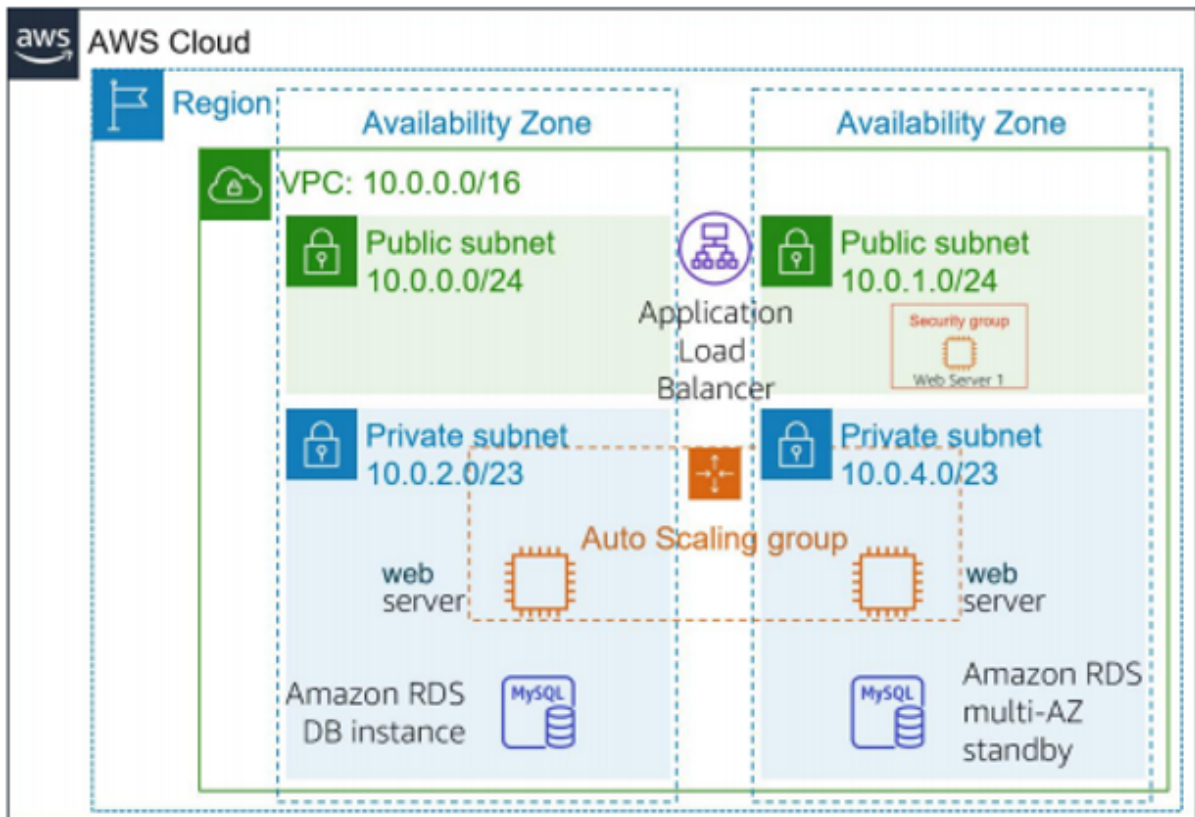


Livrable du projet AWS

Déploiement d'une application web hautement disponible (HA) sur AWS



Ouerghi Mourad

IGL 4

Table de matières

1. Création du VPC.....	2
2. Création des subnets.....	3
3. Routes tables.....	5
a. Création des route tables.....	5
b. Association des route tables	6
4. Création du Internet Gateway.....	7
5. Création des Nat-Gateways.....	9
6. Création d'une base de données MYSQL avec RDS.....	10
a. Subnet Group.....	10
b. L'instance RDS.....	11
c. Security Group	12
7. Création des instances EC2.....	13
a. Création du machine bastion.....	13
i. L'instance Bastion.....	13
ii. Security Group.....	14
b. Création du machine Web Server.....	15
i. L'instance Web Server.....	15
ii. Security Group.....	16
8. Déploiement de l'application web	17
a. Liaison de l'application web avec l'instance RDS.....	18
b. Installation de l'environnement nécessaire	18
c. Déploiement de l'application web dans l'instance EC2.....	19
9. Création du load balancer.....	20
a. Création du Target Group.....	20
b. Security Group.....	21
c. LoadBalancer.....	22
10. Création du AutoScaling Group.....	23
a. Création du Launch Configuration.....	23
i. Création d'une AMI à partir du web server.....	23
ii. Création du Launch Configuration.....	23
b. Création d'Auto Scaling Group.....	24

1. Création du VPC

Dans cette partie j'ai créé un vpc avec un IPV4 CIDR = 10.0.0.0/16 sous le nom de "project-vpc"

The screenshot shows the AWS Management Console interface for creating a new VPC. The page title is "Create VPC" with an "Info" link. Below the title, a brief description states: "A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances." The "VPC settings" section contains the following options:

- Resources to create:** Two radio buttons are present: "VPC only" (selected) and "VPC and more".
- Name tag - optional:** A text input field contains the value "project-vpc".
- IPv4 CIDR block:** Two radio buttons are present: "IPv4 CIDR manual input" (selected) and "IPAM-allocated IPv4 CIDR block". Below this, a text input field contains the value "10.0.0.0/16".
- IPv6 CIDR block:** Three radio buttons are present: "No IPv6 CIDR block" (selected), "IPAM-allocated IPv6 CIDR block", and "Amazon-provided IPv6 CIDR block".
- Tenancy:** A dropdown menu is set to "Default".

The "Tags" section includes a description: "A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs." It features a table with two columns: "Key" and "Value - optional". One tag is listed with the key "Name" and the value "project-vpc". There are "X" icons to remove the tag and a "Remove" button. Below the table is an "Add new tag" button and a note: "You can add 0/5 more tags." At the bottom of the settings panel are "Cancel" and "Create VPC" buttons. The footer of the console shows a "Feedback" link, a language selection prompt, and copyright information for 2022.

2. Création des subnets

Dans cette étape j'ai créé une public subnet sous le nom de project-public-subnet 1 avec un CIDR IPV4 = 10.0.0.0/24 et une private subnet le nom de project-private-subnet 1 avec un CIDR IPV4 = 10.0.0.2/23, les deux dans l'availability zone us-east-1a .

The screenshot displays the AWS Management Console interface for creating subnets. It shows two subnets being configured in the 'us-east-1a' availability zone. The first subnet, 'project-public-subnet-1', has a CIDR block of 10.0.0.0/24. The second subnet, 'project-private-subnet-1', has a CIDR block of 10.0.0.2/23. Both subnets are tagged with 'Name' and their respective names. The interface includes fields for subnet name, availability zone, IPv4 CIDR block, and tags, along with buttons for adding and removing tags.

Subnet 1 of 4

Subnet name
Create a tag with a key of 'Name' and a value that you specify.
project-public-subnet-1
The name can be up to 256 characters long.

Availability Zone: [info](#)
Choose the zone in which your subnet will reside, or let Amazon choose one for you.
US East (N. Virginia) / us-east-1a

IPv4 CIDR block: [info](#)
10.0.0.0/24

Tags - optional

Key	Value - optional
Q, Name	Q, project-public-subnet-1

[Add new tag](#)
You can add 49 more tags.

[Remove](#)

Subnet 2 of 4

Subnet name
Create a tag with a key of 'Name' and a value that you specify.
project-private-subnet-1
The name can be up to 256 characters long.

Availability Zone: [info](#)
Choose the zone in which your subnet will reside, or let Amazon choose one for you.
US East (N. Virginia) / us-east-1a

IPv4 CIDR block: [info](#)
10.0.0.2/23

Tags - optional

Key	Value - optional
Q, Name	Q, project-private-subnet-1

[Add new tag](#)

© 2022, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

Dans cette étape j'ai créé une public subnet sous le nom de project-public-subnet 2 avec un CIDR IPV4 = 10.0.1.0/24 et une private subnet le nom de project-private-subnet 2 avec un CIDR IPV4 = 10.0.4.0/23, les deux dans l'availability zone us-east-1b .

The screenshot displays the AWS Management Console interface for configuring subnets. The top navigation bar shows the AWS logo, 'Services' menu, a search bar, and the user's profile. The main content area is divided into two panels for 'Subnet 3 of 4' and 'Subnet 4 of 4'.

