



République Tunisienne
Ministère de l'Enseignement Supérieur
et de la Recherche Scientifique
Université tunis el manar
Faculté des Sciences Mathématiques, Physiques et
Naturelles de Tunis



RAPPORT DE PROJET

Projet AWS

Par
JBELI MOHAMED

Année Universitaire: 2022-2023

Contents

1	Le cadre générale de projet	1
1.1	Introduction	1
1.2	Présentation du projet	1
1.3	Le travail demandé	1
1.4	Conclusion	10

List of Figures

1.1	Nouveau VPC	2
1.2	Tables de routages associées au VPC	2
1.3	détails de table de routage aws-project-2-rtb-private1-us-east-1c	2
1.4	détails de table de routage aws-project-2-rtb-public	2
1.5	les détails des SUBNETS	3
1.6	détails du NAT GATEWAY	3
1.7	les details du security groupe pour les machines EC2 publiques	3
1.8	les details du Load Balancer	3
1.9	les détails du TARGET GROUP	4
1.10	'accès à la machine EC2 bastion à travers le Secure Shell Protocol (SSH)	4
1.11	La création d'un nouveau template pour configurer l'auto scale group	4
1.12	les détails du auto scaling group	5
1.13	L'ensemble des règles du security groupe attachées à la machine EC2 hébergée dans le sousréseaux privé	5
1.14	L'accès à la machine EC2 backend à travers le Secure Shell Protocol (SSH)	6
1.15	La configuration du security group de la base de données MYSQL	6
1.16	La configuration du fichier docker pour la partie backend	7
1.17	La nouvelle configuration MYSQL pour la partie backend	7
1.18	La publication de notre image associée au projet backend dans le Docker repository	8
1.19	L'hébergement de la solution backend dans la machine EC2 privée	8
1.20	le déploiement	9
1.21	la configuration du fichier docker pour la partie frontend	9
1.22	La création de la machine EC2 publique pour héberger la partie fontend	9
1.23	La création d'un target group pour lier load balancer applicative et la machine EC2	10
1.24	les détails du security group de la machine publique	10
1.25	La liaison de la connexion entre le load balancer applicative et la machine EC2	10
1.26	la connexion du base de données	11
1.27	la création des tables	11

Chapter 1

Le cadre générale de projet

1.1 Introduction

Dans ce chapitre, nous allons présenter en premier lieu le sujet de travail proposé tout en expliquant ses objectifs ainsi les étapes nécessaires pour le réaliser.

1.2 Présentation du projet

Le cloud computing est à la demande de ressources informatiques sur Internet avec une tarification à l'utilisation. Au lieu d'acheter, de posséder et d'entretenir des centres de données et des serveurs physiques, vous pouvez accéder à des services technologiques, tels que la puissance de calcul, le stockage et les bases de données, selon vos besoins auprès d'un fournisseur de cloud comme Amazon Web Services (AWS). Il fournit un environnement complet de solutions qui vous aident à résoudre les problèmes courants et à créer plus rapidement. Chaque solution AWS est fournie avec une architecture détaillée, un guide de déploiement et des instructions pour le déploiement automatisé et manuel.

1.3 Le travail demandé

L'objectif du projet est de déployer une application web hautement disponible (HA) sur AWS.

Nous avons choisi Docker car il nous permet de créer, tester et déployer rapidement des applications.

- Voici l'imprime d'écran qui montre les détails du nouveau VPC créé. Son IP adresse est 10.0.0.0/16 (voir figure 1.1).

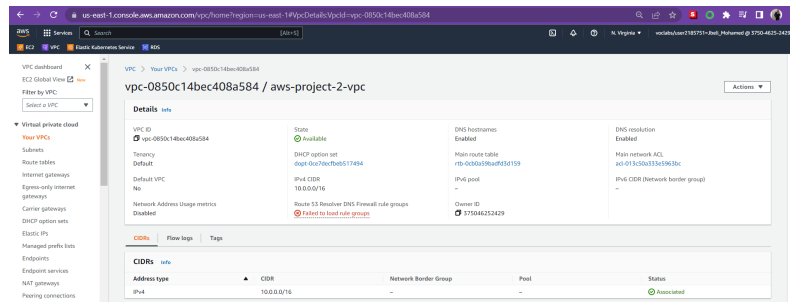


Figure 1.1: Nouveau VPC

- Voici l'imprime d'écran qui montre les tables de routages associées au VPC (voir figure 1.6).

