



**Faculté des Sciences de Tunis**



**Université Tunis El Manar**

**MATIÈRE :** Cloud et Virtualisation

## **Déploiement d'une Application 3 Niveaux**

**ÉLABORÉ PAR :** Bel Haj Said Mahdi

Koutini Oumayma

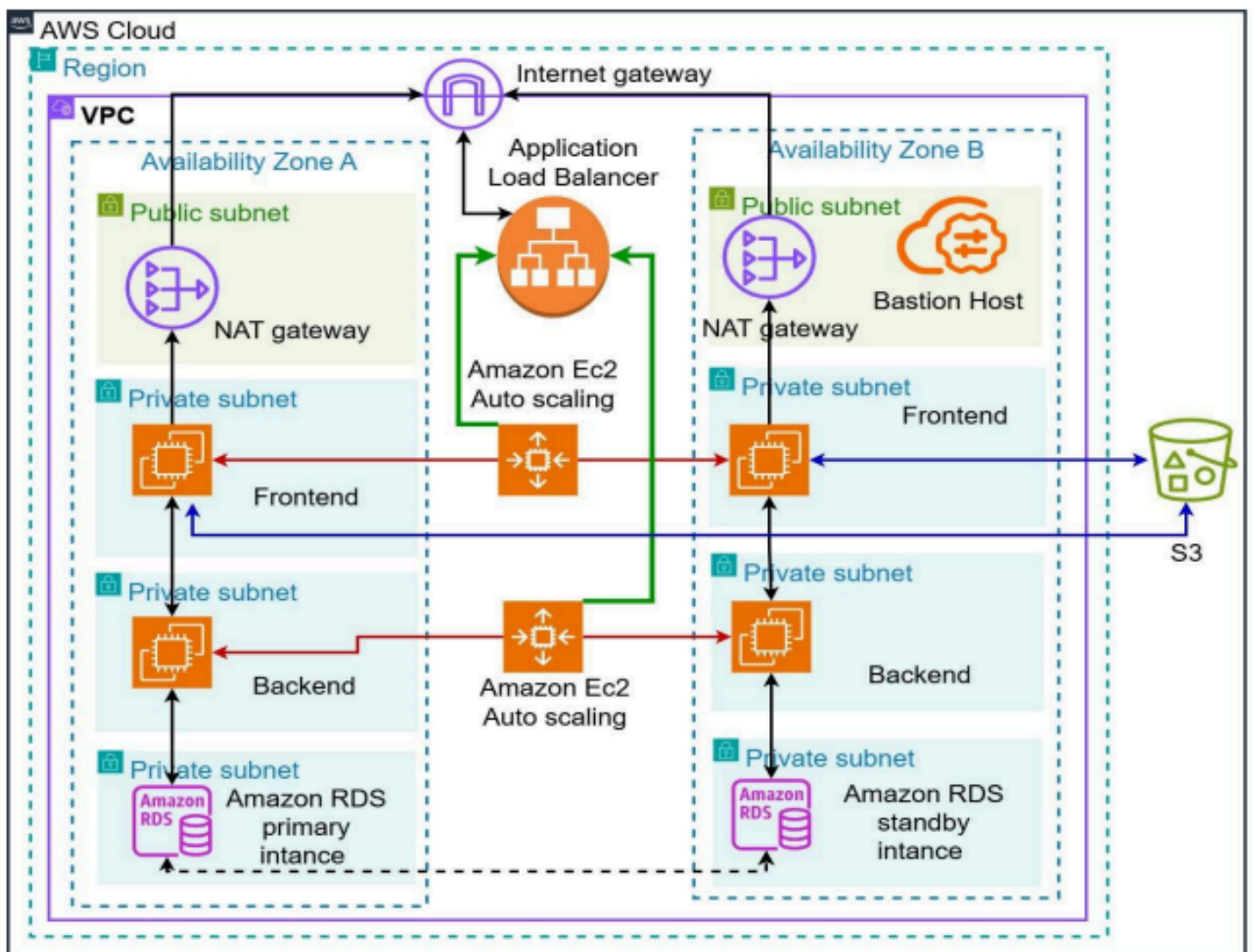
**ANNÉE UNIVERSITAIRE:** 2024/2025

**SECTION:** ICE4

## Objectif :

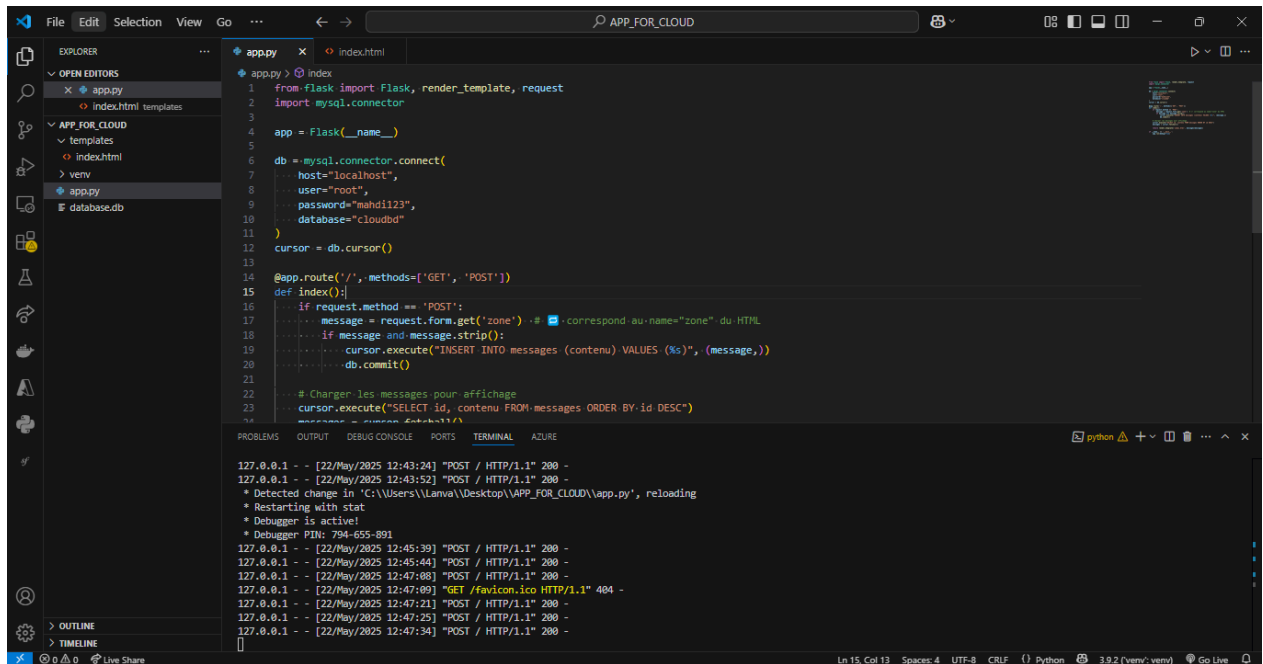
Créer une architecture cloud sécurisée et scalable sur AWS pour une application web et la déployer en utilisant les technologies et les ressources AWS

## Architecture :



## ETAPE 1 : Création de l'application Web

pour la création de l'application web on a utiliser html css pour le frontend et flask pour le backend et MySQL pour tester l'application localement

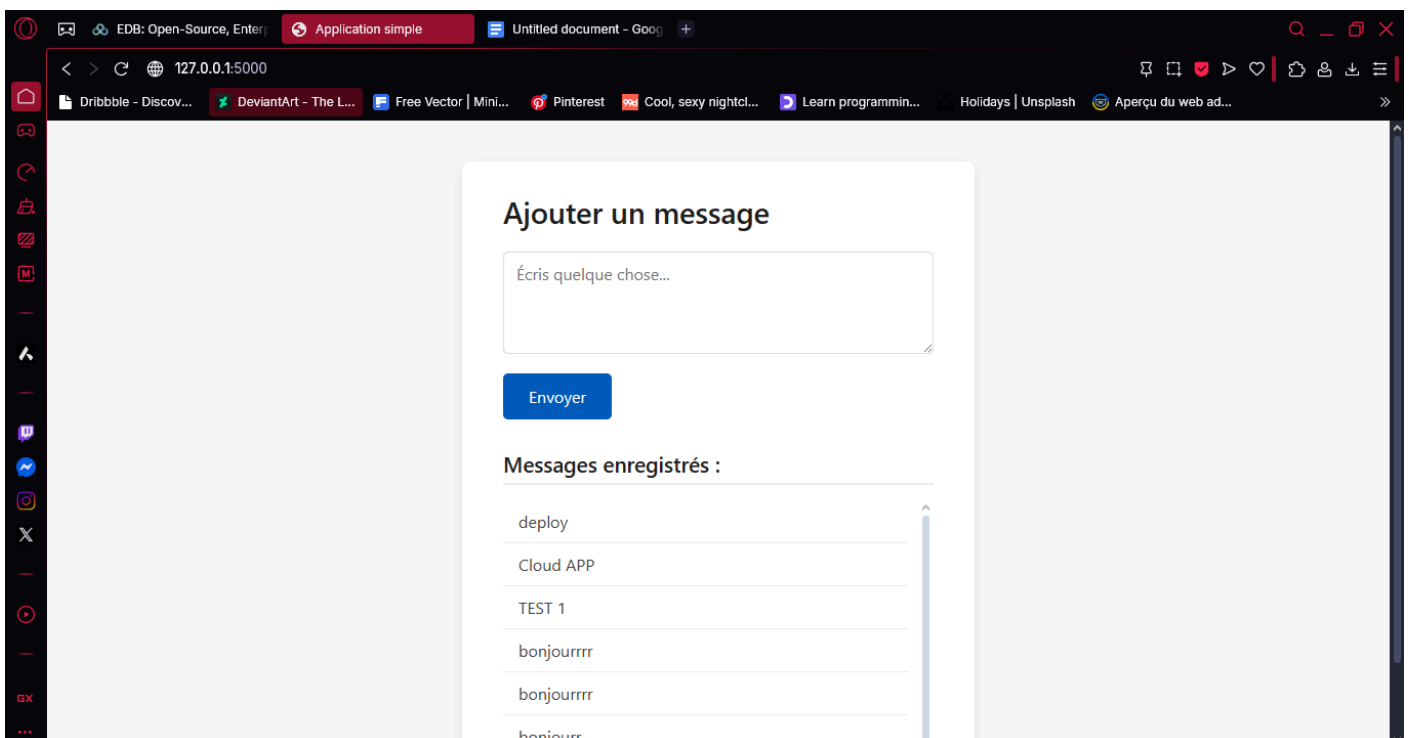


The screenshot shows a VS Code editor with a project named 'APP\_FOR\_CLOUD'. The Explorer sidebar shows the file structure: 'app.py', 'index.html', 'templates', 'venv', and 'database.db'. The main editor displays the 'app.py' file with the following code:

```
1 from flask import Flask, render_template, request
2 import mysql.connector
3
4 app = Flask(__name__)
5
6 db = mysql.connector.connect(
7     host="localhost",
8     user="root",
9     password="mahdi123",
10    database="cloudbd"
11)
12 cursor = db.cursor()
13
14 @app.route('/', methods=['GET', 'POST'])
15 def index():
16     if request.method == 'POST':
17         message = request.form.get('zone')
18         if message and message.strip():
19             cursor.execute("INSERT INTO messages (contenu) VALUES (%s)", (message,))
20             db.commit()
21
22     # Charger les messages pour affichage
23     cursor.execute("SELECT id, contenu FROM messages ORDER BY id DESC")
24     messages = cursor.fetchall()
```

The bottom panel shows the terminal output, indicating the application is running on 127.0.0.1:5000. It shows several POST requests and a GET request for the favicon, all returning 200 or 404 status codes.

