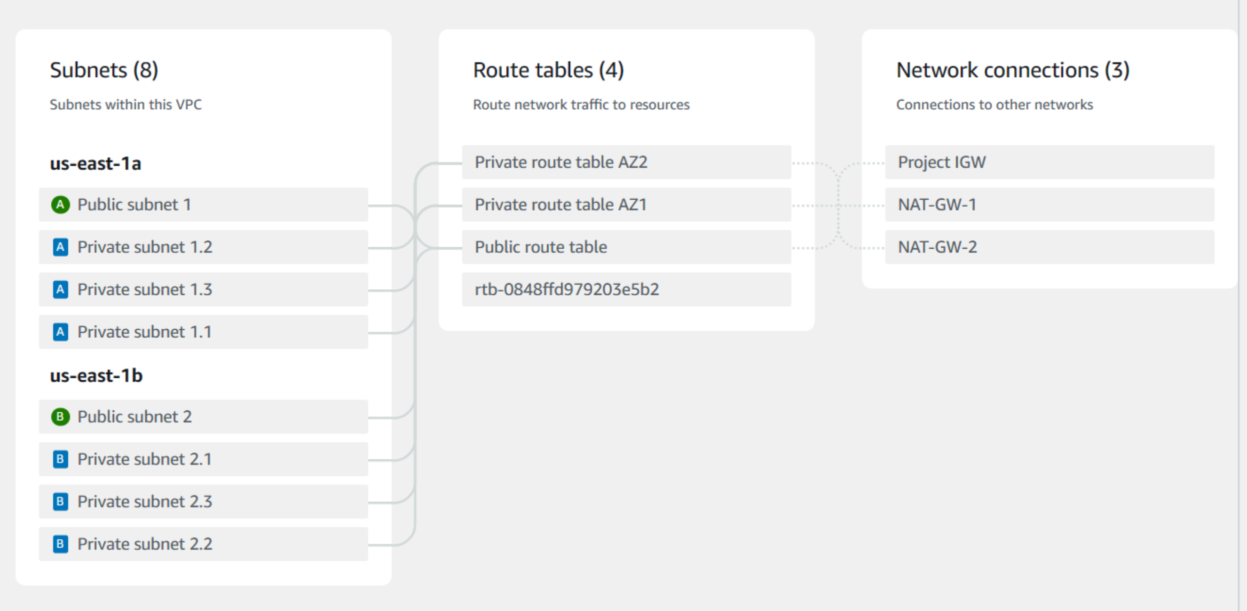
Cloud 3-tier architecture project

Adam Zouari

1. VPC
2. Security groups
3. RDS
4. App Server
5. Web Server

VPC

I started by creating a VPC named "project" in the us-east-1 region with two availability zones (AZ1 being us-east-1a and AZ2 being us-east-1b), each containing one public subnet and three private subnets.



Subnets (8)

Subnets within this VPC

us-east-1a

- Public subnet 1
- Private subnet 1.2
- Private subnet 1.3
- Private subnet 1.1

us-east-1b

- Public subnet 2
- Private subnet 2.1
- Private subnet 2.3
- Private subnet 2.2

Route tables (4)

Route network traffic to resources

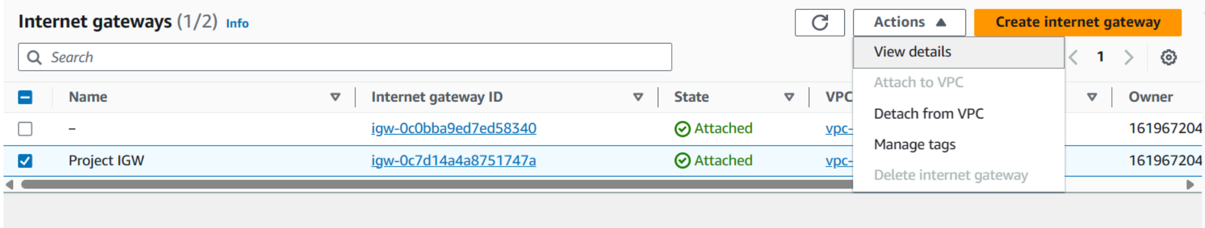
- Private route table AZ2
- Private route table AZ1
- Public route table
- rtb-0848ffd979203e5b2

Network connections (3)

Connections to other networks

- Project IGW
- NAT-GW-1
- NAT-GW-2

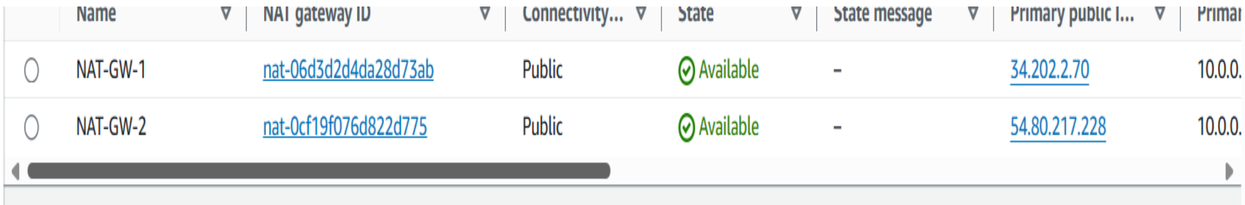
First, I created an Internet Gateway and attached it to the VPC.