<input type="checkbox"/>	aws-project-2-rtb-private1-us-east-1c	rtb-0b85d41da67e2971b	subnet-0c40e10e64ce3...	-	No	vpc-0850c14bec408a584 aw...	375046252429
<input type="checkbox"/>	aws-project-2-rtb-private2-us-east-1d	rtb-01a9362428df5cae5	subnet-0f57c05ad3958...	-	No	vpc-0850c14bec408a584 aw...	375046252429
<input type="checkbox"/>	aws-project-2-rtb-public	rtb-0ad60fed15dd3c437	2 subnets	-	No	vpc-0850c14bec408a584 aw...	375046252429

Figure 1.2: Tables de routages associées au VPC

- Voici l'imprime d'écran qui montre les détails de table de routage aws-project-2-rtb-private1-us-east-1c

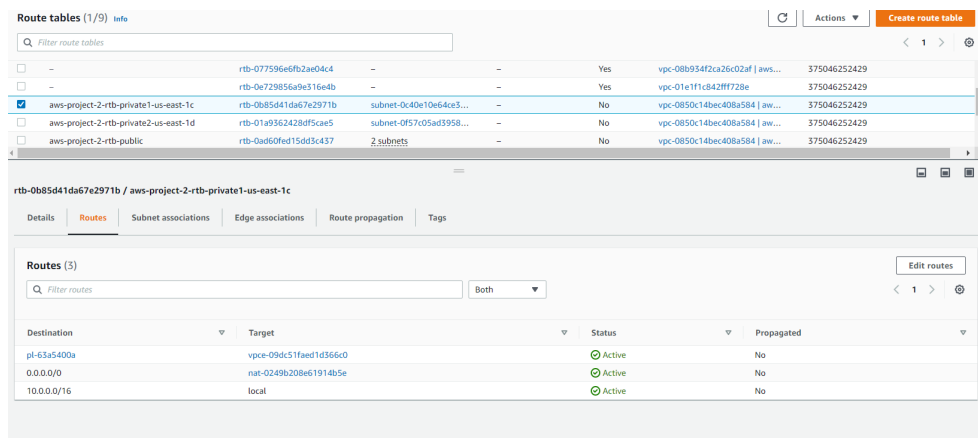


Figure 1.3: détails de table de routage aws-project-2-rtb-private1-us-east-1c

- Voici l'imprime d'écran qui montre les détails de table de routage aws-project-2-rtb-public(voir figure 1.4).

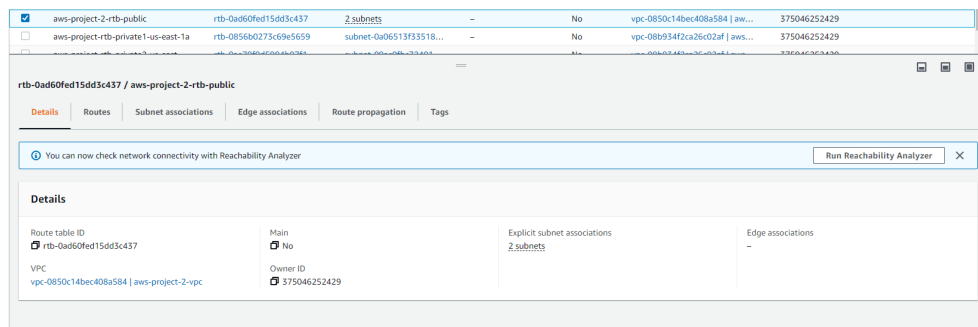


Figure 1.4: détails de table de routage aws-project-2-rtb-public

- Voici l'imprime d'écran qui montre les détails des SUBNETS (voir figure 1.5).

<input type="checkbox"/>	aws-project-2-subnet-private-1-us-east-1c	subnet-0c40e10e64ce3071d	Available	vpc-0850c14bec408a584 aw...	10.0.2.0/23	-	505
<input type="checkbox"/>	aws-project-2-subnet-private-2-us-east-1d	subnet-0f57c05ad39585255	Available	vpc-0850c14bec408a584 aw...	10.0.4.0/23	-	507
<input type="checkbox"/>	aws-project-2-subnet-public-1-us-east-1c	subnet-039a20d764e3a7d6c	Available	vpc-0850c14bec408a584 aw...	10.0.0.0/24	-	248
<input type="checkbox"/>	aws-project-2-subnet-public-2-us-east-1d	subnet-0669ee48f1ded56c	Available	vpc-0850c14bec408a584 aw...	10.0.1.0/24	-	250

Figure 1.5: les détails des SUBNETS

- Voici l'imprime d'écran qui montre les détails du NAT GATEWAY (voir figure 1.6).

<input type="checkbox"/>	Name	Internet gateway ID	State	VPC ID	Owner
<input type="checkbox"/>	aws-project-igw	igw-08b2dc467435a3baa	Attached	vpc-08b934f2ca26c02af aws-project...	375046252429
<input type="checkbox"/>	aws-project-2-igw	igw-0b5884ab53abcb24b	Attached	vpc-0850c14bec408a584 aws-project...	375046252429

Figure 1.6: détails du NAT GATEWAY

- Voici l'imprime d'écran qui montre les détails du security groupe pour les machines EC2 publiques (voir figure 1.7).