Subnet 3 of 4 (Public Subnet):

- Subnet name:** project-public-subnet-2
- Availability Zone:** US East (N. Virginia) / us-east-1b
- IPv4 CIDR block:** 10.0.1.0/24
- Tags - optional:** Key: Name, Value: project-public-subnet-2

Subnet 4 of 4 (Private Subnet):

- Subnet name:** project-private-subnet-2
- Availability Zone:** US East (N. Virginia) / us-east-1b
- IPv4 CIDR block:** 10.0.4.0/23
- Tags - optional:** Key: Name, Value: project-private-subnet-2

The bottom of the console shows a footer with 'Feedback', 'Looking for language selection? Find it in the new Unified Settings', and copyright information for Amazon Web Services, Inc. or its affiliates.

3. Routes tables

a. Création des route tables

Dans cette étape j'ai créé une public route table sous le nom de project-public-route-table dans le vpc "project-vpc"

The screenshot shows the AWS Management Console interface for creating a new route table. The breadcrumb navigation at the top indicates the path: VPC > Route tables > Create route table. The main heading is 'Create route table' with a help icon. Below this, a brief description states: 'A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.' The form is divided into two main sections: 'Route table settings' and 'Tags'. In the 'Route table settings' section, there is a text input for 'Name - optional' containing 'project-public-route-table' and a dropdown menu for 'VPC' showing 'vpc-0060554dc37d5399f (project-vpc)'. The 'Tags' section includes a description of tags and a table with two columns: 'Key' and 'Value - optional'. A single tag is present with the key 'Name' and the value 'project-public-route-table'. There are buttons for 'Add new tag', 'Cancel', and 'Create route table' (which is orange).

Dans cette étape j'ai créé une private route table sous le nom de project-private route table dans le vpc "project-vpc" et j'ai créé une deuxième ensuite puisque on utilisera deux nat gateways ensuite

This screenshot is similar to the previous one, showing the 'Create route table' page in the AWS console. The breadcrumb navigation remains the same. The 'Route table settings' section now shows the name 'project-private-route-table' in the 'Name - optional' field, while the 'VPC' dropdown still shows 'vpc-0060554dc37d5399f (project-vpc)'. In the 'Tags' section, the tag value has been updated to 'project-private-route-table'. The 'Add new tag', 'Cancel', and 'Create route table' (orange) buttons are visible at the bottom of the form.

b. Association des route tables

Association des deux public subnets créé précédemment avec public route table

The screenshot shows the AWS Management Console interface for editing subnet associations. The breadcrumb trail is VPC > Route tables > rtb-04d59608740b0c29c > Edit subnet associations. The page title is 'Edit subnet associations' with a subtitle 'Change which subnets are associated with this route table.' Below this, there is a section for 'Available subnets (2/4)' with a search bar. A table lists the available subnets:

	Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
<input type="checkbox"/>	project-private-subnet-2	subnet-070907a87d6e44469	10.0.4.0/23	--	rtb-0a13440ac6a1d286 / project-private-route-table-2
<input checked="" type="checkbox"/>	project-public-subnet-2	subnet-00e911318c3b5f65	10.0.1.0/24	--	rtb-04d59608740b0c29c / project-public-route-table
<input checked="" type="checkbox"/>	project-public-subnet-1	subnet-06d40c611ca89db1	10.0.0/24	--	rtb-04d59608740b0c29c / project-public-route-table
<input type="checkbox"/>	project-private-subnet-1	subnet-064e9fa92daca75f	10.0.2.0/23	--	rtb-025a1d5993170069b / project-private-route-table-1

Below the table, the 'Selected subnets' section shows two subnets selected: 'subnet-06d40c611ca89db1 / project-public-subnet-1' and 'subnet-00e911318c3b5f65 / project-public-subnet-2'. At the bottom right, there are 'Cancel' and 'Save associations' buttons.

Association du private subnet “project-private-subnet-1” avec le “project-private-rout-table” et j’ai refait la même étape avec le “project-private-subnet-2” mais avec le “project-private-rout-table-2”

The screenshot shows the AWS Management Console interface for editing subnet associations. The breadcrumb trail is VPC > Route tables > rtb-025a1d5993170069b > Edit subnet associations. The page title is 'Edit subnet associations' with a subtitle 'Change which subnets are associated with this route table.' Below this, there is a section for 'Available subnets (1/4)' with a search bar. A table lists the available subnets:

	Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
<input type="checkbox"/>	project-private-subnet-2	subnet-070907a87d6e44469	10.0.4.0/23	--	Main (rtb-01ff1291ba0841678)
<input type="checkbox"/>	project-public-subnet-2	subnet-00e911318c3b5f65	10.0.1.0/24	--	rtb-04d59608740b0c29c / project-public-route-table
<input type="checkbox"/>	project-public-subnet-1	subnet-06d40c611ca89db1	10.0.0/24	--	rtb-04d59608740b0c29c / project-public-route-table
<input checked="" type="checkbox"/>	project-private-subnet-1	subnet-064e9fa92daca75f	10.0.2.0/23	--	Main (rtb-01ff1291ba0841678)

Below the table, the 'Selected subnets' section shows one subnet selected: 'subnet-064e9fa92daca75f / project-private-subnet-1'. At the bottom right, there are 'Cancel' and 'Save associations' buttons.

4. Création du Internet Gateway

Création d'un Internet Gateway sous le nom de “project-internet-gateway”

The screenshot shows the AWS Management Console interface for creating a new internet gateway. The breadcrumb navigation at the top indicates the path: VPC > Internet gateways > Create internet gateway. The main heading is 'Create internet gateway' with an 'info' icon. A descriptive paragraph states: 'An internet gateway is a virtual router that connects a VPC to the internet. To create a new internet gateway specify the name for the gateway below.' The 'Internet gateway settings' section contains a 'Name tag' field with the value 'project-internet-gateway'. Below this, the 'Tags - optional' section explains that a tag is a label for an AWS resource. It shows a table with one tag: Key 'Name' and Value 'project-internet-gateway'. There are buttons for 'Add new tag', 'Cancel', and 'Create internet gateway'.

aws Services Search [Alt+S]

VPC > Internet gateways > Create internet gateway

Create internet gateway [info](#)

An internet gateway is a virtual router that connects a VPC to the internet. To create a new internet gateway specify the name for the gateway below.

Internet gateway settings

Name tag
Creates a tag with a key of 'Name' and a value that you specify.

project-internet-gateway

Tags - optional
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - optional	
Name	project-internet-gateway	Remove

Add new tag

You can add 49 more tags.

Cancel Create internet gateway

Feedback Looking for language selection? Find it in the new [Unified Settings](#)

© 2022, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Liaison du internet Gateway créé avec le “project-vpc”

The screenshot shows the AWS Management Console interface for attaching an existing internet gateway to a VPC. The breadcrumb navigation at the top indicates the path: VPC > Internet gateways > Attach to VPC (igw-003135f6b078e4ec5). The main heading is 'Attach to VPC (igw-003135f6b078e4ec5)' with an 'info' icon. A descriptive paragraph states: 'Attach an internet gateway to a VPC to enable the VPC to communicate with the internet. Specify the VPC to attach below.' The 'VPC' section contains a search box for 'Available VPCs' with the value 'vpc-006054ac37d5399f'. Below this, there is a section for 'AWS Command Line Interface command'. There are buttons for 'Cancel' and 'Attach internet gateway'.

aws Services Search [Alt+S]

VPC > Internet gateways > Attach to VPC (igw-003135f6b078e4ec5)

Attach to VPC (igw-003135f6b078e4ec5) [info](#)

Attach an internet gateway to a VPC to enable the VPC to communicate with the internet. Specify the VPC to attach below.

VPC

Available VPCs
Attach the internet gateway to this VPC.

vpc-006054ac37d5399f

AWS Command Line Interface command

Cancel Attach internet gateway

Feedback Looking for language selection? Find it in the new [Unified Settings](#)

© 2022, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Ajout d'une route vers le internet gateway créé dans le public route table

VPC > Route tables > rtb-042596087400d629c > Edit routes

Edit routes

Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No
0.0.0.0/0	igw-003155f6d078e4ecf	-	No

[Add route](#)

[Cancel](#) [Preview](#) [Save changes](#)

[Feedback](#) Looking for language selection? Find it in the new [Unified Settings](#)

© 2022, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

5. Création des Nat-Gateways

Création d'un nat gateway sous le nom de “project-nat-gateway-1” dans le “public-subnet-1”. J'ai refait la même étape avec un “project-nat-gateway-2” dans le “public-subnet-2”

Create NAT gateway [info](#)

A highly available, managed Network Address Translation (NAT) service that instances in private subnets can use to connect to services in other VPCs, on-premises networks, or the internet.