Internet gateways (1/2)

	Name	Internet gateway ID	State	VPC	Owner
☐	-	igw-0c0bba9ed7ed58340	✓ Attached	vpc-	161967204
☑	Project IGW	igw-0c7d14a4a8751747a	✓ Attached	vpc-	161967204

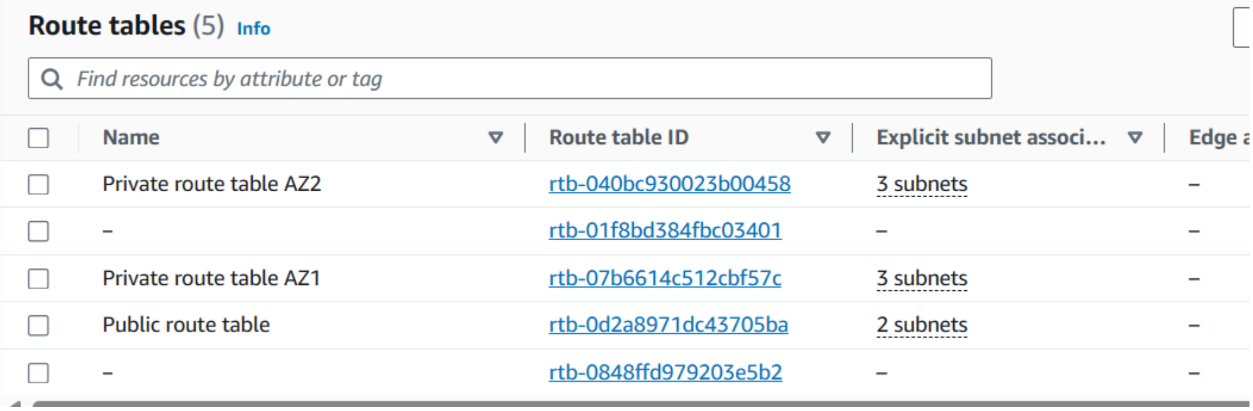
Next, I created two NAT Gateways for each availability zone.



	Name	NAT gateway ID	Connectivity...	State	State message	Primary public I...	Primai
○	NAT-GW-1	nat-06d3d2d4da28d73ab	Public	✓ Available	-	34.202.2.70	10.0.0.
○	NAT-GW-2	nat-0cf19f076d822d775	Public	✓ Available	-	54.80.217.228	10.0.0.

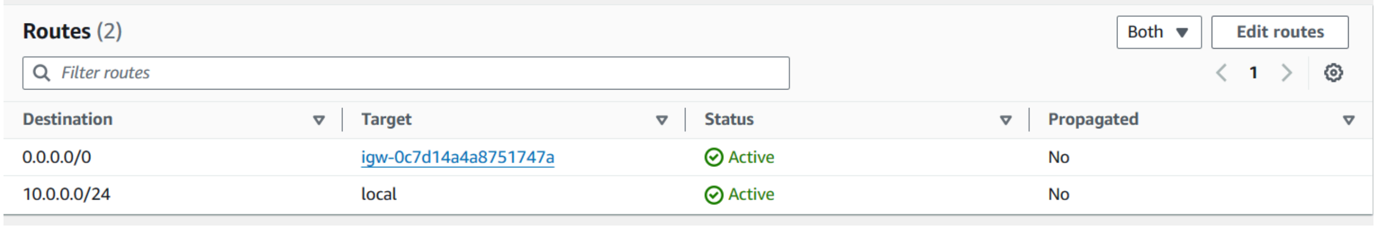
Then, I created the routing tables:

- A public routing table for the public subnets.
- A private routing table for AZ1 in us-east-1a and a private routing table for AZ2 in us-east-1b.



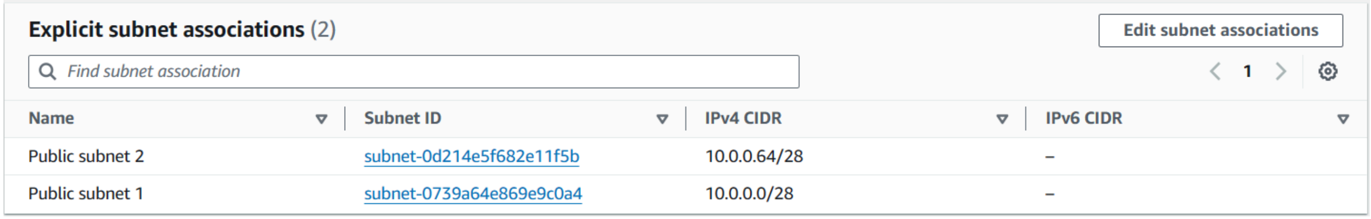
□	Name	Route table ID	Explicit subnet associ...	Edge a
□	Private route table AZ2	rtb-040bc930023b00458	3 subnets	–
□	–	rtb-01f8bd384fbc03401	–	–
□	Private route table AZ1	rtb-07b6614c512cbf57c	3 subnets	–
□	Public route table	rtb-0d2a8971dc43705ba	2 subnets	–
□	–	rtb-0848ffd979203e5b2	–	–

For Public route table :



Routes (2)

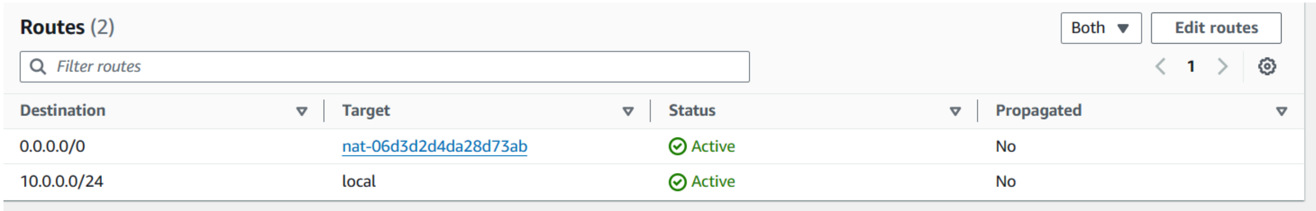
Destination	Target	Status	Propagated
0.0.0.0/0	igw-0c7d14a4a8751747a	✓ Active	No
10.0.0.0/24	local	✓ Active	No



Explicit subnet associations (2)