The screenshot shows the AWS Management Console interface for a security group. The left sidebar contains navigation links for various AWS services. The main content area displays the details for the security group 'sg-05680d71d5f62786c - lunch-template-security-group'. The 'Details' section shows the security group name, ID, description, VPC ID, and owner. Below this, the 'Inbound rules' section is expanded, showing a table of three rules: one for HTTP (port 80), one for SSH (port 22), and one for Custom TCP (port 8080). The 'Outbound rules' and 'Tags' sections are also visible but empty.

Figure 1.7: les détails du security groupe pour les machines EC2 publiques

- Voici l'imprime d'écran qui les détails du Load Balancer (voir figure 1.8).

The screenshot shows the AWS Management Console interface for a Load Balancer. The left sidebar contains navigation links for various AWS services. The main content area displays the details for the Load Balancer 'a3570f161a30a4251b8192e2d3ca1bfb'. The 'Basic Configuration' section shows the load balancer name, DNS name, creation time, hosted zone, status, and VPC. Below this, the 'Port Configuration' section is expanded, showing a table of one rule: one for TCP (port 80) forwarding to 31776 (TCP). The 'Security' and 'Attributes' sections are also visible but empty.

Figure 1.8: les détails du Load Balancer

- Voici l'imprime d'écran qui montre les détails du TARGET GROUP.

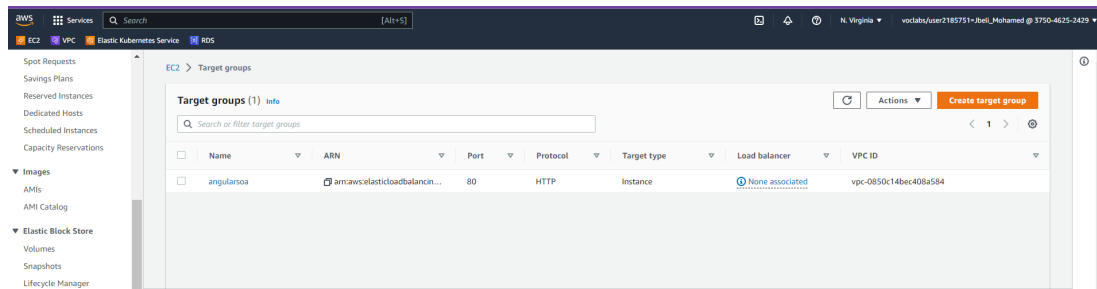


Figure 1.9: les détails du TARGET GROUP

- Voici l'imprime d'écran qui montre l'accès à la machine EC2 bastion à travers le Secure Shell Protocol (SSH) (voir figure 1.10).

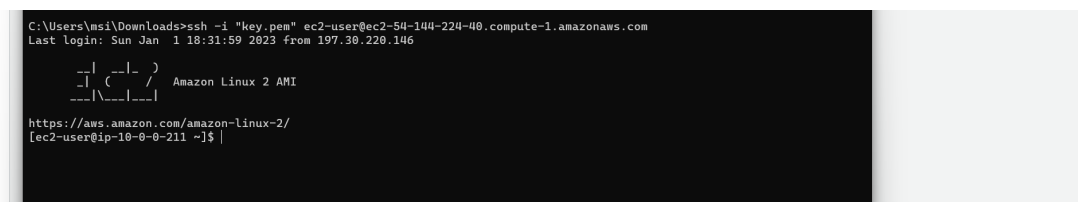


Figure 1.10: 'accès à la machine EC2 bastion à travers le Secure Shell Protocol (SSH)

- Voici l'imprime d'écran qui montre la création d'un nouveau template pour configurer l'auto scale-group. Il est un groupe de mise à l'échelle automatique contient une collection d'instances EC2 qui sont traitées comme un groupement logique. Un groupe de mise à l'échelle automatique vous permet également d'utiliser les fonctions de mise à l'échelle automatique d'Amazon EC2. Le maintien du nombre d'instances dans un groupe Auto Scaling et la mise à l'échelle automatique sont tous deux des fonctionnalités essentielles du service Amazon EC2 Auto Scaling (voir figure 1.11).



Figure 1.11: La création d'un nouveau template pour configurer l'auto scale group

- Voici l'imprime d'écran qui montre les détails du auto scaling group(voir figure 1.12).