## Test de l'application :



## ETAPE 2 : Créer un VPC et ses composants réseau :

- Créer un VPC **MyProjectVPC** avec une plage CIDR 10.0.0.0/16)

The screenshot shows the AWS VPC console interface. A green notification banner at the top states: "You successfully created vpc-0ed0d68cc89827b8e / MyProjectVPC". Below this, a table displays the VPC's configuration:

Property	Value
Enabled	default
Main network ACL	acl-0817444301e782da4
IPv6 CIDR (Network border group)	-
Default VPC	No
Network Address Usage metrics	Disabled
IPv4 CIDR	10.0.0.0/16
Route 53 Resolver DNS Firewall rule groups	Failed to load rule groups
IPv6 pool	-
Owner ID	604333932346

The "Resource map" section shows the VPC structure:

- VPC** (Show details): Your AWS virtual network. Contains "MyProjectVPC".
- Subnets (0)**: Subnets within this VPC.
- Route tables (1)**: Route network traffic to resources. Contains "rtb-0f54a64fd27556409".

- Créer 6 subnets :

The screenshot shows the AWS VPC console interface. A green notification banner at the top states: "You have successfully created 6 subnets: subnet-002f7a29ba1637087, subnet-015107345b9f49590, subnet-08b02cf27e55b4156, subnet-0ce34f3fbe275d9af, subnet-03521ee6465377455, subnet-0a4d7599d364ae000". Below this, the "Subnets (6)" section displays a table of the created subnets:

Name	Subnet ID	State	VPC
Private-BE-A	subnet-03521ee6465377455	Available	vpc-0ed0d68cc89827b8e   MyP...
Public-Subnet-A	subnet-002f7a29ba1637087	Available	vpc-0ed0d68cc89827b8e   MyP...
Public-Subnet-B	subnet-015107345b9f49590	Available	vpc-0ed0d68cc89827b8e   MyP...
Private-FE-A	subnet-08b02cf27e55b4156	Available	vpc-0ed0d68cc89827b8e   MyP...
Private-BE-B	subnet-0a4d7599d364ae000	Available	vpc-0ed0d68cc89827b8e   MyP...
Private-FE-B	subnet-0ce34f3fbe275d9af	Available	vpc-0ed0d68cc89827b8e   MyP...

**2 subnets publics** (un dans chaque zone de disponibilité A et B)

**4 subnets privés** : 2 pour le frontend/backend dans chaque zone

- **Création des routes tables :**

Name	Route table ID	Explicit subnet associations	Edge associations	Main
-	rtb-00c4591795c49ca24	-	-	Yes
Public-rtb	rtb-0126e9f53ebb83ba1	2 subnets	-	No
-	rtb-0f54a64fd27556409	-	-	Yes
Private1-rtb	rtb-0e66e877065fdec7	subnet-08b02cf27e55b4...	-	No
Private2-rtb	rtb-0f366f688a4c1b451	subnet-0ce34f3fbe275d9...	-	No
Private3-rtb	rtb-097e5c398929aea7f	subnet-03521ee6465377...	-	No
Private4-rtb	rtb-0d9b3861c5fa8b0c6	subnet-0a4d7599d364ae...	-	No

**1 route table** : pour les 2 subnet publique

**1 route table** pour chaque subnet privé  
donc on aura 5 route tables

- **On associe chaque route table les subnets et on vérifie avec le VPC overview**

**VPC dashboard**

**Your VPCs (1/2)**

Name	VPC ID	State	Block Public...	IPv4 CIDR	IPv6 CIDR	DHCP option set
-	vpc-0ff97c9972d690947	Available	Off	172.31.0.0/16	-	dopt-03c1fb0f7aaad1e6
MyProjectVPC	vpc-0ed0d68cc89827b8e	Available	Off	10.0.0.0/16	-	dopt-03c1fb0f7aaad1e6

**VPC overview**

**Subnets (6)**

- us-east-1a
  - Public-Subnet-A
  - Private-Subnet-A
  - Private-FE-A
- us-east-1b
  - Public-Subnet-B
  - Private-BE-B
  - Private-FE-B

**Route tables (6)**

- Public-rtb
- Private4-rtb
- Private2-rtb
- Private1-rtb
- rtb-0f54a64fd27556409
- Private3-rtb

- Créer une Internet Gateway MyIGW et l'attacher au VPC

**VPC dashboard** < Internet gateways > igw-0f7be3cf8294ce258

Internet gateway igw-0f7be3cf8294ce258 successfully attached to vpc-0ed0d68cc89827b8e

**igw-0f7be3cf8294ce258 / MyIGW** Actions

**Details** Info

Internet gateway ID: igw-0f7be3cf8294ce258 State: Attached VPC ID: vpc-0ed0d68cc89827b8e | MyProjectVPC Owner: 604333932346

**Tags** Manage tags

Key	Value
Name	MyIGW

- Réserver des adresses IP élastique pour les NAT Gateways

**VPC dashboard** < Elastic IP addresses (2) Info > Actions Allocate Elastic IP address

Find elastic IP addresses by attribute or tag

Name	Allocated IPv4 address	Type	Allocation ID	Reverse DNS record	Associated instance ID	Private IP
-	3.214.60.222	Public IP	eipalloc-017fef9702fa24da0	-	-	-
-	52.201.34.35	Public IP	eipalloc-0e22e1d0ba63ffead	-	-	-

Select an elastic IP address

View IP address usage and recommendations to release unused IPs with [Public IP insights](#)

- **CREER 2 NAT Gateways**  
1 NAT Gateway pour chaque availability zone (A-B)

The screenshot shows the AWS Management Console interface for NAT gateways. The left sidebar contains navigation links for VPC dashboard, Virtual private cloud, Security, and PrivateLink and Lattice. The main content area displays a table of NAT gateways with columns for Name, NAT gateway ID, Connectivity, State, State message, Primary public IP, Primary private IP, and Primary network interface. Two gateways are listed: NAT-GW-B (ID: nat-0c8a630c53574aff8) and NAT-GW-A (ID: nat-0f21e2594fcd65d3), both in an 'Available' state. Below the table, there is a section to 'Select a NAT gateway'.

Name	NAT gateway ID	Connectivity...	State	State message	Primary public IP	Primary private IP	Primary network...
NAT-GW-B	<a href="#">nat-0c8a630c53574aff8</a>	Public	Available	-	<a href="#">52.201.34.35</a>	10.0.2.202	<a href="#">eni-022b7f0160661...</a>
NAT-GW-A	<a href="#">nat-0f21e2594fcd65d3</a>	Public	Available	-	<a href="#">3.214.60.222</a>	10.0.1.71	<a href="#">eni-035a469799849...</a>

URL: <https://us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#NatGatewayDetails:natGatewayId=nat-0f21e2594fcd65d3>

- **Associer les NAT au routes**  
pour garantir l'accès à internet avec les subnets publics

The screenshot shows the AWS Management Console interface for a route table. The left sidebar contains navigation links for VPC dashboard, Virtual private cloud, Security, and PrivateLink and Lattice. The main content area displays the 'Details' tab for the route table 'rtb-0126e9f53ebb83ba1 / Public-rtb'. It shows the route table ID, VPC, and owner ID. Below the details, there is a section for 'Routes' with a table showing two routes: one for destination 0.0.0.0/0 pointing to target 'igw-0f7bc3cf8294cc258' (Active) and another for destination 10.0.0.0/16 pointing to target 'local' (Active).

Destination	Target	Status	Propagated
0.0.0.0/0	<a href="#">igw-0f7bc3cf8294cc258</a>	Active	No
10.0.0.0/16	local	Active	No

**Route tables (1/7) Info**

Find route tables by attribute or tag

Name	Route table ID	Explicit subnet associ...	Edge associations	Main	VPC	Owner ID
<input type="checkbox"/> Public-rtb	rtb-0126e9f53ebb83ba1	2 subnets	–	No	vpc-0ed0d68cc89827b8e   MyP...	604333932346
<input type="checkbox"/> –	rtb-0f54a64fd27556409	–	–	Yes	vpc-0ed0d68cc89827b8e   MyP...	604333932346
<input type="checkbox"/> Private1-rtb	rtb-0e66e877065fdecd7	subnet-08b02cf27e55b4...	–	No	vpc-0ed0d68cc89827b8e   MyP...	604333932346
<input type="checkbox"/> Private2-rtb	rtb-0f366f688a4c1b451	subnet-0ce34f3f6e275d9...	–	No	vpc-0ed0d68cc89827b8e   MyP...	604333932346
<input type="checkbox"/> Private3-rtb	rtb-097e5c398929aea7f	subnet-03521ee6465377...	–	No	vpc-0ed0d68cc89827b8e   MyP...	604333932346
<input checked="" type="checkbox"/> Private4-rtb	rtb-0d9b3861c5fa8b0c6	subnet-0a4d7599d364ae...	–	No	vpc-0ed0d68cc89827b8e   MyP...	604333932346

**rtb-0d9b3861c5fa8b0c6 / Private4-rtb**

Details | **Routes** | Subnet associations | Edge associations | Route propagation | Tags

**Routes (2)**

Filter routes

Destination	Target	Status	Propagated
0.0.0.0/0	nat-0c8a630c53574aff8	Active	No
10.0.0.0/16	local	Active	No