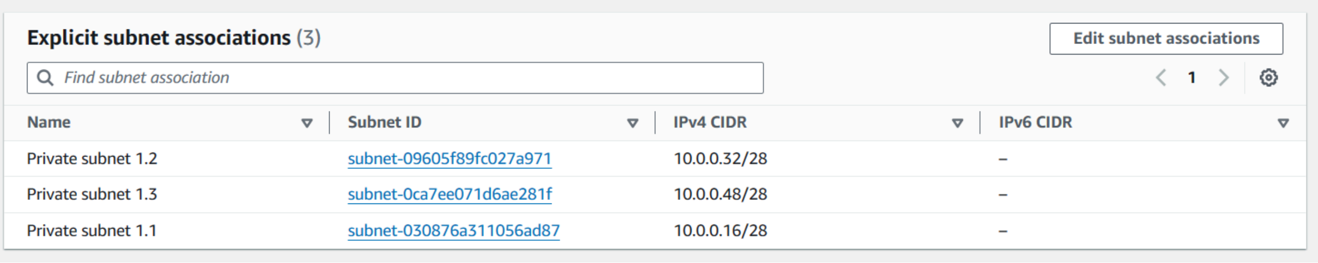
Name	Subnet ID	IPv4 CIDR	IPv6 CIDR
Public subnet 2	subnet-0d214e5f682e11f5b	10.0.0.64/28	–
Public subnet 1	subnet-0739a64e869e9c0a4	10.0.0.0/28	–

For Private route table AZ1 :



Routes (2)

Destination	Target	Status	Propagated
0.0.0.0/0	nat-06d3d2d4da28d73ab	✓ Active	No
10.0.0.0/24	local	✓ Active	No



Explicit subnet associations (3)

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR
Private subnet 1.2	subnet-09605f89fc027a971	10.0.0.32/28	–
Private subnet 1.3	subnet-0ca7ee071d6ae281f	10.0.0.48/28	–
Private subnet 1.1	subnet-030876a311056ad87	10.0.0.16/28	–

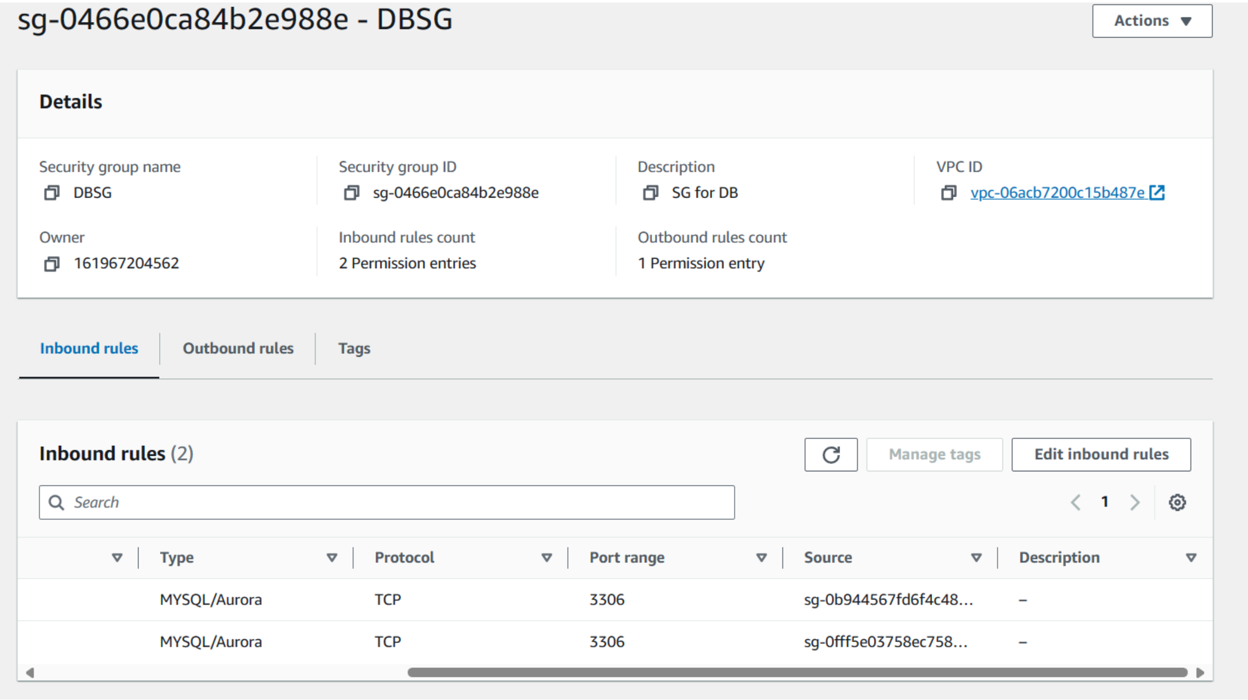
For the private routing table in AZ2:

It is configured similarly to the private routing table in AZ1, except it is attached to its own NAT Gateway and the private subnets created earlier.

After completing the network configuration, I proceeded to the next step.