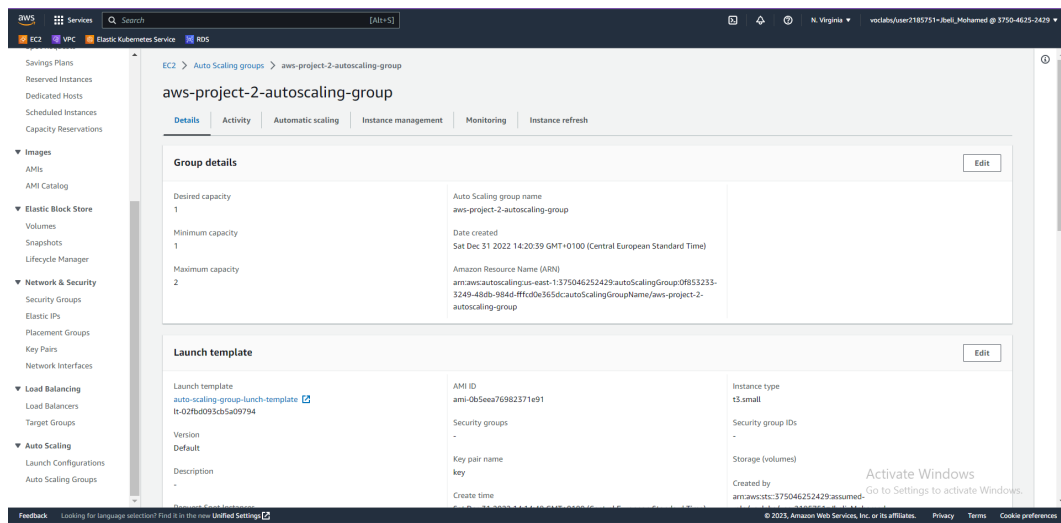


Figure 1.12: les détails du auto scaling group

- Voici l'imprime d'écran qui montre l'ensemble des règles du security groupe attachées à la machine EC2 hébergée dans le sous réseaux privé (voir figure 1.13).

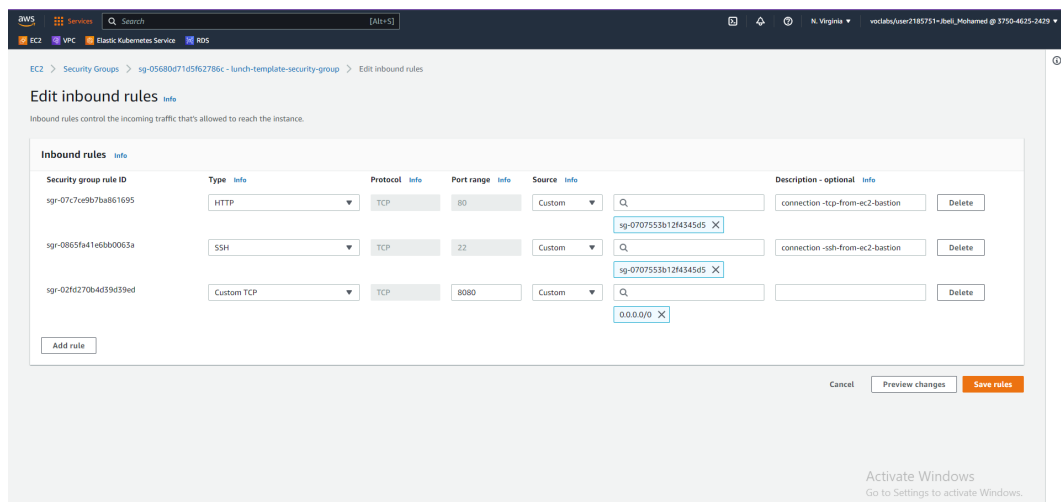
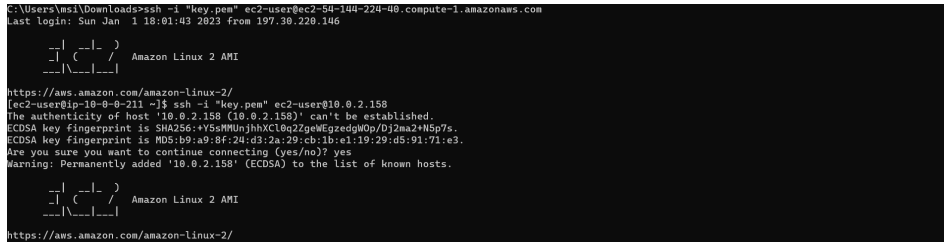


Figure 1.13: L'ensemble des règles du security groupe attachées à la machine EC2 hébergée dans le sousréseaux privé