## ETAPE 3 : Créer les groupes de sécurité

- Création des paires des clés pour les groupes de sécurités

**Create key pair**

**Key pair name**  
Key pairs allow you to connect to your instance securely.  
MyKeyPair

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

**Key pair type**

☒ RSA  
RSA encrypted private and public key pair

☐ ED25519  
ED25519 encrypted private and public key pair

**Private key file format**

☐ .pem  
For use with OpenSSH

☒ .ppk  
For use with PuTTY

When prompted, store the private key in a secure and accessible location on your computer. You will need it later to connect to your instance. [Learn more](#)

Cancel Create key pair



- **Création du groupe de sécurité pour le load balancer SG-LB**

2 inbound rules : HTTP et HTTPS depuis Internet

EC2 Global View [\[?\]](#)

Filter by VPC: [▼](#)

▼ **Virtual private cloud**

Your VPCs

Subnets

Route tables

Internet gateways

Egress-only Internet gateways

Carrier gateways

DHCP option sets

Elastic IPs

Managed prefix lists

NAT gateways

Peering connections

Route servers [New](#)

▼ **Security**

Network ACLs

[Security groups](#)

**sg-0b0662db50570ec8c - SG-LB** [Actions](#) ▼

**Details**

Security group name <a href="#">SG-LB</a>	Security group ID <a href="#">sg-0b0662db50570ec8c</a>	Description <a href="#">SG-LB</a>	VPC ID <a href="#">vpc-0374bf378e4d2dcf5</a>
Owner <a href="#">245126994521</a>	Inbound rules count 2 Permission entries	Outbound rules count 1 Permission entry	

[Inbound rules](#) | [Outbound rules](#) | [Sharing - new](#) | [VPC associations - new](#) | [Tags](#)

**Inbound rules (2)** [Manage tags](#) [Edit inbound rules](#)

Q Search

<input type="checkbox"/>	Name	Security group rule ID	IP version	Type	Protocol	Port range	Source
<input type="checkbox"/>	-	sg-04e0444bd3072ae10	IPv4	HTTP	TCP	80	0.0.0.0/0
<input type="checkbox"/>	-	sg-067ce8ee5c2d44b37	IPv4	HTTPS	TCP	443	0.0.0.0/0

- **Création du groupe de sécurité pour les instances frontend SG-FE**

1 inbound rule : accepte trafic du SG-LB

VPC > [Security Groups](#) > [sg-03226629dfe31097c - SG-FE](#)

VPC dashboard <

EC2 Global View [\[?\]](#)

Filter by VPC: [▼](#)

▼ **Virtual private cloud**

Your VPCs

Subnets

Route tables

Internet gateways

Egress-only Internet gateways

Carrier gateways

DHCP option sets

Elastic IPs

Managed prefix lists

NAT gateways

Peering connections

Route servers [New](#)

**sg-03226629dfe31097c - SG-FE** [Actions](#) ▼

[Details](#)

Security group (sg-03226629dfe31097c [SG-FE]) was created successfully

**Details**

Security group name <a href="#">SG-FE</a>	Security group ID <a href="#">sg-03226629dfe31097c</a>	Description <a href="#">Security group for frontend instances</a>	VPC ID <a href="#">vpc-0374bf378e4d2dcf5</a>
Owner <a href="#">245126994521</a>	Inbound rules count 1 Permission entry	Outbound rules count 1 Permission entry	

[Inbound rules](#) | [Outbound rules](#) | [Sharing - new](#) | [VPC associations - new](#) | [Tags](#)

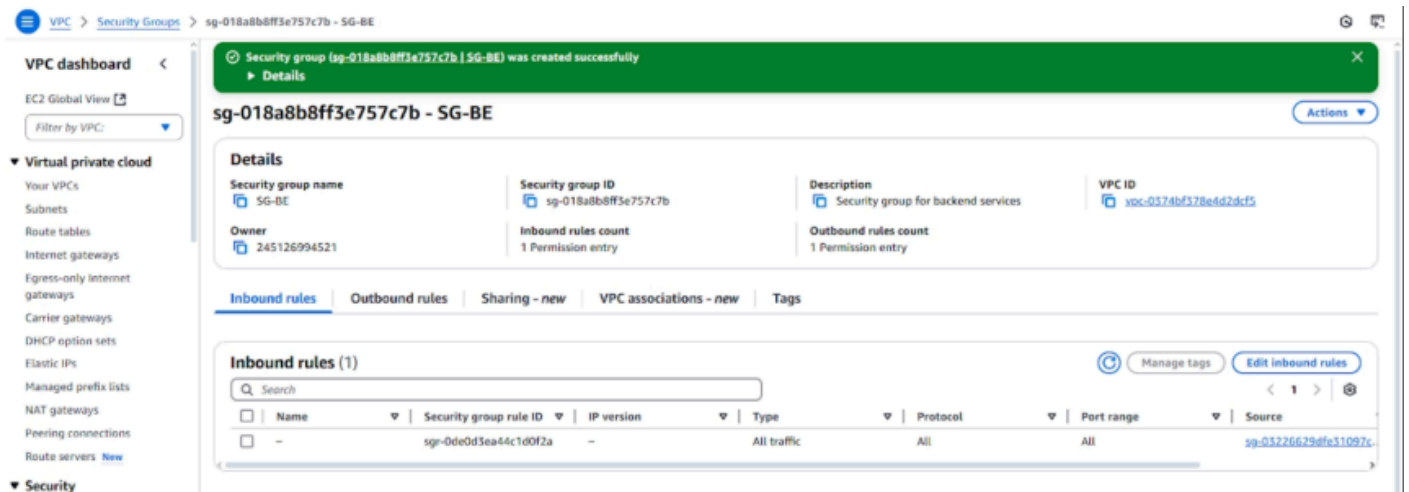
**Inbound rules (1)** [Manage tags](#) [Edit inbound rules](#)

Q Search

<input type="checkbox"/>	Name	Security group rule ID	IP version	Type	Protocol	Port range	Source
<input type="checkbox"/>	-	sg-03e6859bacc0180ef	-	All traffic	All	All	<a href="#">sg-02fc858a679de83a2</a>

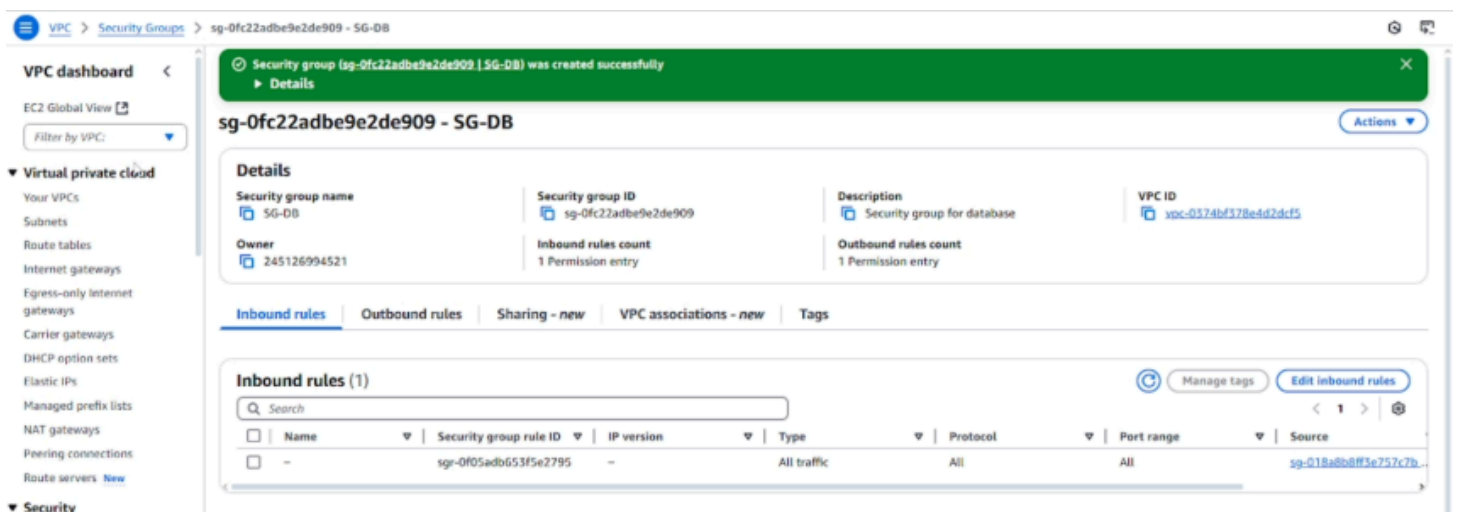
- **Création du groupe de sécurité pour les instances backend SG-BE**

2 inbound rule : accepte trafic du SG-FE et du SG-Bastion



- **Création du groupe de sécurité pour la base de données SG-DB**

1 inbound rule : accepte trafic du SG-BE



## • Création du groupe de sécurité Bastion host (accès SSH)

1 inbound rule : accès SSH depuis mon IP publique

Security group (sg-06062c0731c6a50f6 | SG-Bastion) was created successfully

Security Groups (1/7) Info

Name	Security group ID	Security group name	VPC ID	Description	Owner
SG-LB	sg-09f05bd999c79cee1	SG-LB	vpc-0ed0d68cc89827b8e	Acces HTTPHTTPS depuis Internet	604333932346
SG-BE	sg-0b65392e9bc34cb06	SG-BE	vpc-0ed0d68cc89827b8e	backend depuis frontend	604333932346
default	sg-028c50e895c18efaf	default	vpc-0ed0d68cc89827b8e	default VPC security group	604333932346
SG-Bastion	sg-06062c0731c6a50f6	SG-Bastion	vpc-0ed0d68cc89827b8e	SSH a la Bastion uniquement depuis mon IP publique	604333932346
SG-DB	sg-0c60f94889f561f7d	SG-DB	vpc-0ed0d68cc89827b8e	DB depuis backend uniquement	604333932346
default	sg-0ae4f504dd1cef2c2	default	vpc-0ff97c9972d690947	default VPC security group	604333932346
SG-FE	sg-0dfc365680b6834a6	SG-FE	vpc-0ed0d68cc89827b8e	frontend depuis le Load Balancer	604333932346

sg-09f05bd999c79cee1 - SG-LB

Details | Inbound rules | Outbound rules | Sharing - new | VPC associations - new | Tags

