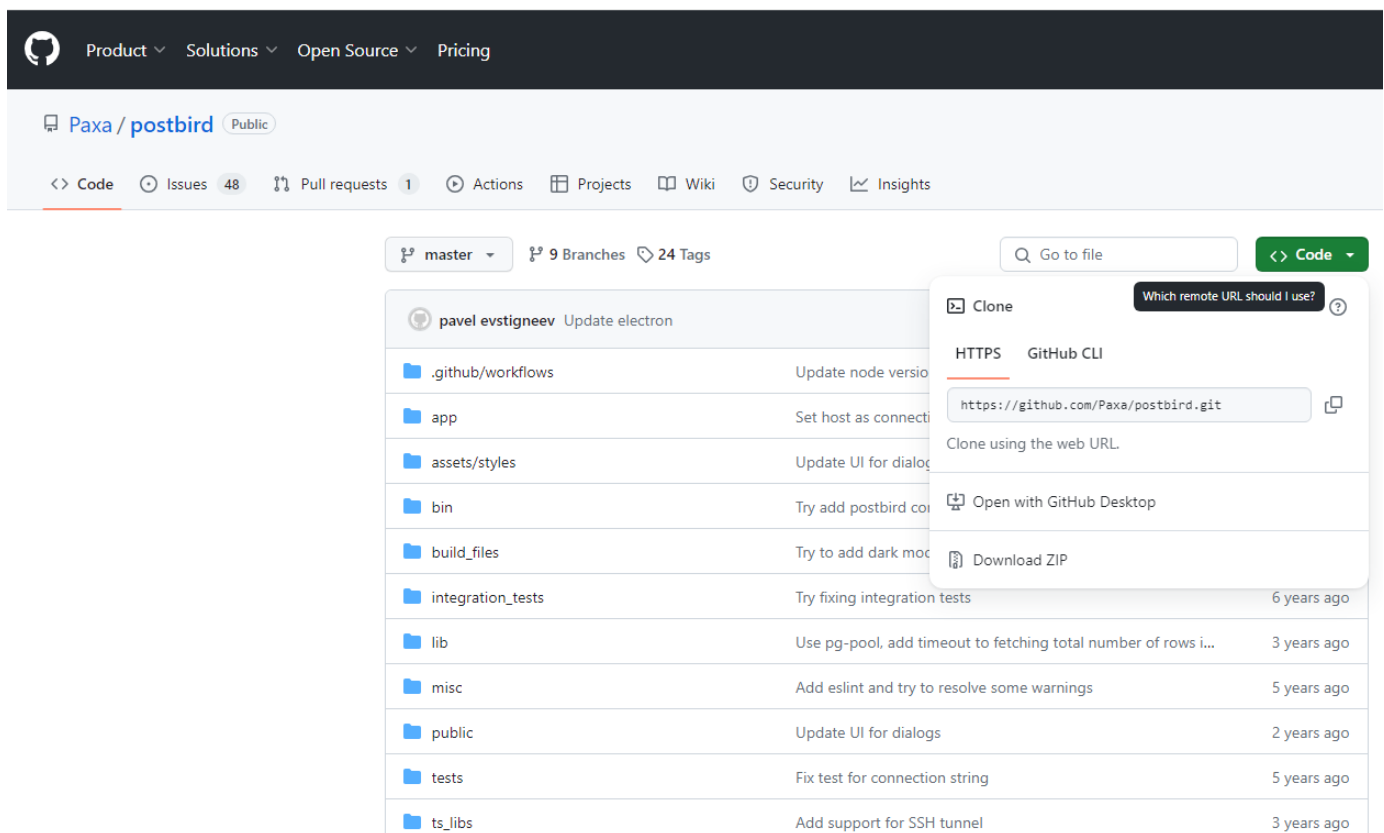
Ensuite on va configurer les paramètres de la base de données, y compris les groupes de sécurité pour autoriser l'accès depuis l'instance EC2.

On configure les règles de sécurité :

On va définir les règles de pare-feu dans les groupes de sécurité pour permettre le trafic approprié entre les différentes composantes de l'architecture, tout en restreignant l'accès non autorisé.

On utilise le lien <https://github.com/Paxa/postbird> pour visualiser les informations de la base de données et on crée des tables.

On peut télécharger directement le fichier en ZIP ou faire git clone + l'URL postbird



The screenshot displays the GitHub repository page for `Paxa/postbird`. The repository is public and has 48 issues, 1 pull request, and 9 branches. A file browser shows a directory structure with folders like `.github/workflows`, `app`, `assets/styles`, `bin`, `build_files`, `integration_tests`, `lib`, `misc`, `public`, `tests`, and `ts_libs`. A 'Clone' dropdown menu is open, showing options for HTTPS, GitHub CLI, Open with GitHub Desktop, and Download ZIP. The HTTPS option is selected, showing the URL `https://github.com/Paxa/postbird.git`.