NAT gateway settings

Name - optional
Create a tag with a key of 'Name' and a value that you specify.
project-nat-gateway-1
The name can be up to 256 characters long.

Subnet
Select a subnet in which to create the NAT gateway.
subnet-06d4bc61ca89d8c1 (project-public-subnet-1)

Connectivity type
Select a connectivity type for the NAT gateway.
☒ Public
☐ Private

Elastic IP allocation ID [info](#)
Assign an Elastic IP address to the NAT gateway.
eipalloc-0360d23866e82b980 [Allocate Elastic IP](#)

Additional settings

Tags
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key **Value** - optional
Name project-nat-gateway-1 [Remove](#)

[Add new tag](#)
You can add 49 more tags.

[Cancel](#) [Create NAT gateway](#)

Ajout d'une route vers le nat gateway créé dans le private route table et j'ai refait la même étape avec le deuxième nat gateway et le deuxième route table

Edit routes

Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No
0.0.0.0/0	nat-0f1b514a6a9d2f1f	-	No

[Add route](#) [Cancel](#) [Preview](#) [Save changes](#)

6. Création d'une base de données MySQL avec RDS

a. Subnet group

Création d'un subnet group qui avec les deux private subnet groups pour l'utiliser ensuite dans la création d'une instance RDS

The screenshot shows the AWS Management Console interface for creating an Amazon RDS subnet group. The left sidebar contains navigation links for Amazon RDS, Dashboard, Databases, Query Editor, Performance insights, Snapshots, Exports in Amazon S3, Automated backups, Reserved instances, Proxies, Subnet groups (selected), Parameter groups, Option groups, Custom engine versions, Events, Event subscriptions, Recommendations, and Certificate update. The top navigation bar shows the AWS logo, Services, a search bar, and user information. The main content area is titled 'Add subnets' and includes the following sections:

- Description:** A text field containing 'project-db-subnet-group'.
- VPC:** A dropdown menu showing 'project-vpc (vpc-0060554ac37d5399f)'.
- Availability Zones:** A section titled 'Choose the Availability Zones that include the subnets you want to add.' with a dropdown menu showing 'us-east-1a' and 'us-east-1b' selected.
- Subnets:** A section titled 'Choose the subnets that you want to add. The list includes the subnets in the selected Availability Zones.' with a dropdown menu showing 'subnet-070907a87d6e44469 (10.0.4.0/23)' and 'subnet-064e9f8a92dca9f5f (10.0.2.0/23)' selected.
- Subnets selected (2):** A table listing the selected subnets.

Availability zone	Subnet ID	CIDR block
us-east-1a	subnet-064e9f8a92dca9f5f	10.0.2.0/23
us-east-1b	subnet-070907a87d6e44469	10.0.4.0/23

At the bottom of the form, there are 'Cancel' and 'Create' buttons.

b. L'instance RDS

Création d'une instance RDS avec Engine MYSQL sous le nom de "project-db"

Create database

Choose a database creation method [Info](#)

☒ **Standard create**
You set all of the configuration options, including ones for availability, security, backup, 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](#)

☐ Amazon Aurora

☒ **MySQL**

☐ MariaDB

☐ PostgreSQL

☐ Oracle

☐ Microsoft SQL Server

Edition

☒ **MySQL Community**

Known issues/limitations
Review the [Known issues/limitations](#) to learn about potential compatibility issues with specific database versions.

▼ Hide filters

☒ Show versions that support the Multi-AZ DB cluster [Info](#)

MySQL 8.0.28

Templates
Choose a sample template to meet your use case.

☐ **Production**
Use defaults for high availability and fast, consistent performance.

☐ **Dev/Test**
This instance is intended for development use outside of a production environment.

☒ **Free tier**
Use RDS Free Tier to develop new applications, test existing applications, or gain hands-on experience with Amazon RDS. [Info](#)

Availability and durability

Deployment options [Info](#)
The deployment options below are limited to those supported by the engine you selected above.

☒ **Multi-AZ DB Cluster - new**
Creates a DB cluster with a primary DB instance and two readable standby DB instances, with each DB instance in a different Availability Zone (AZ). Provides high availability, data redundancy and increases capacity to serve read workloads.

☐ **Multi-AZ DB instance (not supported for Multi-AZ DB cluster snapshot)**
Creates a primary DB instance and a standby DB instance in a different AZ. Provides high availability and data redundancy, but the standby DB instance doesn't support connections for read workload.

☐ **Single DB instance (not supported for Multi-AZ DB cluster snapshot)**
Creates a single DB instance with no standby DB instances.

Settings

DB instance identifier [Info](#)
Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.

project-db

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 63 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

▼ **Credentials Settings**

Master username [Info](#)
Type a login ID for the master user of your DB instance.

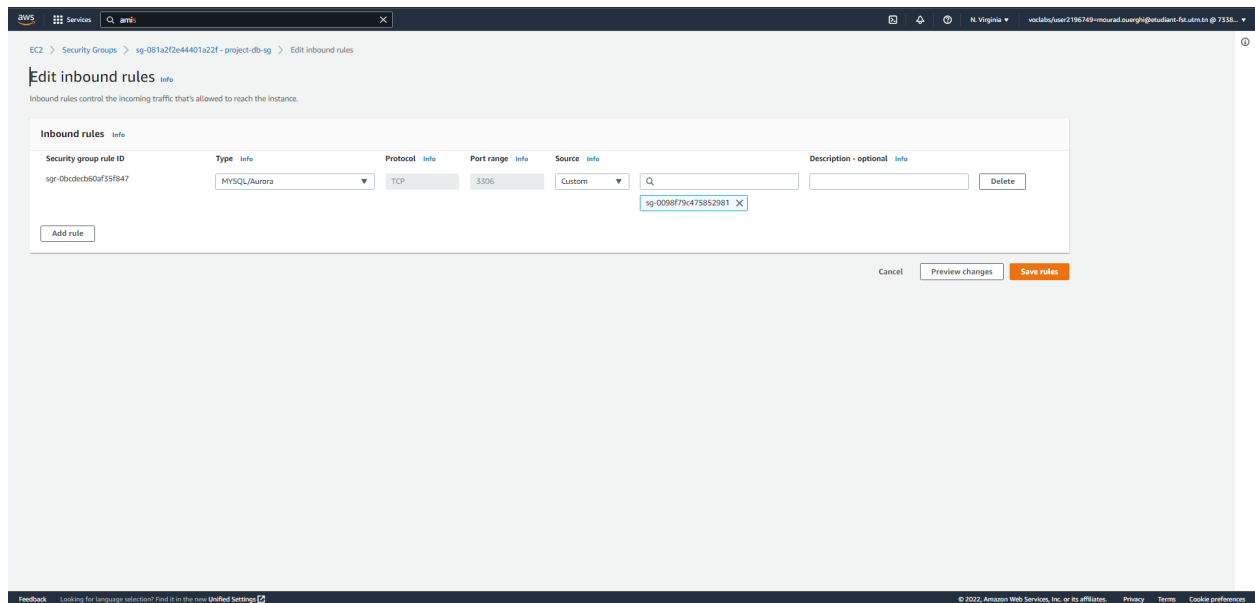
mourdouerghi

1 to 16 alphanumeric characters. First character must be a letter.