**Details**

Security group name SG-LB	Security group ID sg-09f05bd999c79cee1	Description Acces HTTPHTTPS depuis Internet	VPC ID vpc-0ed0d68cc89827b8e
Owner 604333932346	Inbound rules count 2 Permission entries	Outbound rules count 1 Permission entry	

## ETAPE 4 : Déployer les ressources EC2

### • Créer une instance EC2 frontend pour AZ

EC2 > Instances

Instances (1/2) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...
Frontend-A	i-00a028e67c25dc2fd	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1a	-	-
Frontend-B	i-0187283131c6aad81	Initializing	t2.micro	Initializing	View alarms +	us-east-1b	-	-

i-00a028e67c25dc2fd (Frontend-A)

Details | Status and alarms | Monitoring | Security | Networking | Storage | Tags

**Instance summary**

Instance ID i-00a028e67c25dc2fd	Public IPv4 address 10.0.3.148	Private IPv4 addresses 10.0.3.148
IPv6 address -	Instance state Running	Public DNS -
Hostname type IP name: ip-10-0-3-148.ec2.internal	Private IP DNS name (IPv4 only) ip-10-0-3-148.ec2.internal	Elastic IP addresses -
Answer private resource DNS name -	Instance type t2.micro	

- Création des instances EC2 backend et EC2 bastion

The screenshot displays the AWS Management Console for EC2 instances. The left sidebar shows the navigation menu with 'Instances' selected. The main content area shows a list of instances. The 'Bastion-Instance' (i-05d2b5f9cae950dfa) is selected, and its details are shown in the 'Instance summary' section. The instance is a t2.micro type, running in the us-east-1a availability zone. It has a public IP address of 3.222.205.229 and is currently in the 'Running' state. The instance is associated with the 't2.micro' VPC ID.

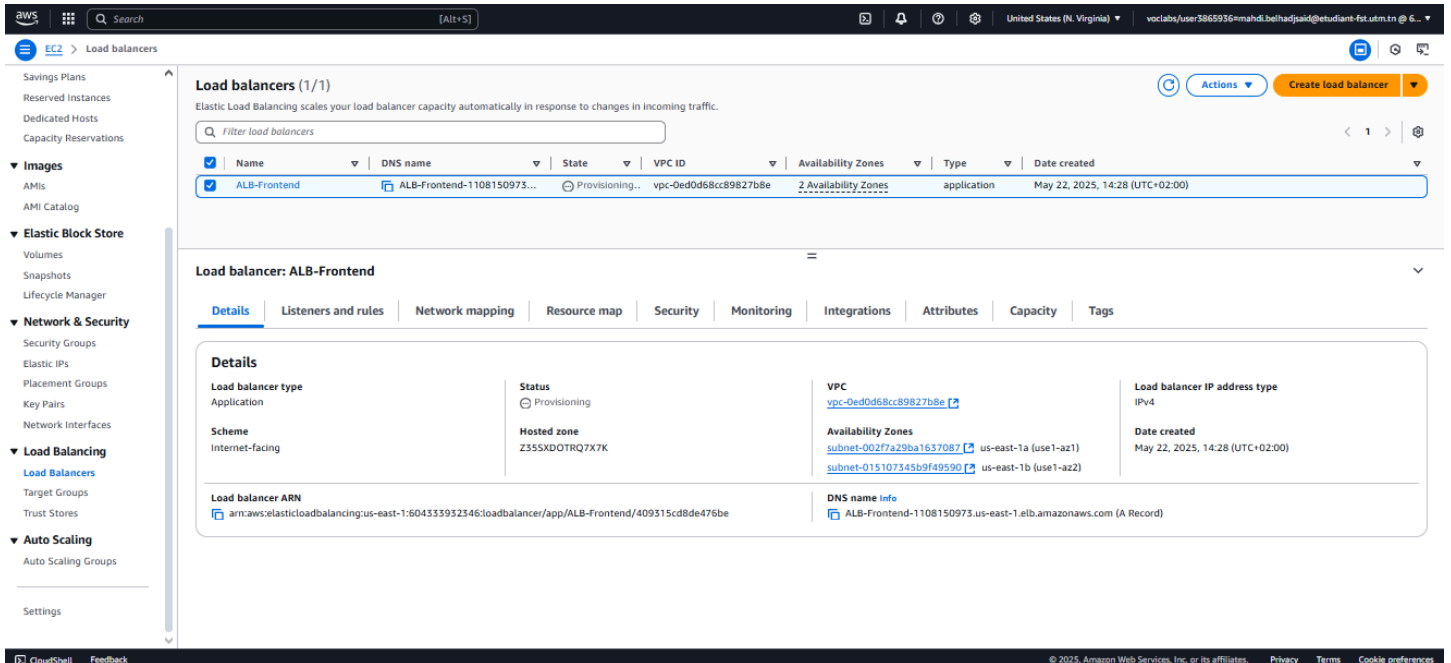
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...
Frontend-A	i-00a028e67c25dc2fd	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1a	-	-
Backend-A	i-0ad4757b6d7af7d06	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1a	-	-
Backend-B	i-03def803ec3b34b01	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1b	-	-
Bastion-Instance	i-05d2b5f9cae950dfa	Running	t2.micro	Initializing	View alarms +	us-east-1a	-	3.222.205.229
Frontend-B	i-0187283131e6aad81	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1b	-	-

- Création et association d'un Target group pour les instances frontend

The screenshot shows the 'Create target group' wizard in the AWS Management Console. The 'Review targets' step is displayed, showing two targets that have been added to the target group. Both targets are in the 'Running' state and are associated with the 'SG-FE' security group. The targets are Frontend-B (i-0187283131e6aad81) and Frontend-A (i-00a028e67c25dc2fd). The wizard is showing the 'Review targets' step, and the 'Create target group' button is visible at the bottom right.

Instance ID	Name	Port	State	Security groups	Zone	Private IPv4 address	Subnet ID	Launch time
i-0187283131e6aad81	Frontend-B	80	Running	SG-FE	us-east-1b	10.0.4.17	subnet-0ce34f3f7e55b4156	May 22, 2025, 13:45 (UTC+02:00)
i-00a028e67c25dc2fd	Frontend-A	80	Running	SG-FE	us-east-1a	10.0.3.148	subnet-08b02cf27e55b4156	May 22, 2025, 13:34 (UTC+02:00)

## • Création du Load Balancer pour le trafic du frontend



**Load balancers (1/1)**

Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.

Filter load balancers

Name	DNS name	State	VPC ID	Availability Zones	Type	Date created
ALB-Frontend	ALB-Frontend-1108150973...	Provisioning...	vpc-0ed0d68cc89827b8e	2 Availability Zones	application	May 22, 2025, 14:28 (UTC+02:00)

**Load balancer: ALB-Frontend**

Details | Listeners and rules | Network mapping | Resource map | Security | Monitoring | Integrations | Attributes | Capacity | Tags

**Details**

**Load balancer type**  
Application

**Status**  
Provisioning

**VPC**  
vpc-0ed0d68cc89827b8e

**Load balancer IP address type**  
IPv4

**Scheme**  
Internet-facing

**Hosted zone**  
Z355XDOTRQ7X7K

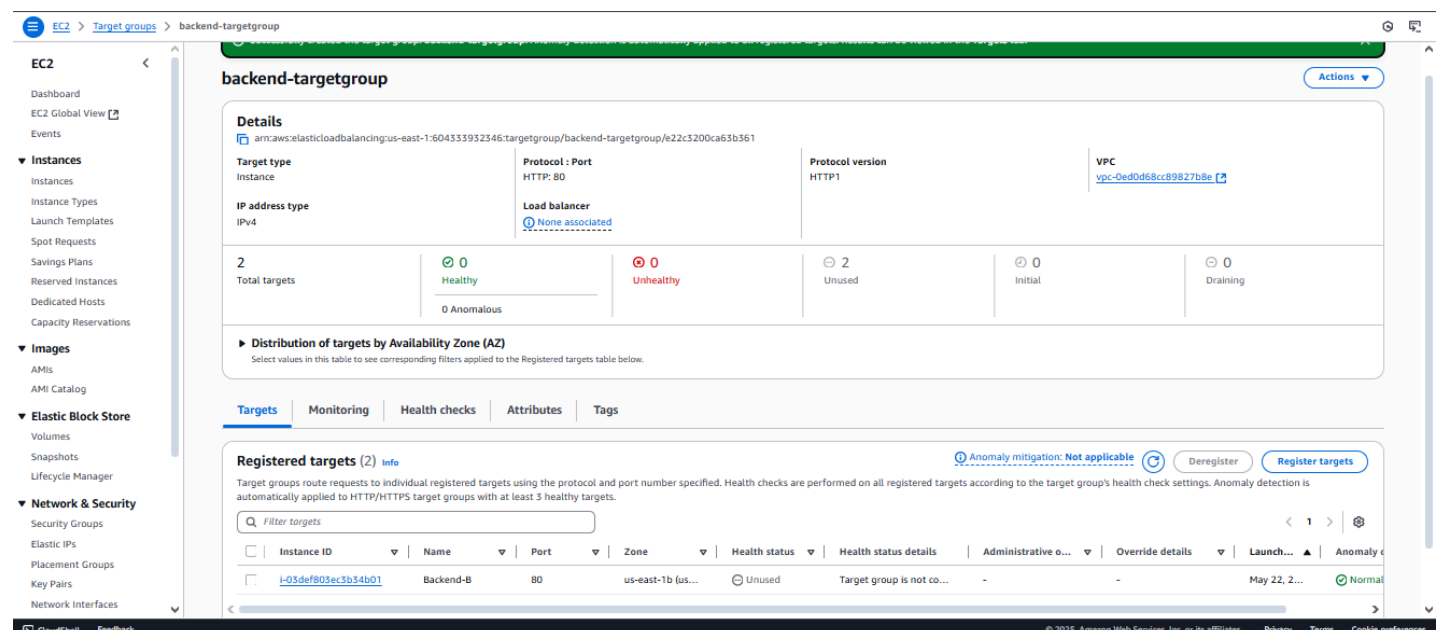
**Availability Zones**  
subnet-00277a29ba1637087 us-east-1a (use1-a21)  
subnet-015107345b9f49590 us-east-1b (use1-a22)

**Date created**  
May 22, 2025, 14:28 (UTC+02:00)

**Load balancer ARN**  
arn:aws:elasticloadbalancing:us-east-1:604333932346:loadbalancer/app/ALB-Frontend/409315cd8de476be

**DNS name info**  
ALB-Frontend-1108150973.us-east-1.elb.amazonaws.com (A Record)

## • Création et association d'un Target group pour les instances backend



**backend-targetgroup**

Details | Targets | Monitoring | Health checks | Attributes | Tags

**Details**

arn:aws:elasticloadbalancing:us-east-1:604333932346:targetgroup/backend-targetgroup/e22c3200ca63b361

**Target type**  
Instance

**Protocol : Port**  
HTTP: 80

**Protocol version**  
HTTP1

**VPC**  
vpc-0ed0d68cc89827b8e

**IP address type**  
IPv4

**Load balancer**  
None associated

**2** Total targets

**0** Healthy

**0** Unhealthy

**0** Anomalous

**2** Unused

**0** Initial

**0** Draining

**Distribution of targets by Availability Zone (AZ)**  
Select values in this table to see corresponding filters applied to the Registered targets table below.