Security groups

First, I created a security group for the database.



sg-0466e0ca84b2e988e - DBSG

Details

Security group name: DBSG

Security group ID: sg-0466e0ca84b2e988e

Description: SG for DB

VPC ID: vpc-06acb7200c15b487e

Owner: 161967204562

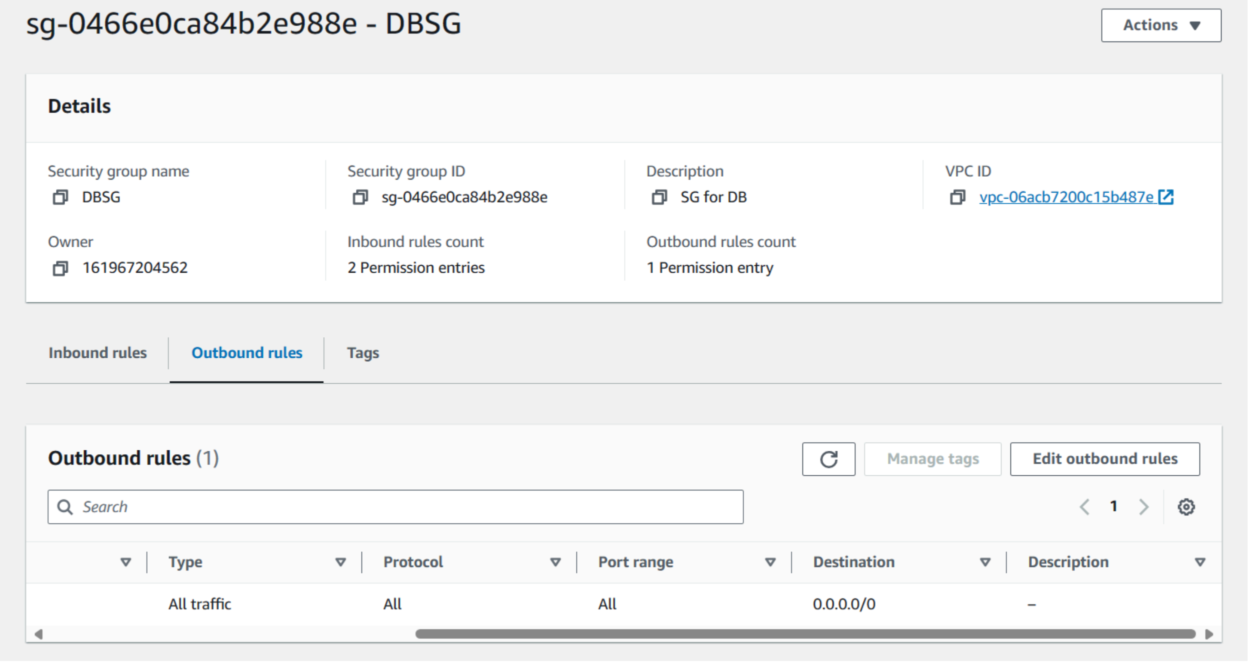
Inbound rules count: 2 Permission entries

Outbound rules count: 1 Permission entry

Inbound rules | Outbound rules | Tags

Inbound rules (2)

Type	Protocol	Port range	Source	Description
MYSQL/Aurora	TCP	3306	sg-0b944567fd6f4c48…	–
MYSQL/Aurora	TCP	3306	sg-0fff5e03758ec758…	–



sg-0466e0ca84b2e988e - DBSG

Details

Security group name: DBSG

Security group ID: sg-0466e0ca84b2e988e

Description: SG for DB

VPC ID: vpc-06acb7200c15b487e

Owner: 161967204562

Inbound rules count: 2 Permission entries

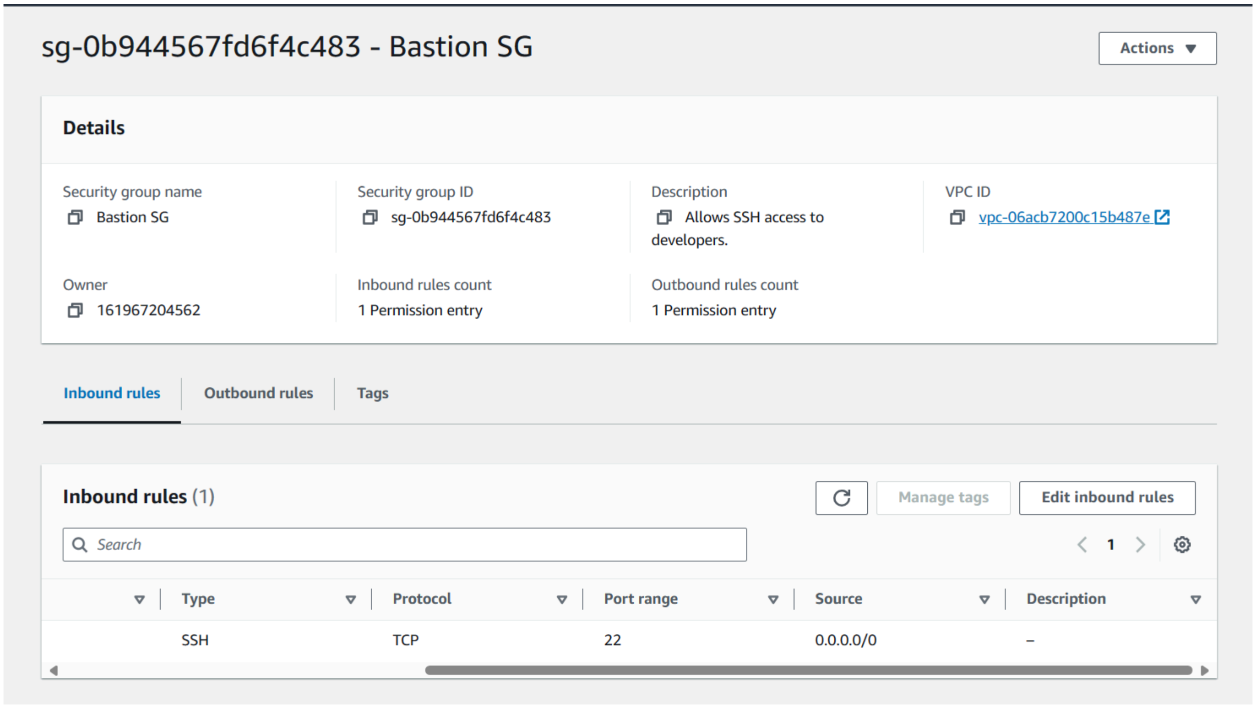
Outbound rules count: 1 Permission entry

Inbound rules | Outbound rules | Tags

Outbound rules (1)

Type	Protocol	Port range	Destination	Description
All traffic	All	All	0.0.0.0/0	–

Next, I created a security group for the bastion host.



sg-0b944567fd6f4c483 - Bastion SG

Details

Security group name: Bastion SG

Security group ID: sg-0b944567fd6f4c483

Description: Allows SSH access to developers.