- Voici l'imprime d'écran qui montre l'accès à la machine EC2 backend à travers le Secure Shell Protocol (SSH) (voir figure 1.14).



```

C:\Users\moi\Downloads>ssh -i "key.pem" ec2-user@ec2-54-144-224-48.compute-1.amazonaws.com
Last login: Sun Jan 1 18:01:43 2023 from 197.30.220.146

 _ _ _ _ _
| |   | |   | |   | |
|_|   |_|   |_|   |_|
 _ _ _ _ _

Amazon Linux 2 AMI

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-10-0-0-211 ~]$ ssh -i "key.pem" ec2-user@10.0.2.158
The authenticity of host '10.0.2.158 (10.0.2.158)' can't be established.
ECDSA key fingerprint is SHA256:+Ys5MMUnjhXCl0q2ZgeEgzedgWOp/Dj2ma2+NSp7s.
ECDSA key fingerprint is MD5:b9:a9:8f:2d:d3:2a:29:cb:1b:e1:19:29:d5:91:71:e3.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.2.158' (ECDSA) to the list of known hosts.

 _ _ _ _ _
| |   | |   | |   | |
|_|   |_|   |_|   |_|
 _ _ _ _ _

Amazon Linux 2 AMI

https://aws.amazon.com/amazon-linux-2/

```

Figure 1.14: L'accès à la machine EC2 backend à travers le Secure Shell Protocol (SSH)

- Voici l'imprime d'écran qui montre la configuration du security group de la base de données MYSQL (voir figure 1.15).

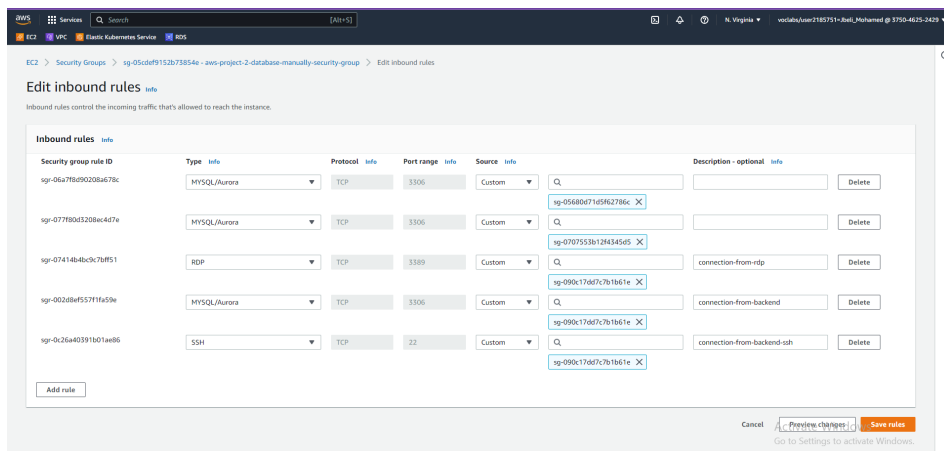


Figure 1.15: La configuration du security group de la base de données MYSQL

- Voici l'imprime d'écran qui montre la configuration du fichier docker pour la partie backend (voir figure 1.16).



Figure 1.16: La configuration du fichier docker pour la partie backend

- Voici l'imprime d'écran qui montre la nouvelle configuration MYSQL pour la partie backend(voir figure 1.17).



Figure 1.17: La nouvelle configuration MYSQL pour la partie backend

- Voici La publication de notre image associée au projet backend dans le Docker repository (voir figure 1.18).

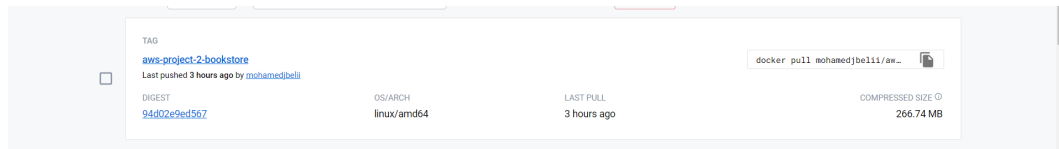


Figure 1.18: La publication de notre image associée au projet backend dans le Docker repository

-Voici l'imprime d'écran qui montre les étapes de l'hébergement de la solution backend dans la machine EC2 privée(voir figure 1.19).