**Registered targets (2)** Info

Target groups route requests to individual registered targets using the protocol and port number specified. Health checks are performed on all registered targets according to the target group's health check settings. Anomaly detection is automatically applied to HTTP/HTTPS target groups with at least 3 healthy targets.

Filter targets

Instance ID	Name	Port	Zone	Health status	Health status details	Administrative o...	Override details	Launch...	Anomaly c
i-03def803ec3b34b01	Backend-B	80	us-east-1b (us...	Unused	Target group is not co...	-	-	May 22, 2...	Normal

## • Création du Load Balancer pour le trafic du backend

The screenshot shows the AWS Management Console 'Load balancers' page. The 'Backend-ALB' is selected, and its details are displayed. The console shows the following information:

Name	DNS name	State	VPC ID	Availability Zones	Type	Date created
ALB-Frontend	ALB-Frontend-1108150973...	Active	vpc-0ed0d68cc89827b8e	2 Availability Zones	application	May 22, 2025, 14:28 (UTC+02:00)
Backend-ALB	Backend-ALB-1617119928...	Provisioning...	vpc-0ed0d68cc89827b8e	2 Availability Zones	application	May 22, 2025, 14:41 (UTC+02:00)

**Load balancer: Backend-ALB**

**Details**

- Load balancer type:** Application
- Scheme:** Internet-facing
- Status:** Provisioning
- Hosted zone:** Z355XDOTRQ7X7K
- VPC:** vpc-0ed0d68cc89827b8e
- Availability Zones:** subnet-03521ee6465377455 (us-east-1a (use1-az1)), subnet-0a4d7399d364ae000 (us-east-1b (use1-az2))
- Load balancer IP address type:** IPv4
- Date created:** May 22, 2025, 14:41 (UTC+02:00)
- Load balancer ARN:** arn:aws:elasticloadbalancing:us-east-1:604333932346:loadbalancer/app/Backend-ALB/59d021b7561ced5b
- DNS name info:** Backend-ALB-1617119928.us-east-1.elb.amazonaws.com (A Record)

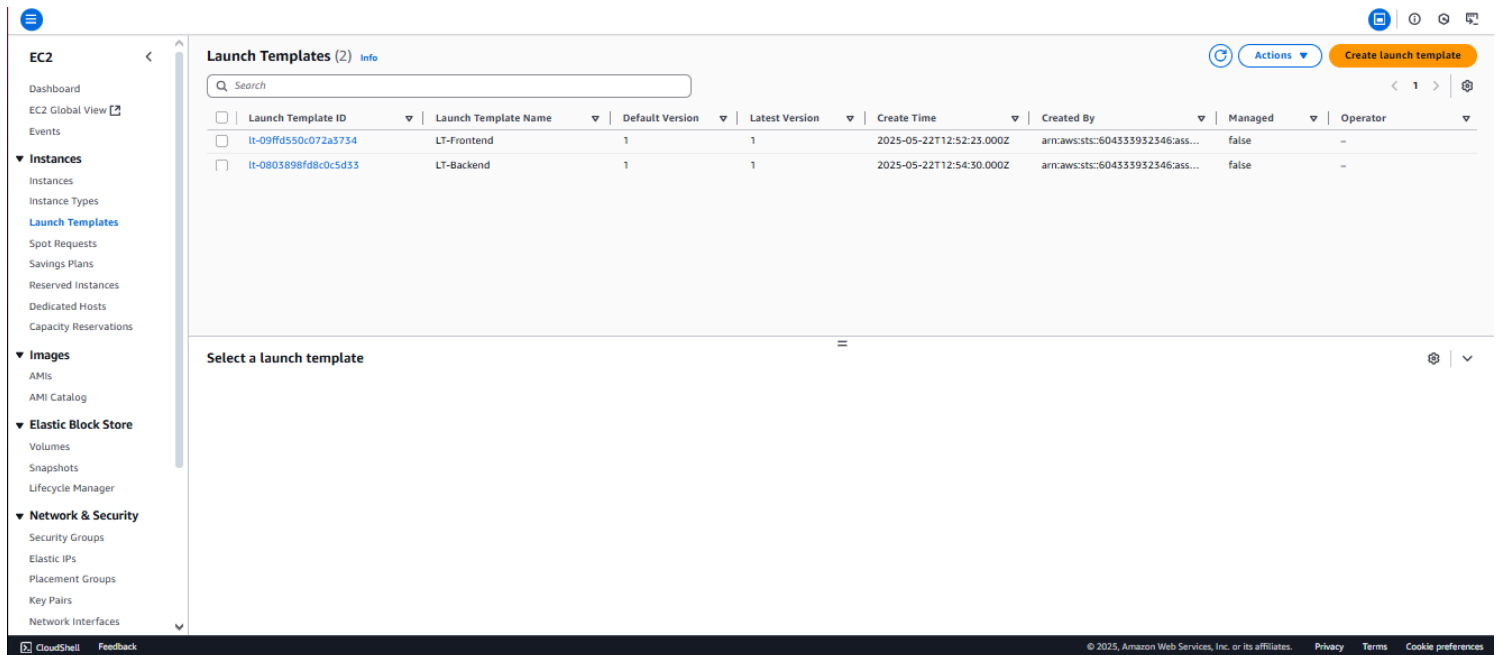
## • Création des Launch templates du backend

The screenshot shows the AWS Management Console 'Launch Templates' page. Two launch templates are listed:

Launch Template ID	Launch Template Name	Default Version	Latest Version	Create Time	Created By	Managed	Operator
lt-09ffd550c072a3734	LT-Frontend	1	1	2025-05-22T12:52:23.000Z	arn:aws:sts:604333932346:ass...	false	-
lt-0803898fd8c0c5d53	LT-Backend	1	1	2025-05-22T12:54:30.000Z	arn:aws:sts:604333932346:ass...	false	-

**Select a launch template**

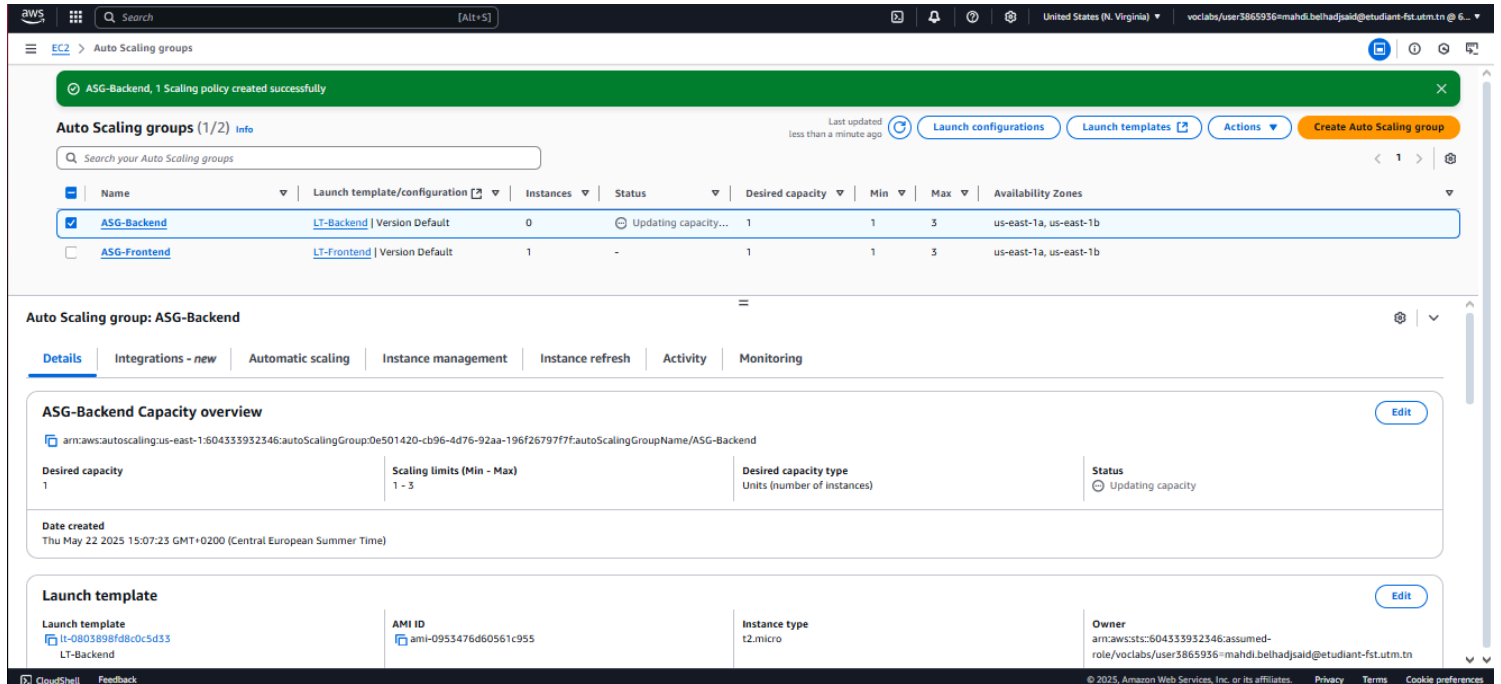
## ● Création des Launch templates du frontend



## ● Creation des Auto Scaling Groupes

1 ASG pour le frontend avec  
un scaling limite = 3  
associé au Launch template du frontend  
> 50% CPU usage pour le Scale In  
< 50% CPU usage pour le Scale Out

1 ASG pour le backend avec  
un scaling limite = 3  
associé au Launch template du backend  
> 50% CPU usage pour le Scale In  
< 50% CPU usage pour le Scale Out