VPC ID: vpc-06acb7200c15b487e

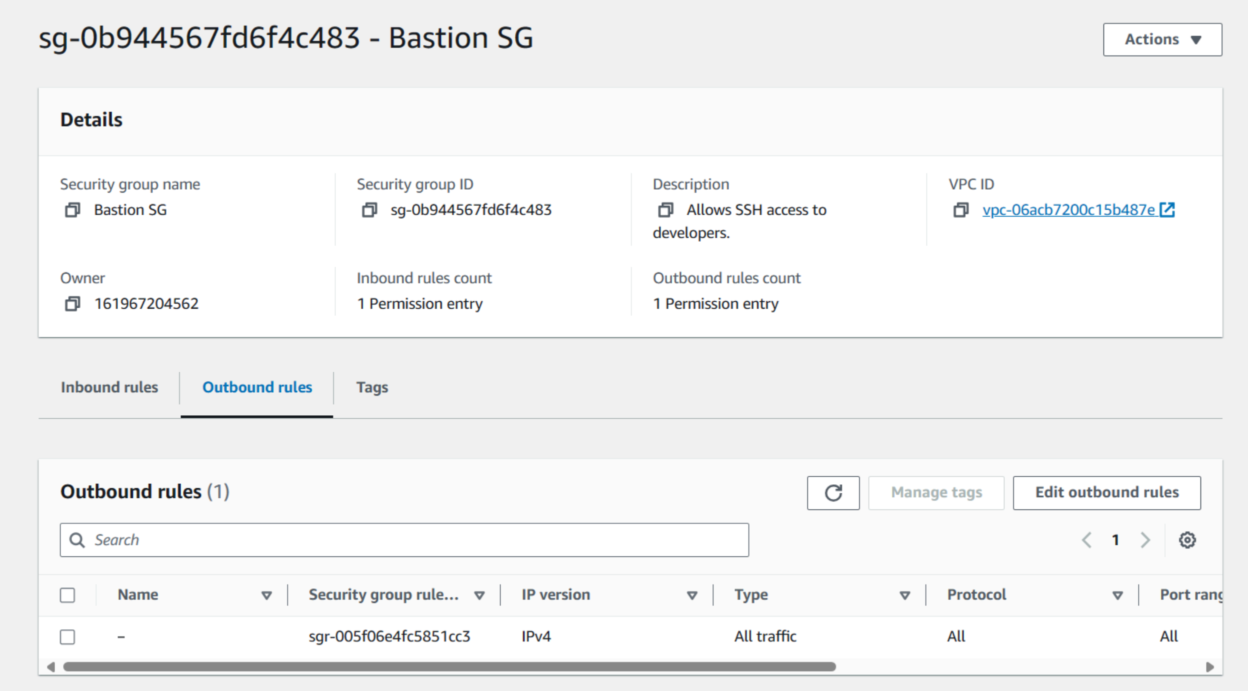
Owner: 161967204562

Inbound rules count: 1 Permission entry

Outbound rules count: 1 Permission entry

Inbound rules (1)

Type	Protocol	Port range	Source	Description
SSH	TCP	22	0.0.0.0/0	–



sg-0b944567fd6f4c483 - Bastion SG

Details

Security group name: Bastion SG

Security group ID: sg-0b944567fd6f4c483

Description: Allows SSH access to developers.

VPC ID: vpc-06acb7200c15b487e

Owner: 161967204562

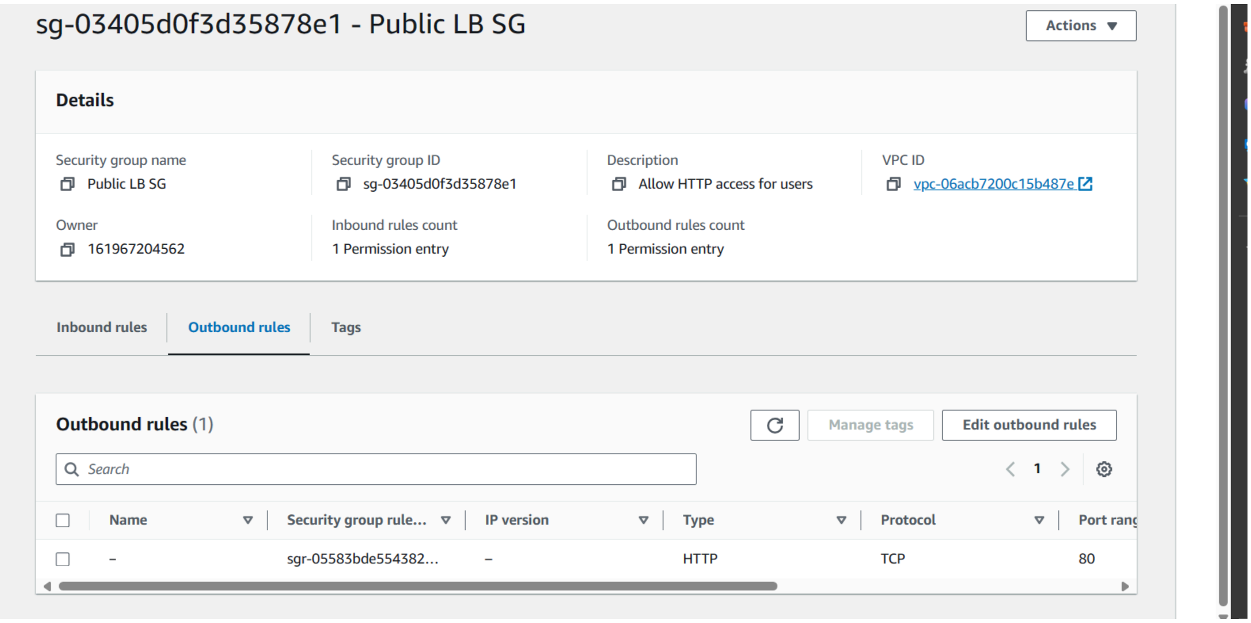
Inbound rules count: 1 Permission entry

Outbound rules count: 1 Permission entry

Outbound rules (1)

□	Name	Security group rule...	IP version	Type	Protocol	Port rang
□	–	sgr-005f06e4fc5851cc3	IPv4	All traffic	All	All

Next, I created a security group for the internet-facing load balancer.