☐ Manage master credentials in AWS Secrets Manager

c. Security Group

C'est le security group du "projet-db" après avoir terminé le projet, contenant une seule inbound rule qui permet aux instances du web server d'accéder à la base de données rds



7. Création des instances EC2

a. Création du machine bastion

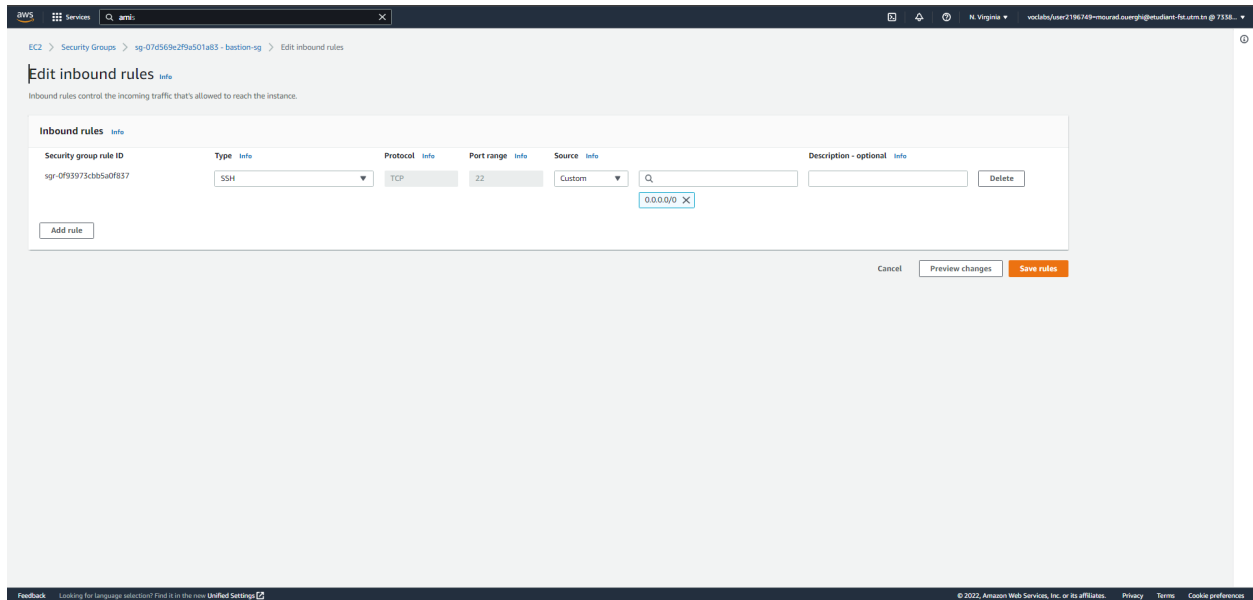
i. L'instance Bastion

Création d'une machine bastion basé sur Amazon Linux de type t2.micro dans le public subnet "project-public-subnet-1" avec une clé ssh que j'ai téléchargé pour pouvoir accéder à la machine ensuite

The screenshot displays the AWS Management Console interface for creating a new EC2 instance. The 'Network settings' section is expanded, showing the VPC (vpc-0060554dc37d5399f) and Subnet (subnet-06d4b0c11ca89db0c1) selected. The 'Auto-assign public IP' option is set to 'Enable'. The 'Firewall (security group)' section is also expanded, showing the 'Create security group' option selected. The 'Summary' section on the right shows the instance configuration: 1 instance, Amazon Linux 2 Kernel 5.10 AML, t2.micro instance type, new security group, and 1 volume (8 GiB). A 'Free tier' notice is displayed, indicating that the first year includes 750 hours of t2.micro usage. The 'Launch Instance' button is visible at the bottom right.

ii. Security Group

Le security group du machine bastion ne permet l'accès qu'à travers ssh de n'importe quelle source



b. Création du machine Web Server

i. L'instance Web Server

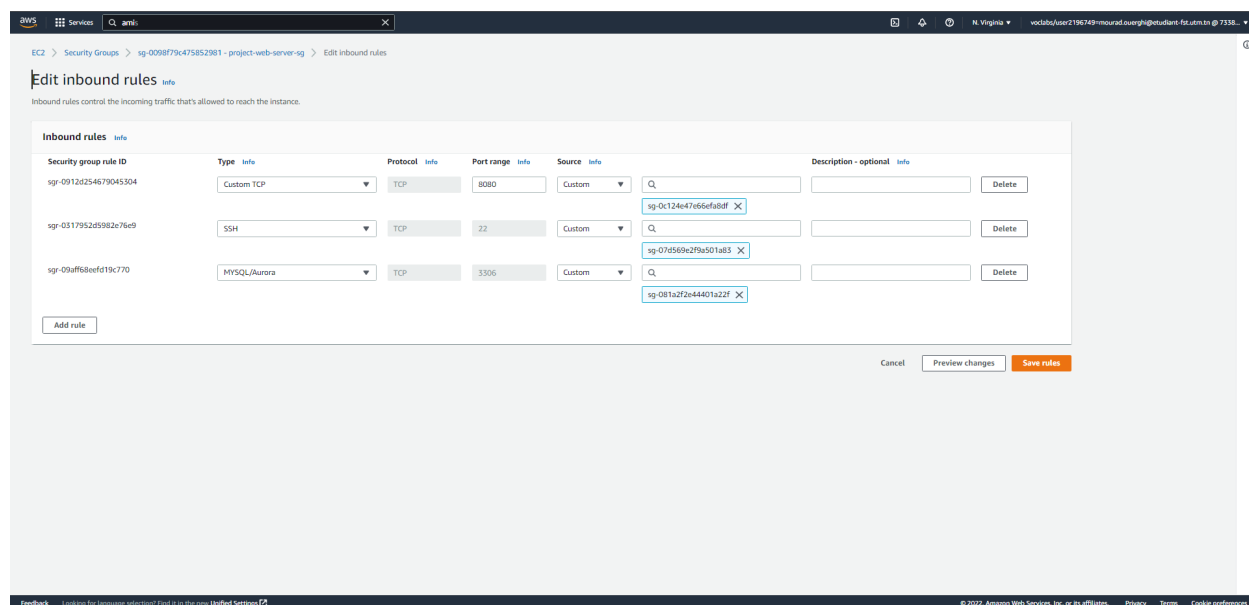
C'est presque les même configurations du machine bastion mais dans une private subnet

The screenshot shows the AWS 'Launch Instance' wizard. The 'Instance type' is set to 't2.micro'. The 'Key pair (login)' is 'bastion-key-pair'. The 'Network settings' show the instance is being launched in a private subnet ('project-private-subnet-1') with 'Auto-assign public IP' disabled. The 'Firewall (security groups)' section has 'Create security group' selected. The 'Summary' panel on the right shows the instance configuration: 1 instance, Amazon Linux 2 Kernel 5.10 AMI, t2.micro instance type, New security group, and 1 volume (8 GiB). A 'Free tier' alert is visible, indicating that the first year includes 750 hours of t2.micro usage.

This screenshot shows the 'Configure storage' and 'Firewall (security groups)' sections of the AWS 'Launch Instance' wizard. The 'Firewall (security groups)' section shows a new security group named 'project-web-server-sg' with the description 'project-web-server-sg'. The 'Inbound security groups rules' section shows a rule for 'ssh' on port 22, allowing traffic from 'Custom' source type. The 'Configure storage' section shows a single volume of 8 GiB, gp2 type, as the root volume. The 'Summary' panel on the right remains the same as in the previous screenshot.