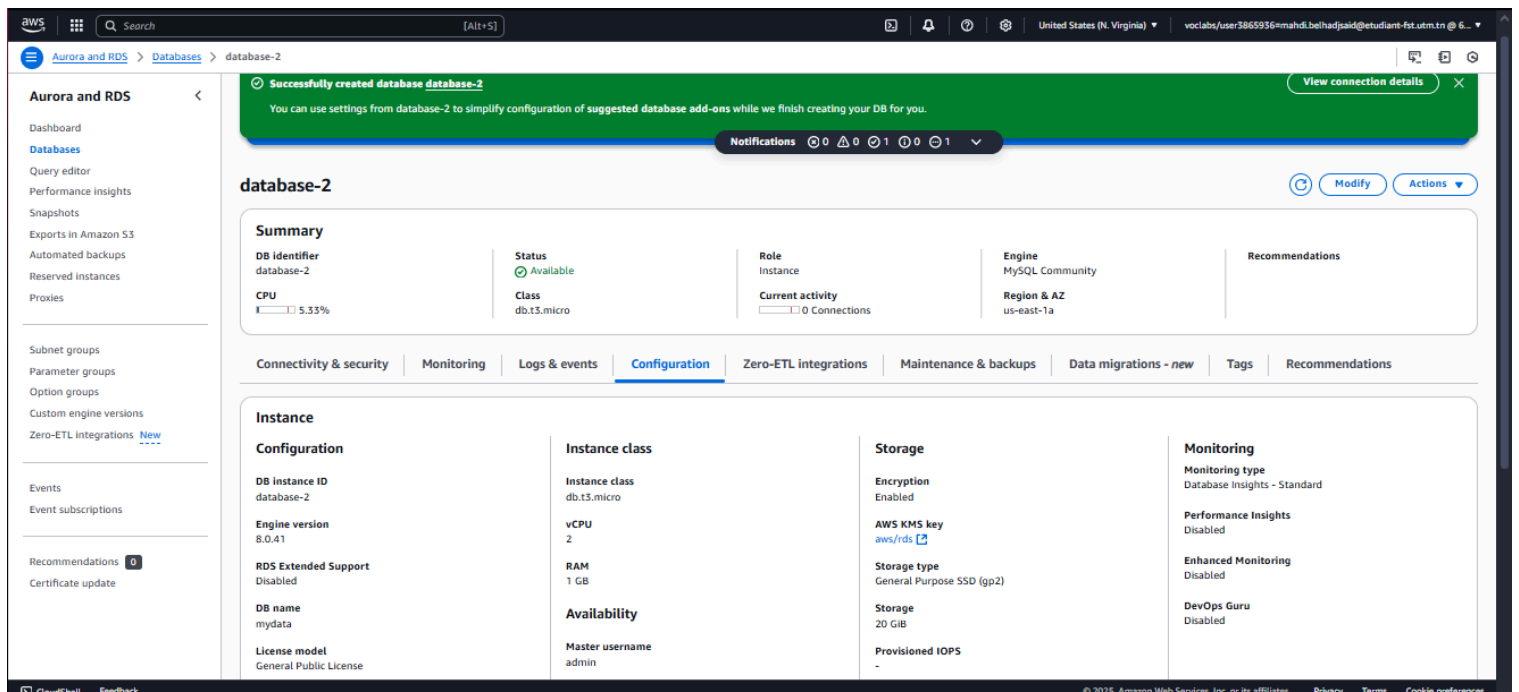


## Étape 5 : Déployer la base de données (Amazon RDS)

### • Création d'une base de données RDS

BD primary + BD standby pour assurer la haute disponibilité.

Attacher le SG-DB pour contrôler l'accès au données

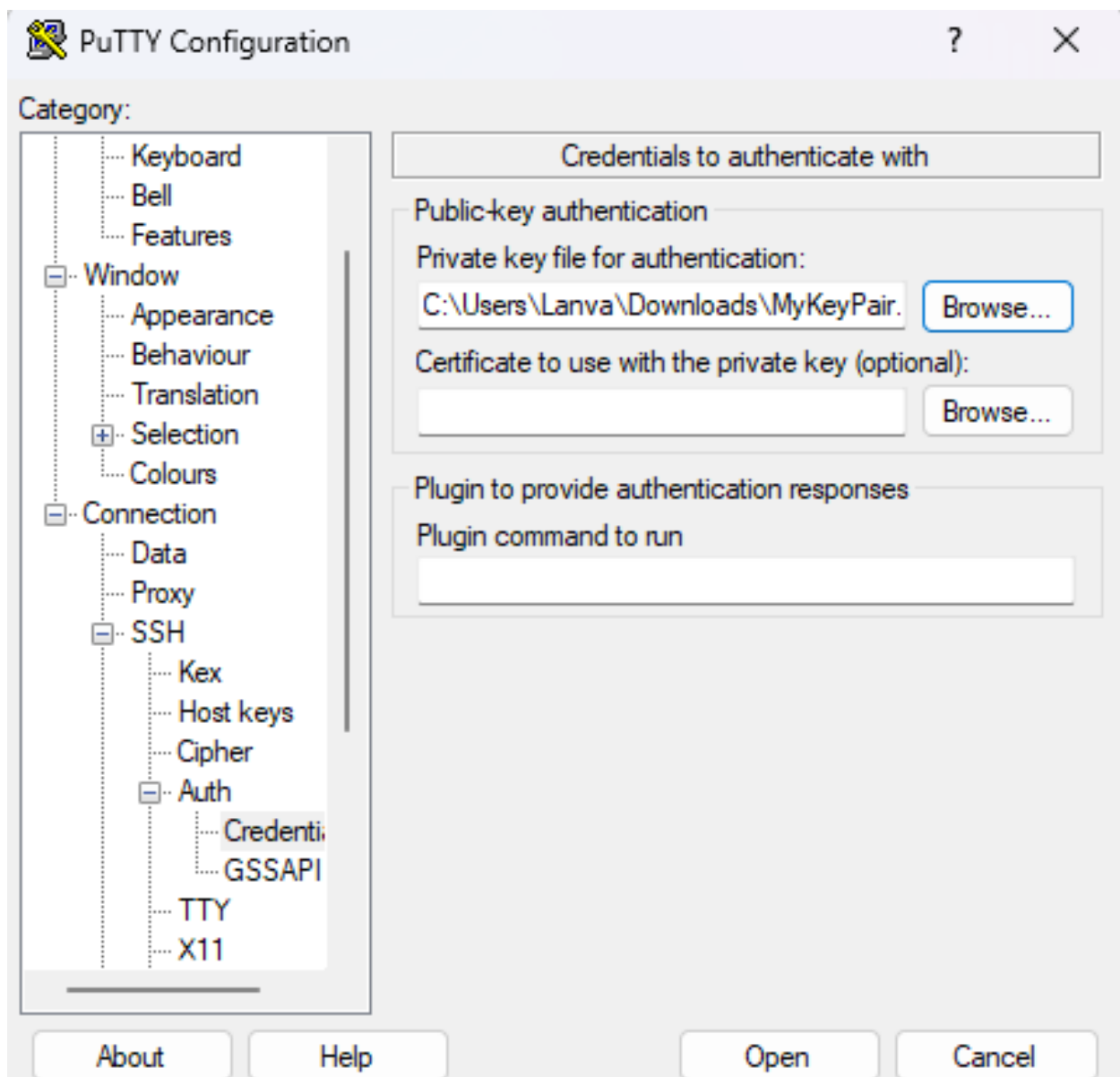




- **Ajouter le script MySQL au base de données**

pour ajouter le script il faut tout d'abord accéder au RDS a travers :

- 1) SSH vers l'instance bastion en utilisant PUTTY et le clé MyKeyPair.ppk





Installation du MySQL avec succès !

```
Complete!  
[ec2-user@ip-10-0-1-196 ~]$ mysql --version  
mysql Ver 15.1 Distrib 10.5.25-MariaDB, for Linux (x86_64) using EditLine wrapper  
[ec2-user@ip-10-0-1-196 ~]$
```

il faut maintenant accéder au RDS pour ajouter le script MySQL

```
Installed:  
  nmap-ncat-3:7.93-4.amzn2023.x86_64  
  
Complete!  
[ec2-user@ip-10-0-1-196 ~]$ nc -zv cloudbd-instance-1.cfj4os55burw.us-east-1.rds.amazonaws.com 3306  
ncat: Version 7.93 ( https://nmap.org/ncat )  
ncat: Connected to 10.0.3.106:3306.  
ncat: 0 bytes sent, 0 bytes received in 0.01 seconds.  
[ec2-user@ip-10-0-1-196 ~]$ ^C  
[ec2-user@ip-10-0-1-196 ~]$ ^[[200~mysql -h cloudbd-instance-1.cfj4os55burw.us-east-1.rds.amazonaws.com -u admin -p  
-bash: $'\E[200~mysql': command not found  
[ec2-user@ip-10-0-1-196 ~]$ mysql -h cloudbd-instance-1.cfj4os55burw.us-east-1.rds.amazonaws.com -u admin -p  
Enter password:  
Welcome to the MariaDB monitor.  Commands end with ; or \g.  
Your MySQL connection id is 393  
Server version: 8.0.39 8bc99e28  
  
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
  
MySQL [(none)]>
```

Connectivité assurée !