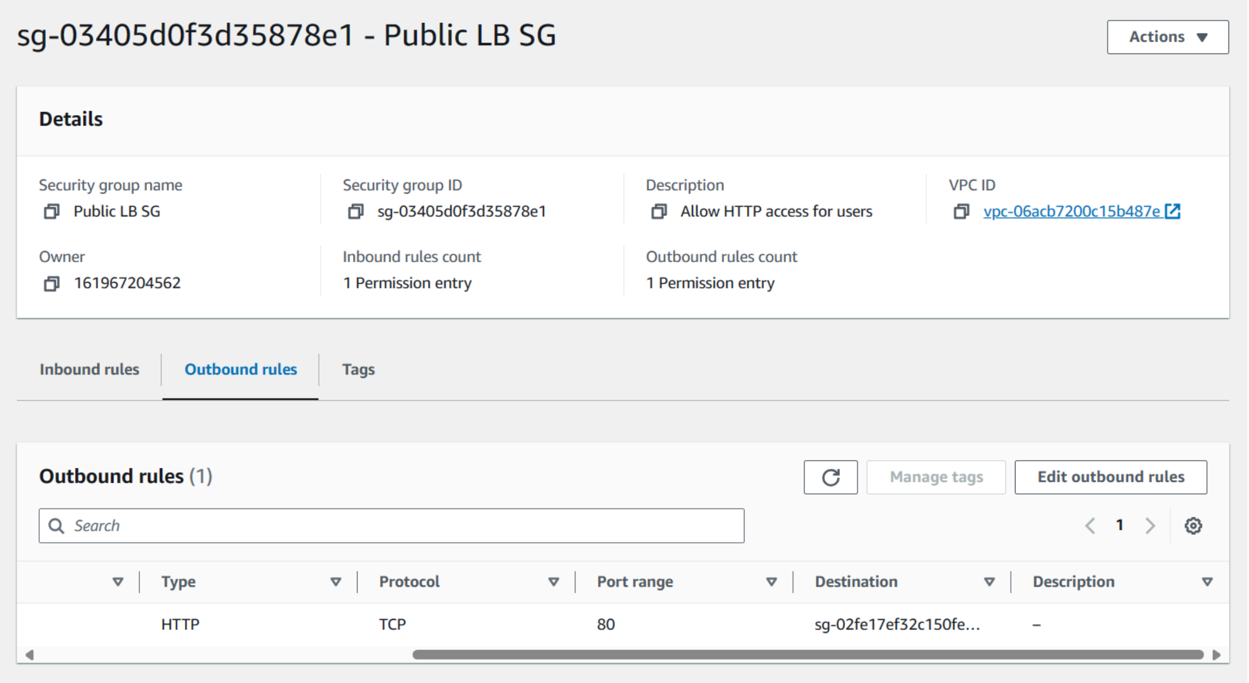
sg-03405d0f3d35878e1 - Public LB SG

Details

Security group name	Security group ID	Description	VPC ID
Public LB SG	sg-03405d0f3d35878e1	Allow HTTP access for users	vpc-06acb7200c15b487e
Owner	**Inbound rules count**	**Outbound rules count**	
161967204562	1 Permission entry	1 Permission entry	

Outbound rules (1)

□	Name	Security group rule...	IP version	Type	Protocol	Port rang
□	–	sgr-05583bde554382...	–	HTTP	TCP	80