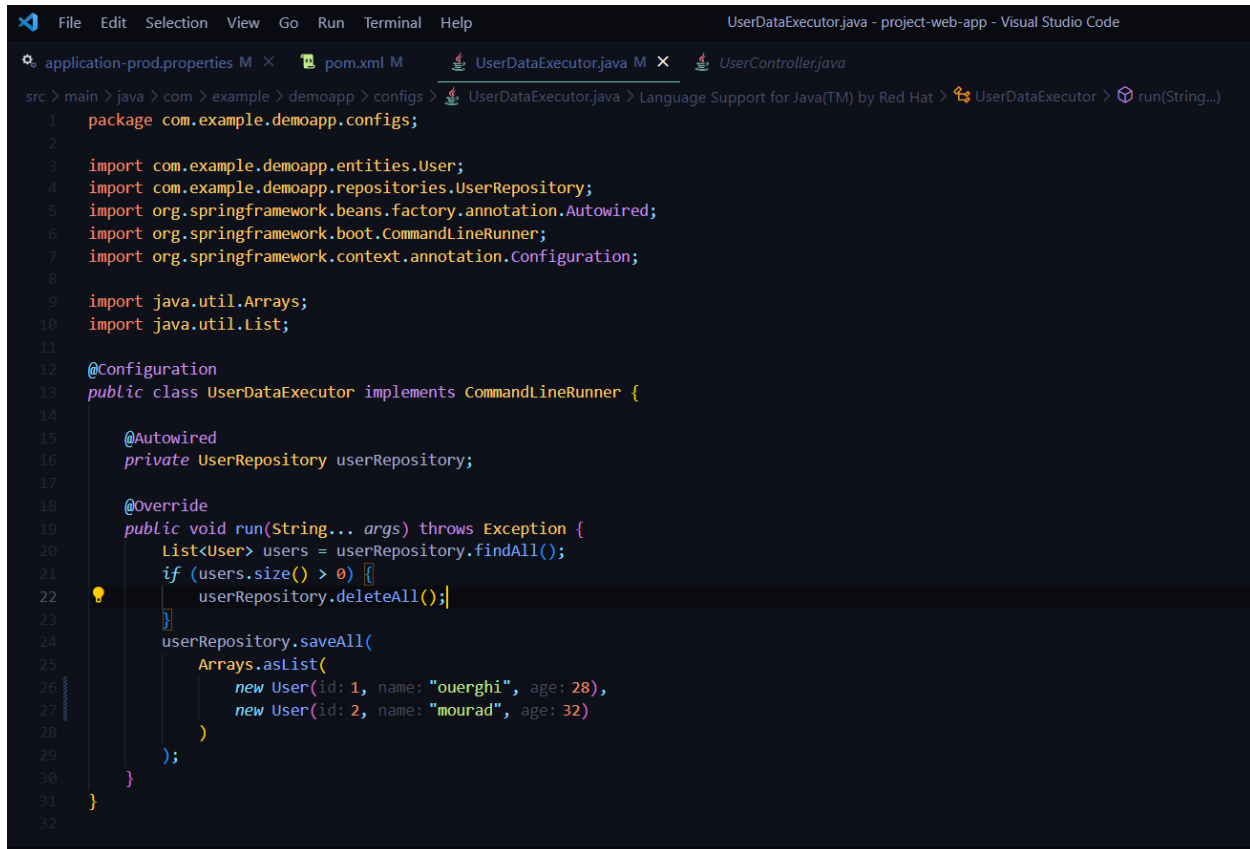
ii. Security Group

C'est le security group du "projet-web-server" après avoir terminé le projet, contenant une trois inbound rules, la première sur le port 8080 en TCP avec le security group du load Balancer pour permettre au load balancer d'envoyer des health checks, la deuxième sur le port 22 avec ssh avec le security group du machine bastion pour permettre l'accès au machine bastion et la troisième sur le port 3306 en TCP avec le security group de l'instance rds pour recevoir les données transmises



8. Déploiement de l'application web

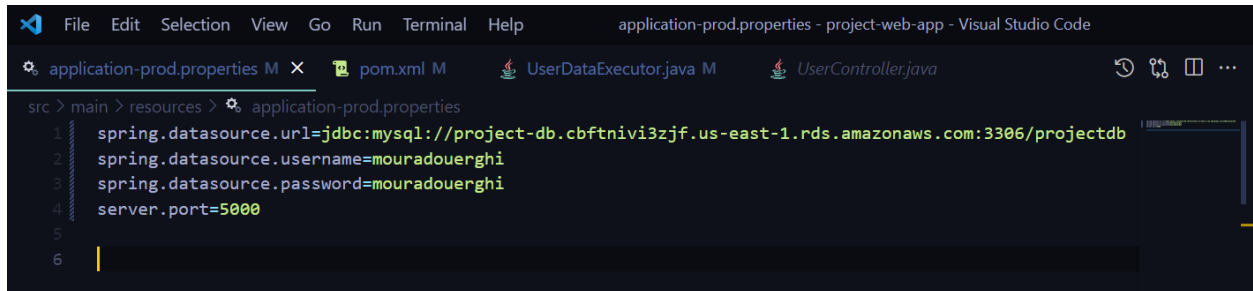
C'est une simple application contenant une seul endpoint “/api/users”
qui crée deux utilisateurs dans la base de données dès qu'elle se lance



```
1 package com.example.demoapp.configs;
2
3 import com.example.demoapp.entities.User;
4 import com.example.demoapp.repositories.UserRepository;
5 import org.springframework.beans.factory.annotation.Autowired;
6 import org.springframework.boot.CommandLineRunner;
7 import org.springframework.context.annotation.Configuration;
8
9 import java.util.Arrays;
10 import java.util.List;
11
12 @Configuration
13 public class UserDataExecutor implements CommandLineRunner {
14
15     @Autowired
16     private UserRepository userRepository;
17
18     @Override
19     public void run(String... args) throws Exception {
20         List<User> users = userRepository.findAll();
21         if (users.size() > 0) {
22             userRepository.deleteAll();
23         }
24         userRepository.saveAll(
25             Arrays.asList(
26                 new User(id: 1, name: "ouerghi", age: 28),
27                 new User(id: 2, name: "mourad", age: 32)
28             )
29         );
30     }
31 }
32
```

a. Liaison de l'application web avec l'instance RDS

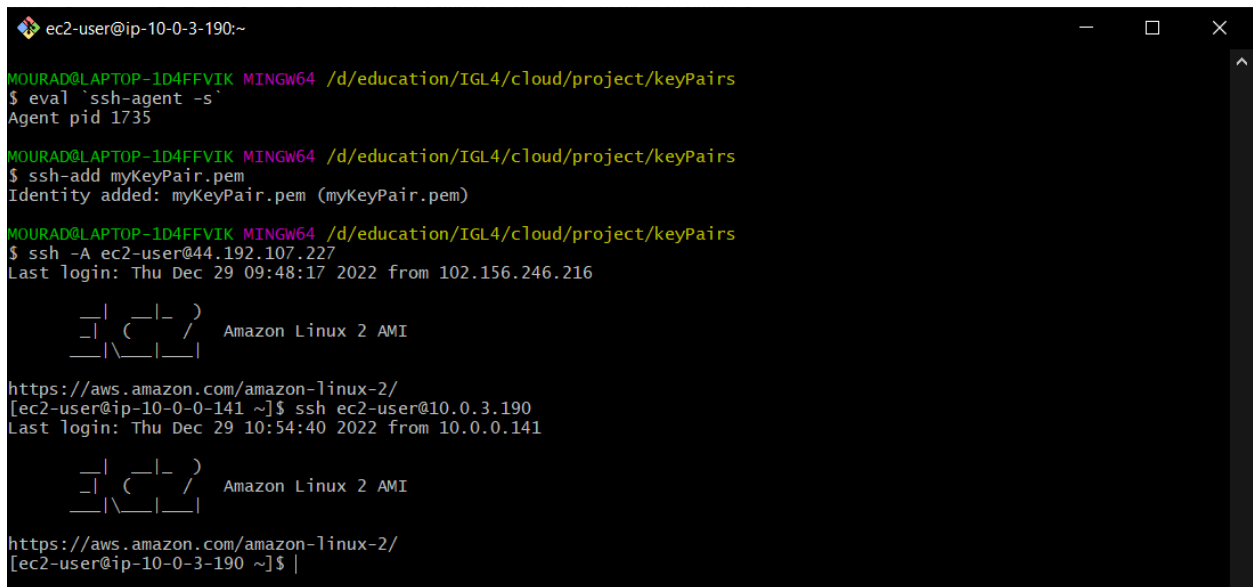
Pour lier l'application avec la base de données rds créé j'ai mis l'url de l'instance rds le nom d'utilisateur et le mot de passe lorsque j'ai créé l'instance rds

A screenshot of the Visual Studio Code editor interface. The title bar shows 'application-prod.properties - project-web-app - Visual Studio Code'. The file explorer on the left shows the project structure: 'src > main > resources > application-prod.properties'. The editor displays the content of 'application-prod.properties' with the following properties:

```
spring.datasource.url=jdbc:mysql://project-db.cbftnivi3zjf.us-east-1.rds.amazonaws.com:3306/projectdb
spring.datasource.username=mouradouwerghi
spring.datasource.password=mouradouwerghi
server.port=5000
```

b. Installation de l'environnement nécessaire

Pour se connecter à la machine bastion j'ai utilisé Git Bash, ensuite j'ai exécuter les commandes suivantes pour avoir l'accès ssh à la machine.