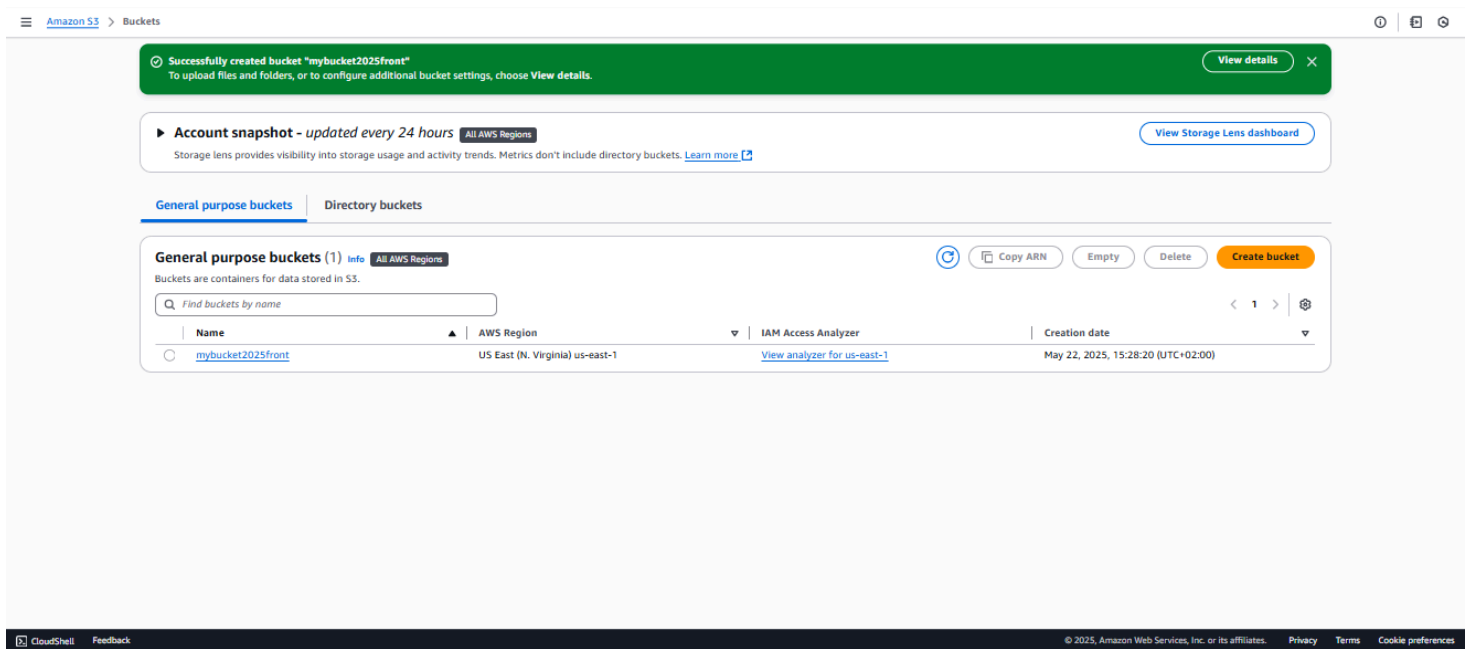
```
ec2-user@ip-10-0-1-196:~  
MySQL [(none)]> CREATE DATABASE cloudbd;  
Query OK, 1 row affected (0.003 sec)  
  
MySQL [(none)]> USE cloudbd;  
Database changed  
MySQL [cloudbd]> CREATE TABLE IF NOT EXISTS messages (  
-> id INT AUTO_INCREMENT PRIMARY KEY,  
-> contenu TEXT NOT NULL  
-> );  
Query OK, 0 rows affected (0.022 sec)  
  
MySQL [cloudbd]> INSERT INTO messages (contenu) VALUES  
-> ('Bonjour, ceci est le message 1'),  
-> ('Voici un deuxième message'),  
-> ('Et enfin, le troisième message');  
Query OK, 3 rows affected (0.004 sec)  
Records: 3 Duplicates: 0 Warnings: 0  
  
MySQL [cloudbd]> SELECT * FROM messages ;  
+----+-----+  
| id | contenu |  
+----+-----+  
| 1 | Bonjour, ceci est le message 1 |  
| 2 | Voici un deuxième message |  
| 3 | Et enfin, le troisième message |  
+----+-----+  
3 rows in set (0.001 sec)  
  
MySQL [cloudbd]>
```

on a exécuté le script MySQL

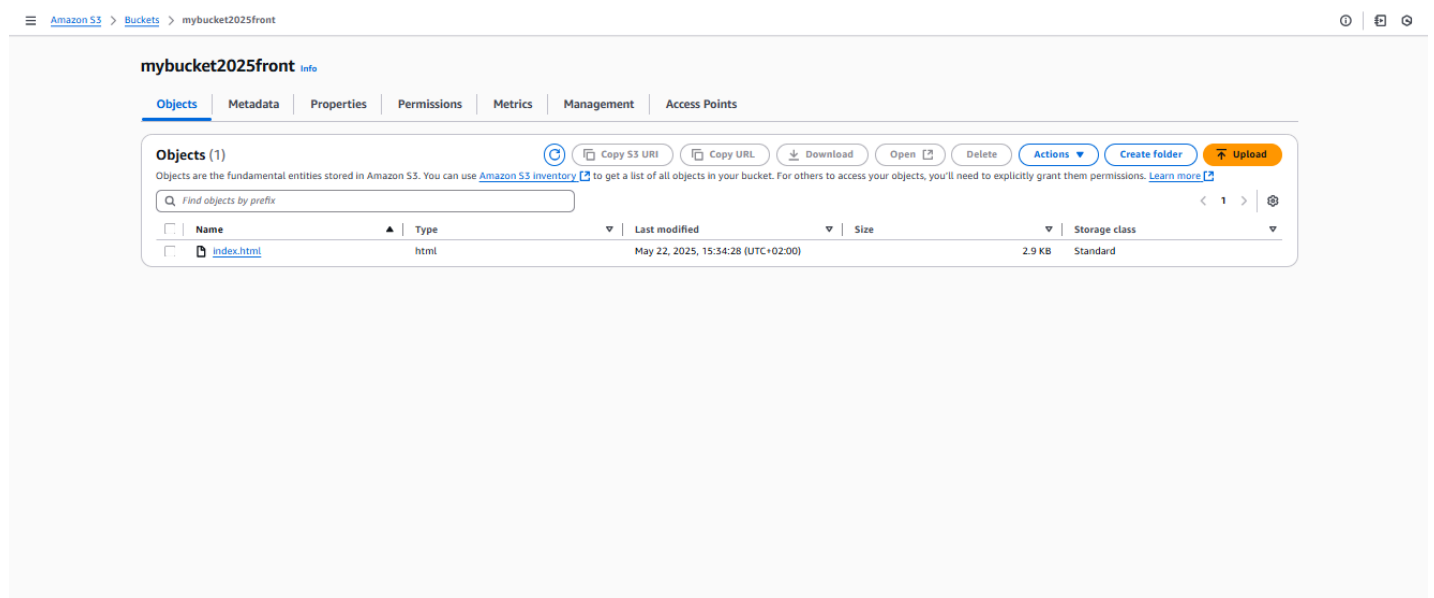
Donc maintenant la base de données est prêt et bien virtualisé

## Étape 6 : Déployer un bucket S3 et CloudFront

- Création d'une S3 bucket pour les ressources statics qu'on a :



- Ajouter des fichier index.html et app.py dans la bucket



## • Modifier les Launch Templates

Pour LT-frontend on va modifier la partie UserData pour ajouter la configuration (les bibliothèques nécessaires , les frameworks , les langages et même la connexion avec la S3 bucket en utilisant la bucket endpoint ) de l'instance d'est le démarrage .

### Script UserData du LT-frontend :

```
#!/bin/bash

# Met à jour le système et installer nginx
sudo yum update -y
sudo yum install -y nginx

# Crée le dossier web si il n'existe pas
sudo mkdir -p /usr/share/nginx/html

# Télécharge le fichier index.html depuis S3 (bucket public)
sudo wget
https://mybucket2025front.s3.amazonaws.com/frontend/index.html -O
/usr/share/nginx/html/index.html

# Crée le fichier de configuration nginx
sudo bash -c 'cat > /etc/nginx/conf.d/default.conf <<EOF
server {
    listen 80 default_server;
    listen [::]:80 default_server;

    root /usr/share/nginx/html;
    index index.html;

    location = /health {
        default_type text/plain;
        return 200 "OK";
        access_log off;
    }
}
```

```
location / {  
    try_files $uri $uri/ =404;  
}  
}  
EOF'
```

```
# Démarre nginx  
sudo systemctl start nginx  
sudo systemctl enable nginx
```

## Script UserData du LT-backend :

```
#!/bin/bash
```

```
# 1. Mettre à jour le système  
sudo yum update -y
```

```
# 2. Installer Python, MySQL client, et outils nécessaires  
sudo yum install -y python3 python3-pip mysql wget gcc openssl-devel  
bzip2-devel libffi-devel python3-devel
```

```
# 3. Créer le dossier de l'application backend  
sudo mkdir -p /var/www/backend  
cd /var/www/backend
```

```
# 4. Télécharger app.py depuis ton bucket S3  
sudo wget https://mybucket2025front.s3.amazonaws.com/backend/app.py  
-O app.py
```

```
# 5. Créer un environnement virtuel Python  
python3 -m venv venv
```

```
# 6. Installer les dépendances Flask  
source venv/bin/activate  
pip install flask pymysql Flask-SQLAlchemy flask-cors python-dotenv  
cryptography  
deactivate
```

# 7. Créer le fichier .env avec ta configuration MySQL locale

```
sudo bash -c 'echo
```

```
"DATABASE_URL=mysql+pymysql://admin:mahdi123@cloudbd.c4dfg7klvxy  
z.us-east-1.rds.amazonaws.com:3306/myapp" > /var/www/backend/.env'
```

# 8. Protéger le fichier d'environnement

```
sudo chmod 600 /var/www/backend/.env
```

```
sudo chown ec2-user:ec2-user /var/www/backend/.env
```

# 9. Créer un service systemd pour lancer Flask avec le serveur intégré

```
sudo bash -c 'cat > /etc/systemd/system/flask-app.service << EOL
```

```
[Unit]
```

```
Description=Flask Application
```

```
After=network.target
```

```
[Service]
```

```
User=ec2-user
```

```
Group=ec2-user
```

```
WorkingDirectory=/var/www/backend
```

```
EnvironmentFile=/var/www/backend/.env
```

```
ExecStart=/var/www/backend/venv/bin/python3 -m flask run --host=0.0.0.0  
--port=8080
```

```
Restart=always
```

```
[Install]
```

```
WantedBy=multi-user.target
```

```
EOL'
```

# 10. Donner les bons droits

```
sudo chown -R ec2-user:ec2-user /var/www/backend
```

```
sudo chmod -R 755 /var/www/backend
```

# 11. Activer et lancer le service

```
sudo systemctl daemon-reexec
```

```
sudo systemctl daemon-reload
```

```
sudo systemctl start flask-app
```