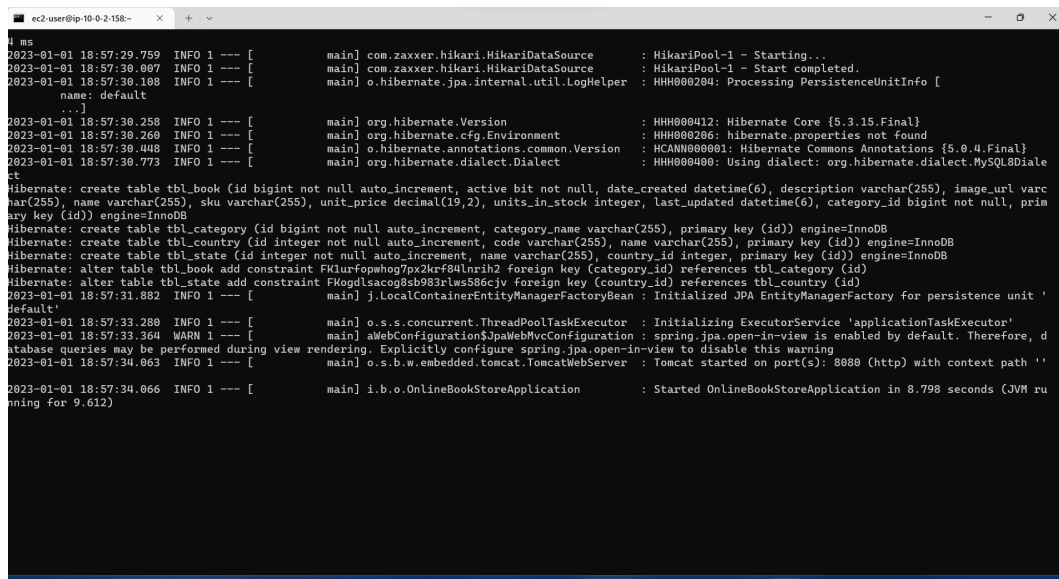


Figure 1.19: L'hébergement de la solution backend dans la machine EC2 privée

- Voici l'imprime d'écran qui montre le déploiement (voir figure 1.20).

```

[ec2-user@ip-10-0-2-158 ~]$ docker pull mohamedjbelii/aws_project:aws-project-2-bookstore
aws-project-2-bookstore: Pulling from mohamedjbelii/aws_project
38a989f2cc0a: Pull complete
de849f1cfbe6: Pull complete
a7203ca35e75: Pull complete
2d5125c00ff0: Pull complete
Digest: sha256:94d02e9ed5674933ffc854b2dcfd2e152d7d825416db551f0b449e75d6f42493
Status: Downloaded newer image for mohamedjbelii/aws_project:aws-project-2-bookstore
docker.io/mohamedjbelii/aws_project:aws-project-2-bookstore
[ec2-user@ip-10-0-2-158 ~]$ docker run --name back-end -p 8080:8080 mohamedjbelii/aws_project:aws-project-2-bookstore

:: Spring Boot ::
(v2.1.12.RELEASE)

2023-01-01 18:57:26.077 INFO 1 --- [main] i.b.o.OnlineBookStoreApplication : Starting OnlineBookStoreApplication v0.0.1-SNAPSHOT on f198
850e1501 with PID 1 (/onlinebookstore-0.0.1-SNAPSHOT.jar started by root in /)
2023-01-01 18:57:26.084 INFO 1 --- [main] i.b.o.OnlineBookStoreApplication : No active profile set, falling back to default profiles: de
fault
2023-01-01 18:57:27.735 INFO 1 --- [main] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repositories in DEFAULT mode.
2023-01-01 18:57:27.836 INFO 1 --- [main] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 87ms. Found 4 J
PA repository interfaces.
2023-01-01 18:57:28.535 INFO 1 --- [main] trationDelegate$BeanPostProcessorChecker : Bean 'org.springframework.transaction.annotation.ProxyTrans
actionManagementConfiguration' of type [org.springframework.transaction.annotation.ProxyTransactionManagementConfiguration$$EnhancerBySpringCGLIB$$7818dc0]
is not eligible for getting processed by all BeanPostProcessors (for example: not eligible for auto-proxying)
2023-01-01 18:57:28.580 INFO 1 --- [main] trationDelegate$BeanPostProcessorChecker : Bean 'org.springframework.hateoas.config.HateoasConfigurati
on' of type [org.springframework.hateoas.config.HateoasConfiguration$$EnhancerBySpringCGLIB$$8681daf2] is not eligible for getting processed by all BeanPost
Processors (for example: not eligible for auto-proxying)
2023-01-01 18:57:29.186 INFO 1 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8080 (http)
2023-01-01 18:57:29.258 INFO 1 --- [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2023-01-01 18:57:29.259 INFO 1 --- [main] org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/9.0.30]
2023-01-01 18:57:29.373 INFO 1 --- [main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext

```

Figure 1.20: le déploiement

- Voici l'imprime d'écran qui montre la configuration du fichier docker pour la partie frontend (voir figure 1.21).

```

angular-app
Dockerfile U X
Dockerfile > FROM
1 FROM nginx:1.17.1-alpine
2 COPY nginx.conf /etc/nginx/nginx.conf
3 COPY /dist/angular-app /usr/share/nginx/html

```

Figure 1.21: la configuration du fichier docker pour la partie frontend

- Voici l'imprime d'écran qui montre la création de la machine EC2 publique pour héberger la partie frontend (voir figure 1.22).