A screenshot of a terminal window showing a series of commands and their outputs. The terminal title is 'ec2-user@ip-10-0-3-190:~'. The commands and outputs are:

```
MOURAD@LAPTOP-1D4FFVIK MINGW64 /d/education/IGL4/cloud/project/keyPairs
$ eval `ssh-agent -s`
Agent pid 1735

MOURAD@LAPTOP-1D4FFVIK MINGW64 /d/education/IGL4/cloud/project/keyPairs
$ ssh-add myKeyPair.pem
Identity added: myKeyPair.pem (myKeyPair.pem)

MOURAD@LAPTOP-1D4FFVIK MINGW64 /d/education/IGL4/cloud/project/keyPairs
$ ssh -A ec2-user@44.192.107.227
Last login: Thu Dec 29 09:48:17 2022 from 102.156.246.216

  _| _| _| _|
  _| ( _| _| _| Amazon Linux 2 AMI
  _| \ _| _| _|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-10-0-0-141 ~]$ ssh ec2-user@10.0.3.190
Last login: Thu Dec 29 10:54:40 2022 from 10.0.0.141

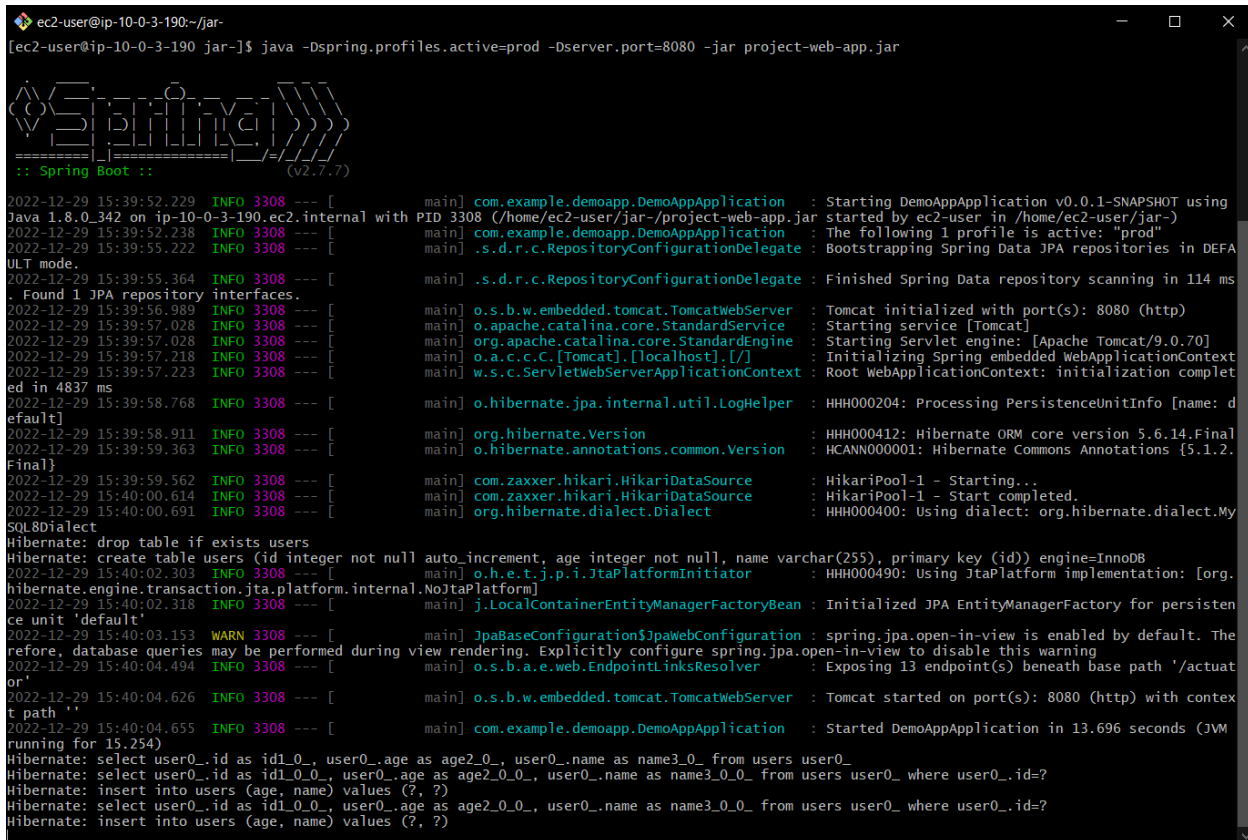
  _| _| _| _|
  _| ( _| _| _| Amazon Linux 2 AMI
  _| \ _| _| _|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-10-0-3-190 ~]$
```

Ensuite j'ai installé git en utilisant la commande **"sudo yum install git"** puisque j'ai mis le fichier ".jar" dans une repository dans github pour faciliter la déploiement, ensuite j'ai installé jdk en utilisant la commande **"sudo yum install java-1.8.0-openjdk"** ensuite j'ai fait un **"git clone"** du fichier ".jar". J'ai oublié de faire des captures pour ces étapes donc j'ai essayé de faire une description détaillé

c. Déploiement de l'application web dans l'instance EC2

Comme le montre la figure ci dessous l'application se lance correctement et se connecte avec l'instance rds et crée les deux utilisateurs



```
ec2-user@ip-10-0-3-190:~/jar-
[ec2-user@ip-10-0-3-190 jar-]$ java -Dspring.profiles.active=prod -Dserver.port=8080 -jar project-web-app.jar

:: Spring Boot ::
(v2.7.7)

2022-12-29 15:39:52.229 INFO 3308 --- [main] com.example.demoapp.DemoAppApplication : Starting DemoAppApplication v0.0.1-SNAPSHOT using
Java 1.8.0_342 on ip-10-0-3-190.ec2.internal with PID 3308 (/home/ec2-user/jar-/project-web-app.jar
2022-12-29 15:39:52.238 INFO 3308 --- [main] com.example.demoapp.DemoAppApplication : The following 1 profile is active: "prod"
2022-12-29 15:39:55.222 INFO 3308 --- [main] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repositories in DEFA
ULT mode.
2022-12-29 15:39:55.364 INFO 3308 --- [main] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 114 ms
. Found 1 JPA repository interfaces.
2022-12-29 15:39:56.989 INFO 3308 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8080 (http)
2022-12-29 15:39:57.028 INFO 3308 --- [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2022-12-29 15:39:57.028 INFO 3308 --- [main] org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/9.0.70]
2022-12-29 15:39:57.218 INFO 3308 --- [main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2022-12-29 15:39:57.223 INFO 3308 --- [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization complet
ed in 4837 ms
2022-12-29 15:39:58.768 INFO 3308 --- [main] o.hibernate.jpa.internal.util.LogHelper : HHH000204: Processing PersistenceUnitInfo [name: d
efault]
2022-12-29 15:39:58.911 INFO 3308 --- [main] org.hibernate.Version : HHH000412: Hibernate ORM core version 5.6.14.Final
2022-12-29 15:39:59.363 INFO 3308 --- [main] o.hibernate.annotations.common.Version : HCANN000001: Hibernate Commons Annotations {5.1.2.
Final}
2022-12-29 15:39:59.562 INFO 3308 --- [main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Starting...
2022-12-29 15:40:00.614 INFO 3308 --- [main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Start completed.
2022-12-29 15:40:00.691 INFO 3308 --- [main] org.hibernate.dialect.Dialect : HHH000400: Using dialect: org.hibernate.dialect.My
SQL8Dialect
Hibernate: drop table if exists users
Hibernate: create table users (id integer not null auto_increment, age integer not null, name varchar(255), primary key (id)) engine=InnoDB
2022-12-29 15:40:02.303 INFO 3308 --- [main] o.h.e.t.j.p.i.JtaPlatformInitiator : HHH000490: Using JtaPlatform implementation: [org.
hibernate.engine.transaction.jta.platform.internal.NoJtaPlatform]
2022-12-29 15:40:02.318 INFO 3308 --- [main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persisten
ce unit 'default'
2022-12-29 15:40:03.153 WARN 3308 --- [main] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view is enabled by default. The
refore, database queries may be performed during view rendering. Explicitly configure spring.jpa.open-in-view to disable this warning
2022-12-29 15:40:04.494 INFO 3308 --- [main] o.s.b.a.e.web.EndpointLinksResolver : Exposing 13 endpoint(s) beneath base path '/actuat
or'
2022-12-29 15:40:04.626 INFO 3308 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http) with contex
t path ''
2022-12-29 15:40:04.655 INFO 3308 --- [main] com.example.demoapp.DemoAppApplication : Started DemoAppApplication in 13.696 seconds (JVM
running for 15.254)
Hibernate: select user0_.id as id1_0_, user0_.age as age2_0_, user0_.name as name3_0_ from users user0_
Hibernate: select user0_.id as id1_0_0_, user0_.age as age2_0_0_, user0_.name as name3_0_0_ from users user0_ where user0_.id=?
Hibernate: insert into users (age, name) values (?, ?)
Hibernate: select user0_.id as id1_0_0_, user0_.age as age2_0_0_, user0_.name as name3_0_0_ from users user0_ where user0_.id=?
Hibernate: insert into users (age, name) values (?, ?)
```

9. Création du load balancer

a. Création du Target Group

Création d'un target Group et enregistrement de l'instance "project-web-server"

(J'ai oublié de prendre une capture lorsque j'ai créé le target donc j'ai pris une capture après avoir terminer le projet)

Register targets

Select instances, specify ports, and add the instances to the list of pending targets. Repeat to add additional combinations of instances and ports to the list of pending targets. Once you are satisfied with your selections, click Register pending targets.