sg-03405d0f3d35878e1 - Public LB SG

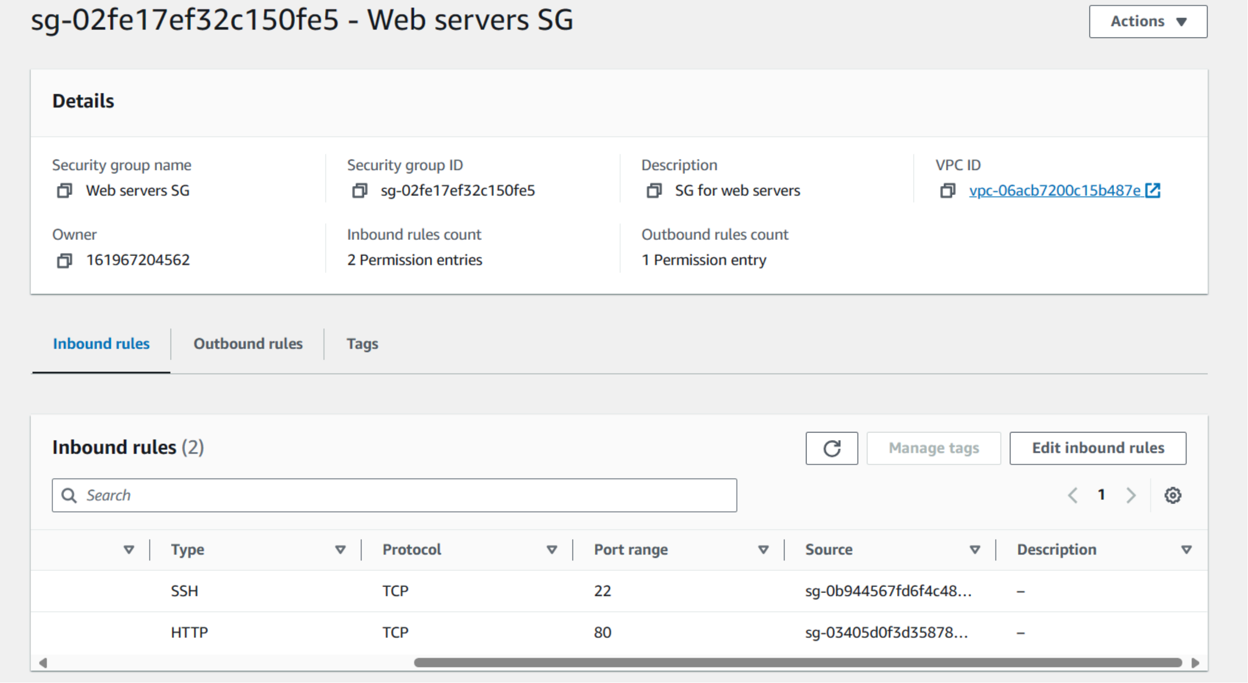
Details

Security group name	Security group ID	Description	VPC ID
Public LB SG	sg-03405d0f3d35878e1	Allow HTTP access for users	vpc-06acb7200c15b487e
Owner	**Inbound rules count**	**Outbound rules count**	
161967204562	1 Permission entry	1 Permission entry	

Outbound rules (1)

Type	Protocol	Port range	Destination	Description
HTTP	TCP	80	sg-02fe17ef32c150fe...	–

Next, I created a security group for the web servers.



sg-02fe17ef32c150fe5 - Web servers SG

Details

Security group name: Web servers SG

Security group ID: sg-02fe17ef32c150fe5

Description: SG for web servers

VPC ID: vpc-06acb7200c15b487e

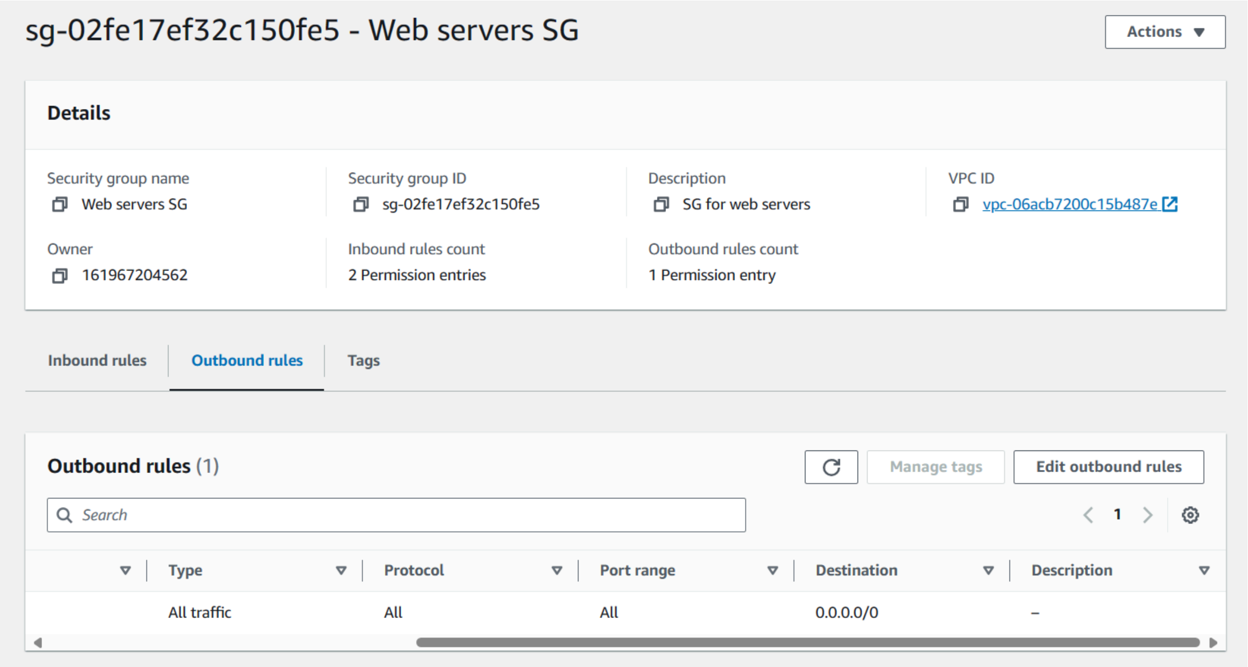
Owner: 161967204562

Inbound rules count: 2 Permission entries

Outbound rules count: 1 Permission entry

Inbound rules (2)

Type	Protocol	Port range	Source	Description
SSH	TCP	22	sg-0b944567fd6f4c48…	–
HTTP	TCP	80	sg-03405d0f3d35878…	–