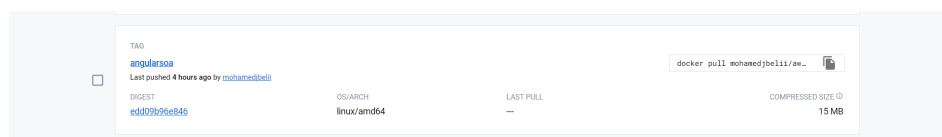


Figure 1.22: La création de la machine EC2 publique pour héberger la partie frontend

- Voici l'imprime d'écran qui montre la création d'un target group pour lier la connexion entre le load balancer applicative et la machine EC2 hébergée dans le sous réseau publique (voir figure 1.23).
- Voici l'imprime d'écran qui montre les détails du security group de la machine publique (voir figure 1.24 - figure 1.25).
- Voici l'imprime d'écran qui montre la page web (voir figure 1.26).
- Voici l'imprime d'écran qui montre la connexion du base de données à travers MYSQL WORKBENCH (voir figure 1.26).
- Voici l'imprime d'écran qui montre la création des tables a travers spring (voir figure 1.27).

```

2 history
admin@ip-10-0-0-42:~$ docker pull mohamedjbelii/aws_project:angularsoa
angularsoa: Pulling from mohamedjbelii/aws_project
e7c96db7181b: Pull complete
3fb6217217ef: Pull complete
3bab2d4dbc05: Pull complete
ebf70ab4900b: Pull complete
Digest: sha256:edd09b96e84615746ae80dc67454446376f2f16f7db8236e63879ade06810c72
Status: Downloaded newer image for mohamedjbelii/aws_project:angularsoa
docker.io/mohamedjbelii/aws_project:angularsoa
admin@ip-10-0-0-42:~$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
mohamedjbelii/aws_project  angularsoa         b1d187c3aa56       9 minutes ago      50.5MB
admin@ip-10-0-0-42:~$ docker run --name ang -p 4200:4200 b1d187c3aa56

```

Figure 1.23: La création d'un target group pour lier load balancer applicative et la machine EC2