Available instances (3)

Instance ID	Name	State	Security groups	Zone	IPv4 address	Subnet ID
i-0b28f188b3b0b3f70	project-web-server	running	project-web-server-sg	us-east-1a		subnet-064c9f8a2daca5f
i-0ff9a3f0d4cc3b0ee	project-web-server	running	project-web-server-sg	us-east-1b		subnet-070907a87d6e44469
i-0ca060c54bca32b09	project-bastion	running	bastion-sg	us-east-1a	44.192.107.227	subnet-064b4bc511ca89abc1

0 selected

Ports for the selected instances

Port for routing traffic to the selected instances:

8080

1-65535 (separate multiple ports with commas)

Include as pending below

Review targets

Targets (2)

Remove	Health status	Instance ID	Name	Port	State	Security groups	Zone	IPv4 address	Subnet ID
	healthy	i-0b28f188b3b0b3f70	project-web-server	8080	running	project-web-server-sg	us-east-1a		subnet-064c9f8a2daca5f
	healthy	i-0ff9a3f0d4cc3b0ee	project-web-server	8080	running	project-web-server-sg	us-east-1b		subnet-070907a87d6e44469

0 pending

b. LoadBalancer

Création d'un load balancer sous le nom de “project-lb” sur les deux public subnets des deux availability zones

The screenshot displays the AWS Management Console interface for an Elastic Load Balancing (ELB) instance named "project-lb". The console is in the "us-east-1" region, and the user is logged in as "victor@user1190749-mcraed-courgh@us-east-1.amazonaws.com".

Details:

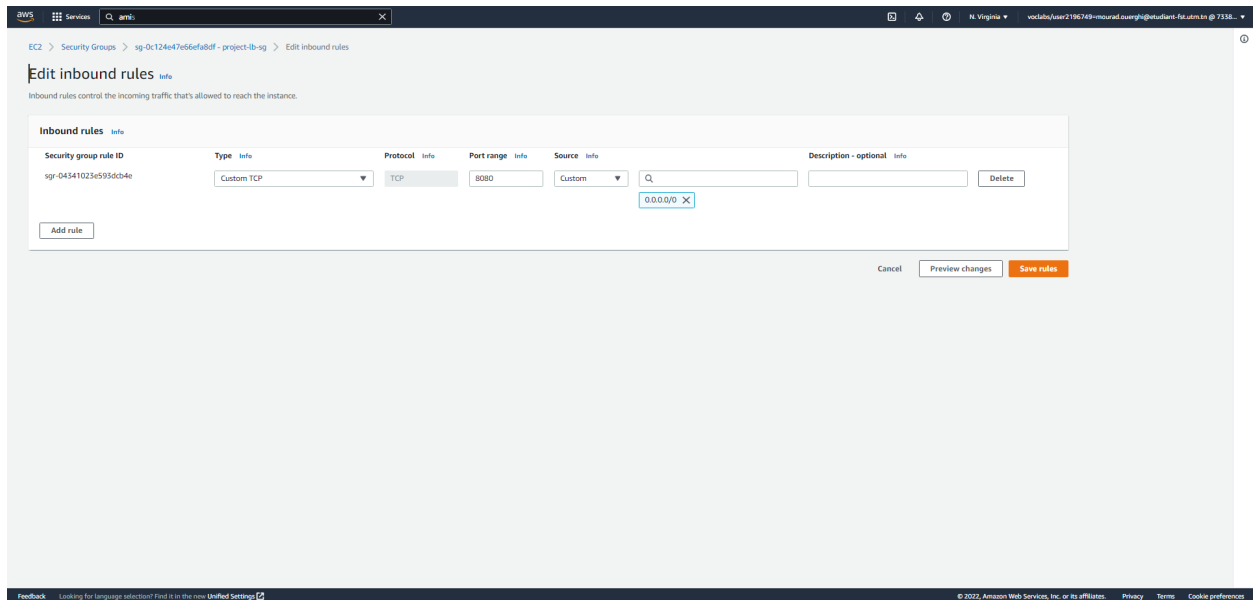
- Load balancer type:** Application
- Application:** project-lb-1608128505-us-east-1.elb.amazonaws.com (A Record)
- IP address type:** IPv4
- Scheme:** Internet-facing
- Status:** Active
- VPC:** vpc-0060554cc37d5399f
- Availability Zones:** subnet-0644bc6d11ca89d9c1 (us-east-1a [use1-az1]), subnet-00e911318e33b5863 (us-east-1b [use1-az2])
- Hosted Zone:** Z355XDOTRQ7X7K
- Created At:** December 29, 2022, 10:45 (UTC+01:00)

Listeners (1):

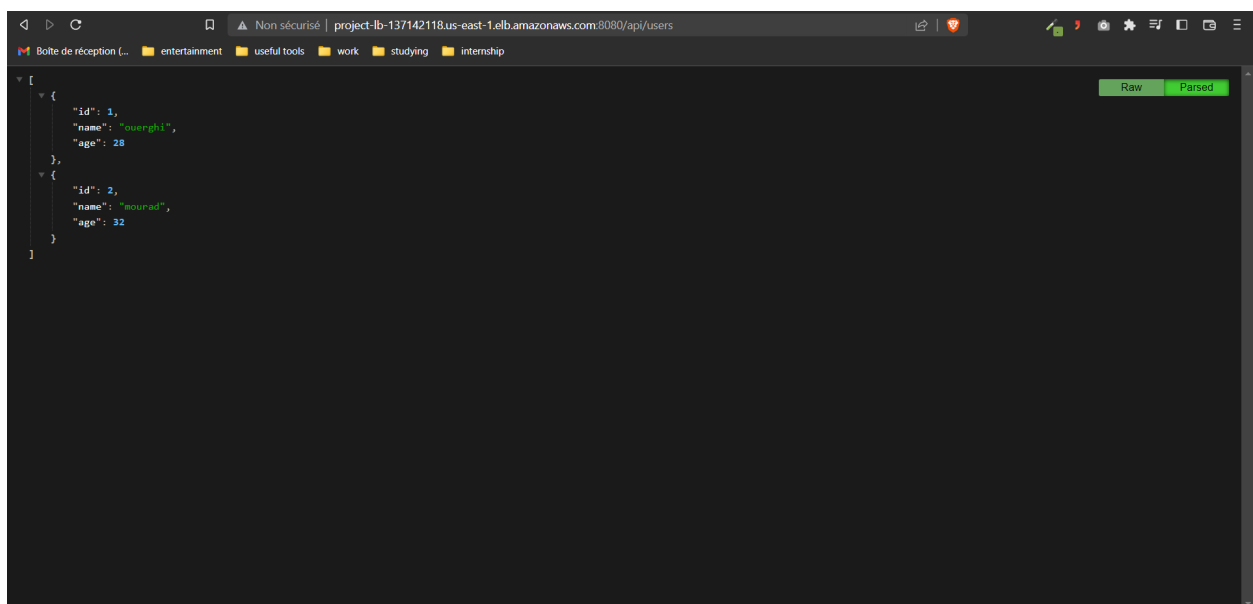
Protocol-Port	ARN	Security policy	Default SSL cert	Default routing rule	Rules	Tags
HTTP:8080	ARN	Not Applicable	Not Applicable	1. Forward to <ul style="list-style-type: none">project-target-group (1 (100%))Group-level stickiness: Off	1	0

c. Security Group

Pour le security group j'ai ouvert seulement le port 8080 puisque l'application se lance sur le port 8080