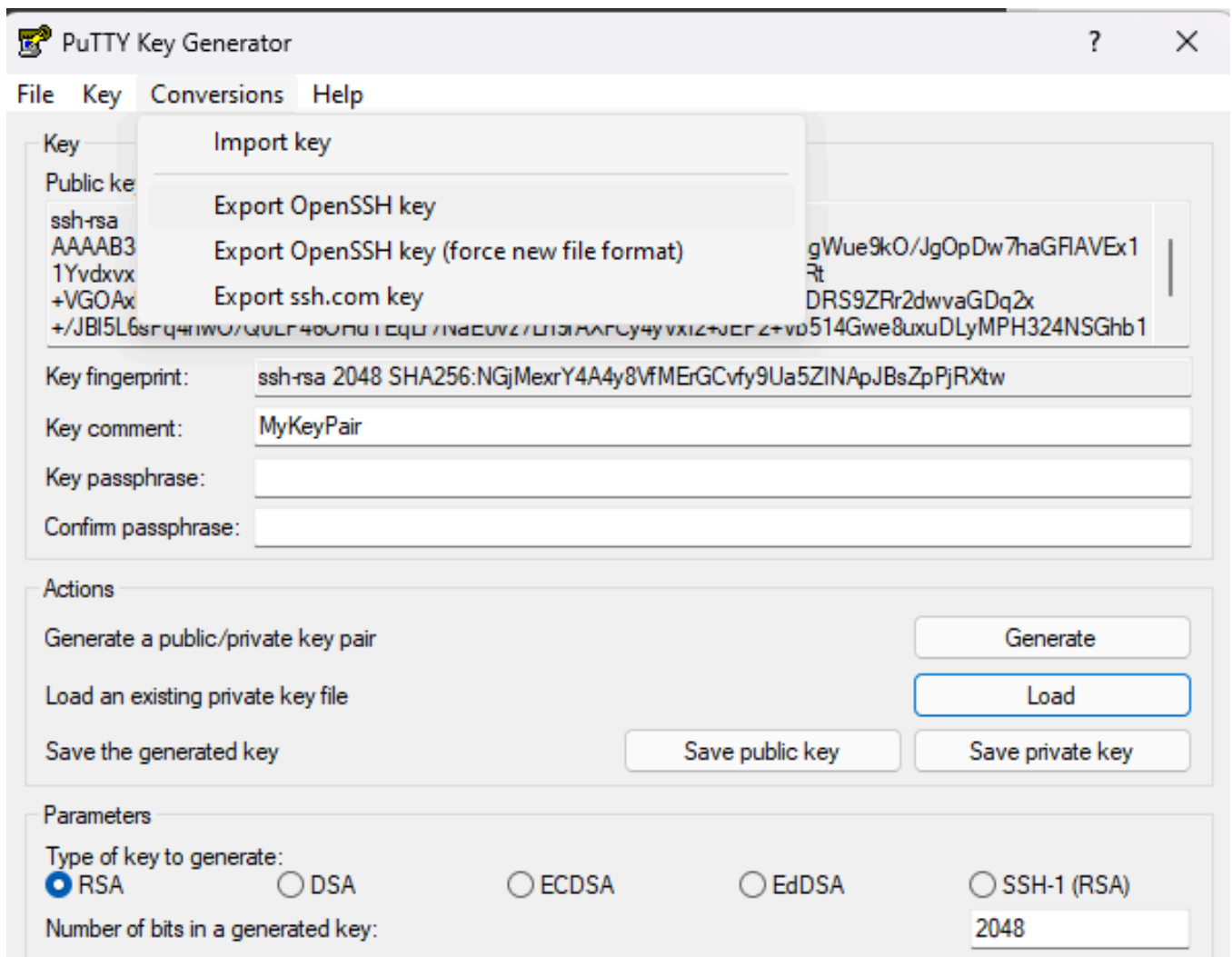
```
sudo systemctl enable flask-app
```



- **Accès au base de données depuis l'instance EC2 backend**

Pour faire ça on a choisi de connecter à l'instance EC2 backend depuis l'instance EC2 bastion à travers un SSH et ensuite connecter EC2 backend au RDS

- 1) **Conversion des clé du MyKeyPair.ppk au MyKeyPair.pem localement car linux utilise la version .pem**



## 2) envoi du nouveau clé vers le bastion EC2 :

```
PS C:\WINDOWS\system32> pscp -1 C:\Users\Lanva\Downloads\MyKeyPair.ppk C:\Users\Lanva\Downloads\MyKeyPair.pem ec2-user@35.175.124.235:/home/ec2-user/
>>
MyKeyPair.pem          | 1 kB | 1.6 kB/s | ETA: 00:00:00 | 100%
PS C:\WINDOWS\system32>
```

### 3) SSH vers la backend EC2 :

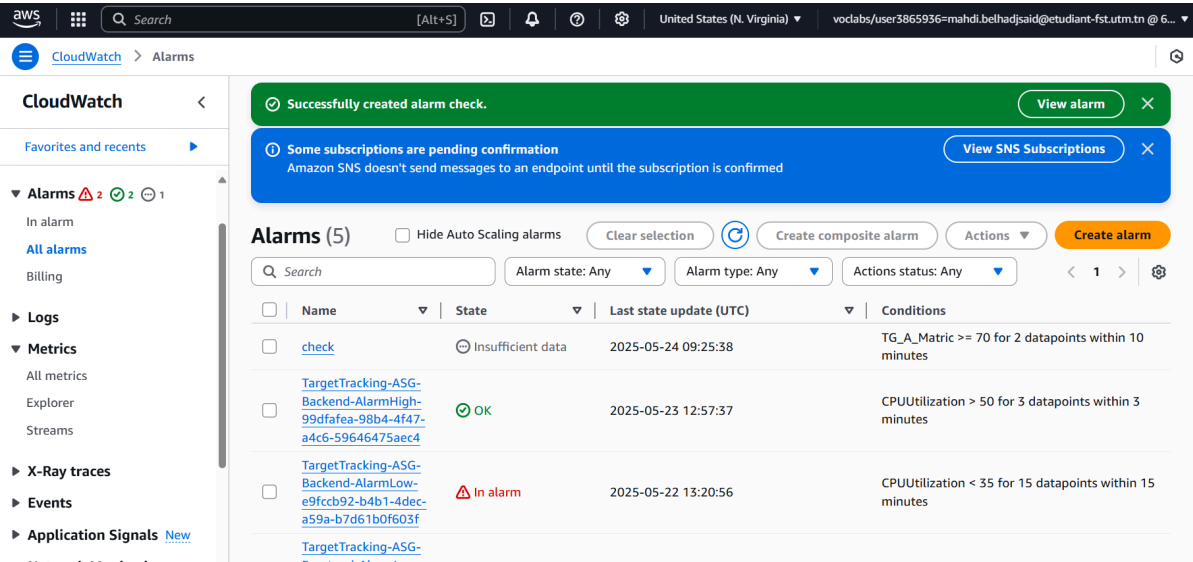
```
ec2-user@ip-10-0-6-243:~  
Last login: Fri May 23 11:59:15 2025 from 41.225.235.206  
[ec2-user@ip-10-0-1-196 ~]$ chmod 400 MyKeyPair.pem  
[ec2-user@ip-10-0-1-196 ~]$ ssh -i MyKeyPair.pem ec2-user@10.0.6.243  
The authenticity of host '10.0.6.243 (10.0.6.243)' can't be established.  
ED25519 key fingerprint is SHA256:jIws+85FPKepd3fDLLXsaNTZwfMHt5RoQqt29CfdYmY.  
This key is not known by any other names  
Are you sure you want to continue connecting (yes/no/[fingerprint])? y  
Please type 'yes', 'no' or the fingerprint: yes  
Warning: Permanently added '10.0.6.243' (ED25519) to the list of known hosts.  
  
#_##### Amazon Linux 2023  
~\#####\  
NN\#####\  
NN\###|  
NN\#/_____  
NNV~'-> https://aws.amazon.com/linux/amazon-linux-2023
```

#### 4) Faire une copie du fichier [app.py](#) dans le backend pour tester l'application

```
lrwxr-xr-x. 2 root root 6 May 23 12:46 backend
ec2-user@ip-10-0-5-214 ~]$ aws s3 cp s3://mybucket2025front/backend/app.py /var
/www/backend/app.py
Fatal error: Unable to locate credentials
ec2-user@ip-10-0-5-214 ~]$ fatal error: Unable to locate cr
-bash: fatal: command not found
ec2-user@ip-10-0-5-214 ~]$ aws configure
AWS Access Key ID [None]: madi
AWS Secret Access Key [None]: mahdi
Default region name [None]: us-east-1
Default output format [None]: json
ec2-user@ip-10-0-5-214 ~]$ aws s3 cp s3://mybucket2025front/backend/app.py /var
/www/backend/app.py
Fatal error: An error occurred (403) when calling the HeadObject operation: Forb
idden
ec2-user@ip-10-0-5-214 ~]$
```

## ETAPE 6 : sécurité :

- **Creation du CloudWatch:**



The screenshot shows the AWS CloudWatch Alarms console. At the top, there's a green notification bar stating "Successfully created alarm check." and a blue notification bar stating "Some subscriptions are pending confirmation". Below these, the "Alarms (5)" section is visible. It includes a search bar, filters for "Alarm state: Any", "Alarm type: Any", and "Actions status: Any". A table lists the alarms:

	Name	State	Last state update (UTC)	Conditions
<input type="checkbox"/>	check	Insufficient data	2025-05-24 09:25:38	TG_A_Metric >= 70 for 2 datapoints within 10 minutes
<input type="checkbox"/>	TargetTracking-ASG-Backend-AlarmHigh-99dfafea-98b4-4f47-a4c6-59646475aec4	OK	2025-05-23 12:57:37	CPUUtilization > 50 for 3 datapoints within 3 minutes
<input type="checkbox"/>	TargetTracking-ASG-Backend-AlarmLow-e9fccb92-b4b1-4dec-a59a-b7d61b0f603f	In alarm	2025-05-22 13:20:56	CPUUtilization < 35 for 15 datapoints within 15 minutes
<input type="checkbox"/>	TargetTracking-ASG-Backend-AlarmLow-e9fccb92-b4b1-4dec-a59a-b7d61b0f603f	In alarm	2025-05-22 13:20:56	CPUUtilization < 35 for 15 datapoints within 15 minutes

- Creation des cloud trails

CloudTrail > Trails

Trail successfully created

Trails

Copy events to LakeDeleteCreate trail

	Name ▲	Home region ▼	Multi-region trail ▼	ARN ▼	Insights ▼	Organization trail ▼	S3 bucket ▼	Log file prefix ▼	CloudWatch Logs log group ▼	Status ▼
	<a href="#">my-trail</a>	US East (N. Virginia)	Yes	arn:aws:cloudtrail:us-east-1:60433393:trail/my-trail	Enabled	No	<a href="#">mybucket2025front</a>	-	-	Logging

CloudShellFeedback© 2025, Amazon Web Services, Inc. or its affiliates. PrivacyTermsCookie preferences