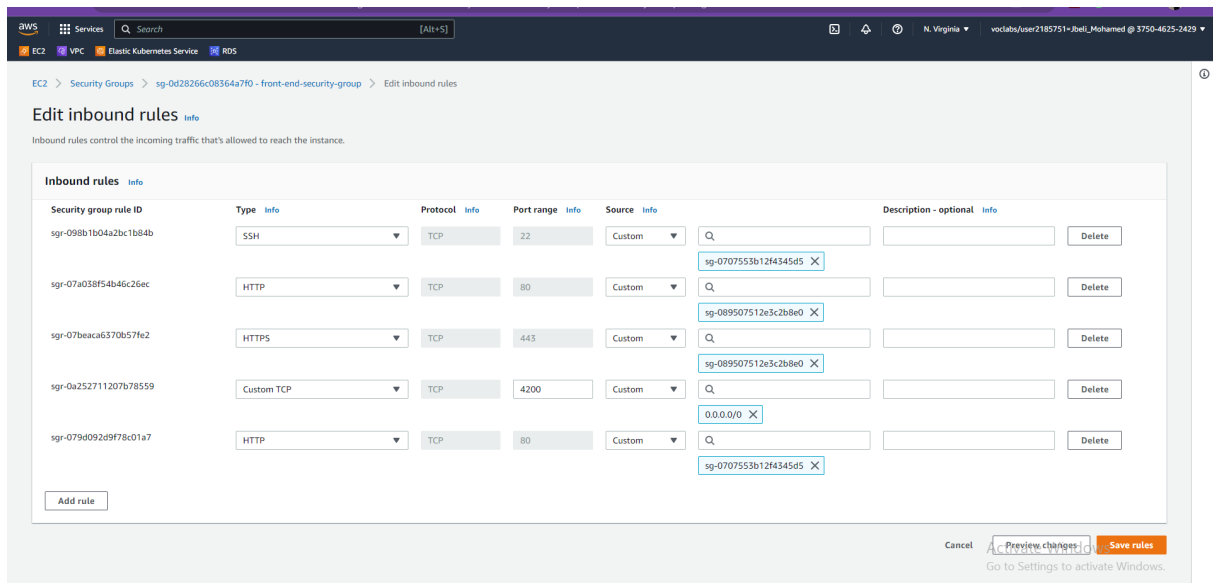


Figure 1.24: les détails du security group de la machine publique

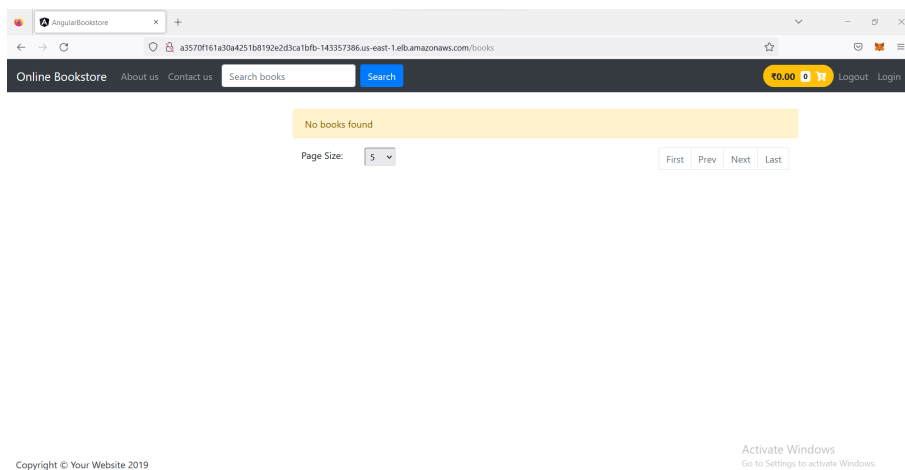


Figure 1.25: La liaison de la connexion entre le load balancer applicative et la machine EC2

1.4 Conclusion

À travers ce chapitre, nous avons décrit les solutions proposées à réaliser ainsi les détails du travail.

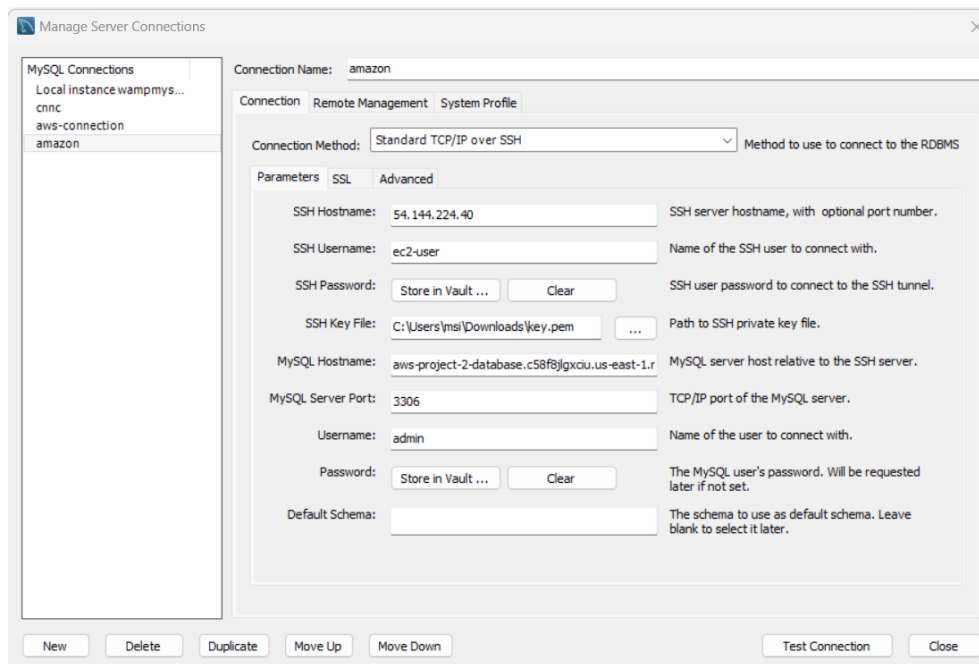


Figure 1.26: la connexion du base de données

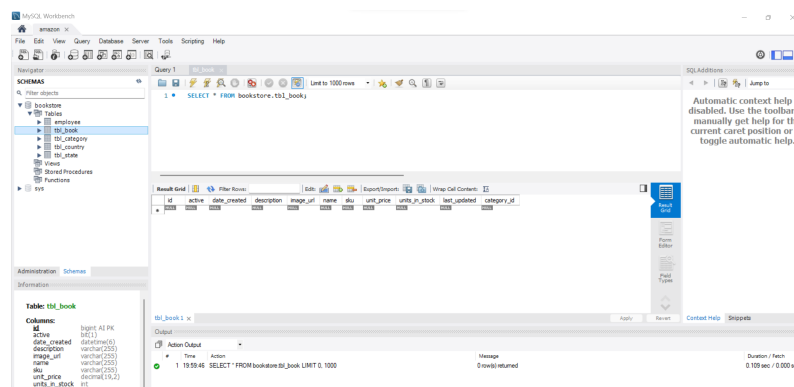


Figure 1.27: la création des tables