L'application est accessible comme le montre la figure ci dessous



10. Création du AutoScaling Group

a. Création du Launch Configuration

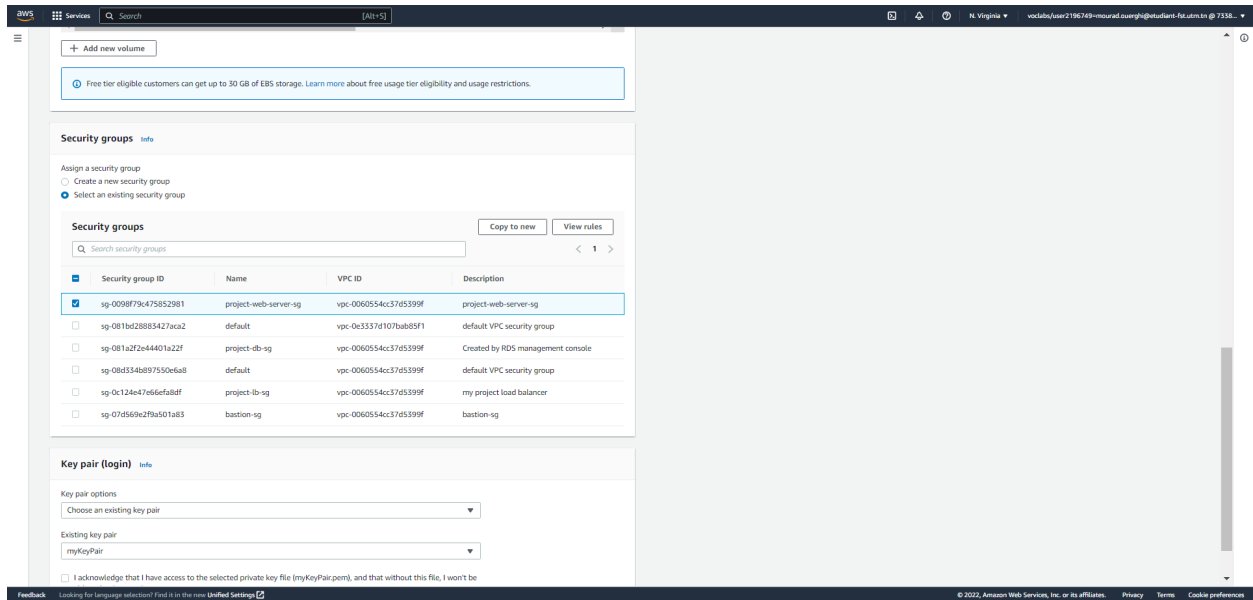
i. Création d'une AMI a partir du web server

The screenshot shows the 'Create image' page in the AWS Management Console. The page is for creating an Amazon Machine Image (AMI) from an existing EC2 instance. The 'Instance ID' is 'i-0b28f188b3b0b3f70' (project-web-server). The 'Image name' is 'project-web-server-image'. The 'Image description - optional' is 'project-web-server-image'. The 'No reboot' checkbox is checked. The 'Instance volumes' section shows a table with columns: Volume type, Device, Snapshot, Size, Volume type, IOPS, Throughput, Delete on termination, and Encrypted. The table has one row: EBS, /dev/sda1, Create new snapshot fr..., 8, EBS General Purpose S..., 100, and Delete on termination is checked. The 'Tags - optional' section has two radio buttons: 'Tag image and snapshots together' (selected) and 'Tag image and snapshots separately'. The 'Add volume' button is visible. The 'Create image' button is at the bottom right.

ii. Création du Launch Configuration

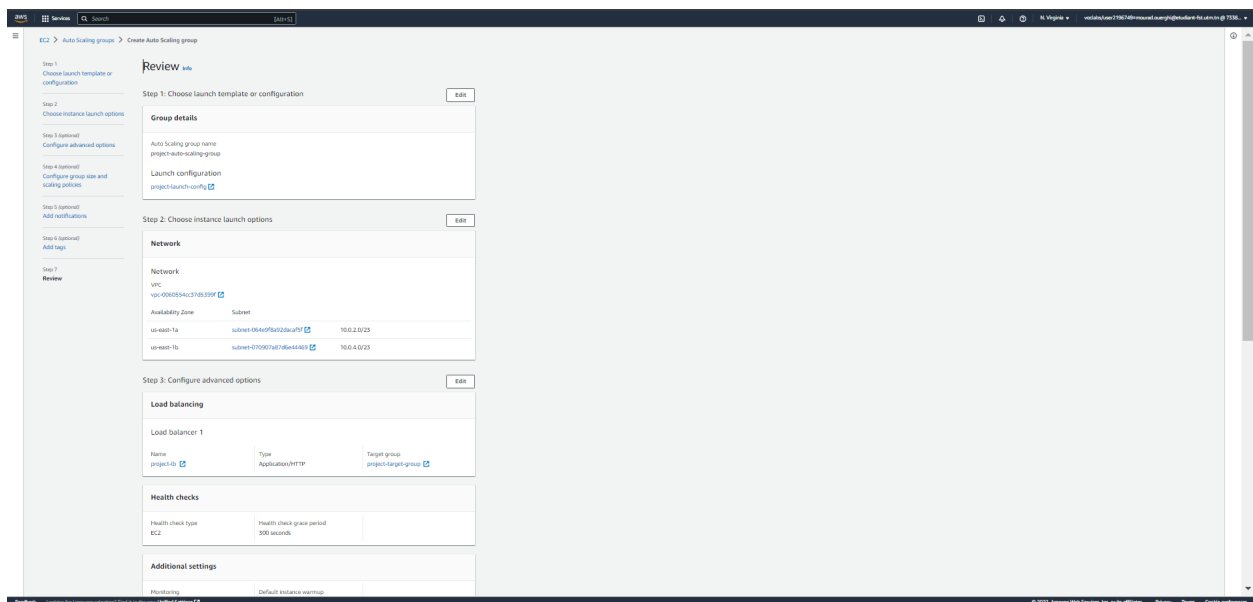
Création d'un launch config sous le nom de “project-launch-configuration” en utilisant l'image du “project-web-server” créé

The screenshot shows the 'Create launch template' page in the AWS Management Console. The page is for creating a launch configuration. The 'Launch configuration name' is 'project-launch-config'. The 'Amazon machine image (AMI)' is 'project-web-server-image'. The 'Instance type' is 't2.micro (1 vCPU, 1 GiB, EBS Only)'. The 'Additional configuration - optional' section has checkboxes for 'Purchasing options' (Request Spot Instances), 'IAM instance profile' (Select IAM role), 'Monitoring' (Enable EC2 instance detailed monitoring within CloudWatch), 'EBS-optimized instance', and 'Launch as EBS-optimized instance'. The 'Create launch template' button is at the top right.



b. Création du AutoScaling Group

Création d'un auto scaling group sur les deux private subnets en utilisant le "project-launch-configuration" avec une desired capacity de 2 instances (je l'ai diminuer ensuite vers 1 instance puisque l'instance "project-web-server est déjà lancé")



Health checks

Health check type	EC2
Health check grace period	300 seconds

Additional settings

Monitoring	Disabled
Default instance warmup	Disabled

Step 4: Configure group size and scaling policies [Edit](#)

Group size

Desired capacity	2
Minimum capacity	1
Maximum capacity	2

Scaling policy

No scaling policy

Instance scale-in protection

Instance scale-in protection	<input checked="" type="checkbox"/> Enables instance protection from scale in
------------------------------	---

Step 5: Add notifications [Edit](#)

Notifications

No notifications

Step 6: Add tags [Edit](#)

Tags (0)

Key	Value	Tag new instances
No tags		

[Cancel](#) [Create Auto Scaling group](#)

La figure suivante montre que les deux target sont “healthy” donc le load balancer peut distribuer la charge entre les deux instances d'où la haute disponibilité de l'application

Target groups (1/1) [Info](#)

<input checked="" type="checkbox"/>	Name	ARN	Port	Protocol	Target type	Load balancer	VPC ID
<input checked="" type="checkbox"/>	project-target-group	arn:aws:elasticloadbalancing...	8080	HTTP	Instance	project-lb	vpc-0060554cc3745399f

Target group: project-target-group

Registered targets (2)

<input type="checkbox"/>	Instance ID	Name	Port	Zone	Health status	Health status details
<input type="checkbox"/>	i-0b28f188b3b0b3f70	project-web-server	8080	us-east-1a	healthy	
<input type="checkbox"/>	i-0f9a3f004cc3b0ee		8080	us-east-1b	healthy	