sg-02fe17ef32c150fe5 - Web servers SG

Details

Security group name: Web servers SG

Security group ID: sg-02fe17ef32c150fe5

Description: SG for web servers

VPC ID: vpc-06acb7200c15b487e

Owner: 161967204562

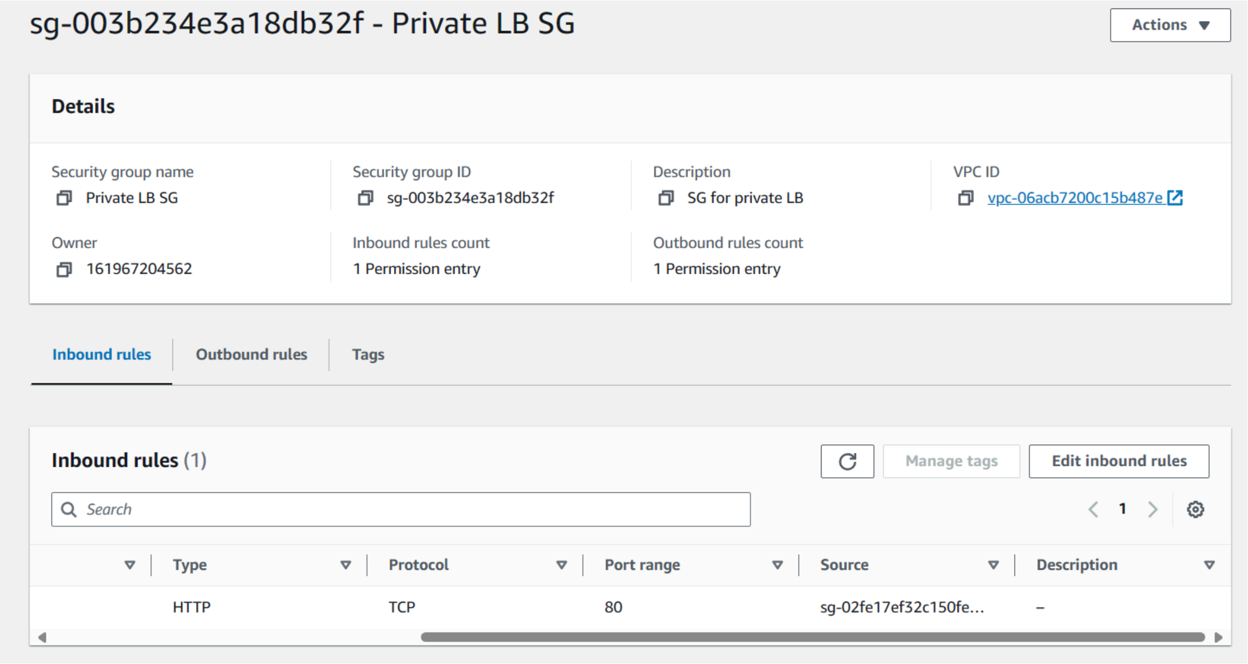
Inbound rules count: 2 Permission entries

Outbound rules count: 1 Permission entry

Outbound rules (1)

Type	Protocol	Port range	Destination	Description
All traffic	All	All	0.0.0.0/0	–

Next, I created a security group for the internal load balancer.



sg-003b234e3a18db32f - Private LB SG

Details

Security group name: Private LB SG

Security group ID: sg-003b234e3a18db32f

Description: SG for private LB

VPC ID: vpc-06acb7200c15b487e

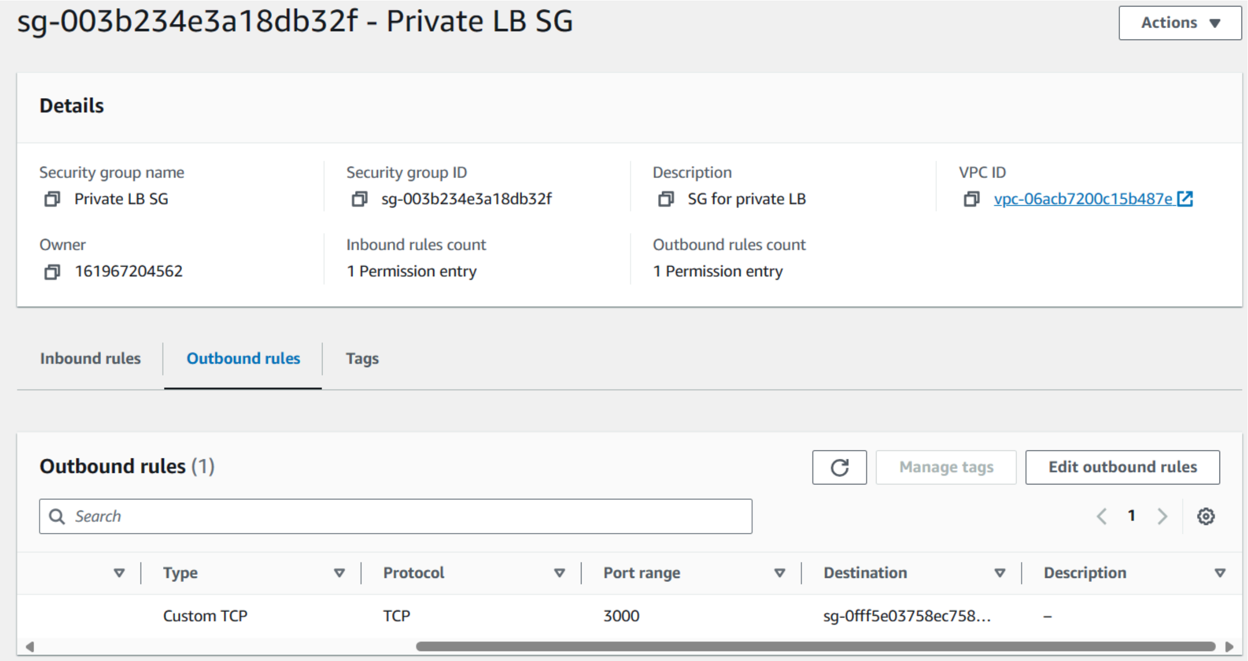
Owner: 161967204562

Inbound rules count: 1 Permission entry

Outbound rules count: 1 Permission entry

Inbound rules (1)

Type	Protocol	Port range	Source	Description
HTTP	TCP	80	sg-02fe17ef32c150fe...	–



sg-003b234e3a18db32f - Private LB SG

Details

Security group name: Private LB SG

Security group ID: sg-003b234e3a18db32f

Description: SG for private LB

VPC ID: vpc-06acb7200c15b487e

Owner: 161967204562

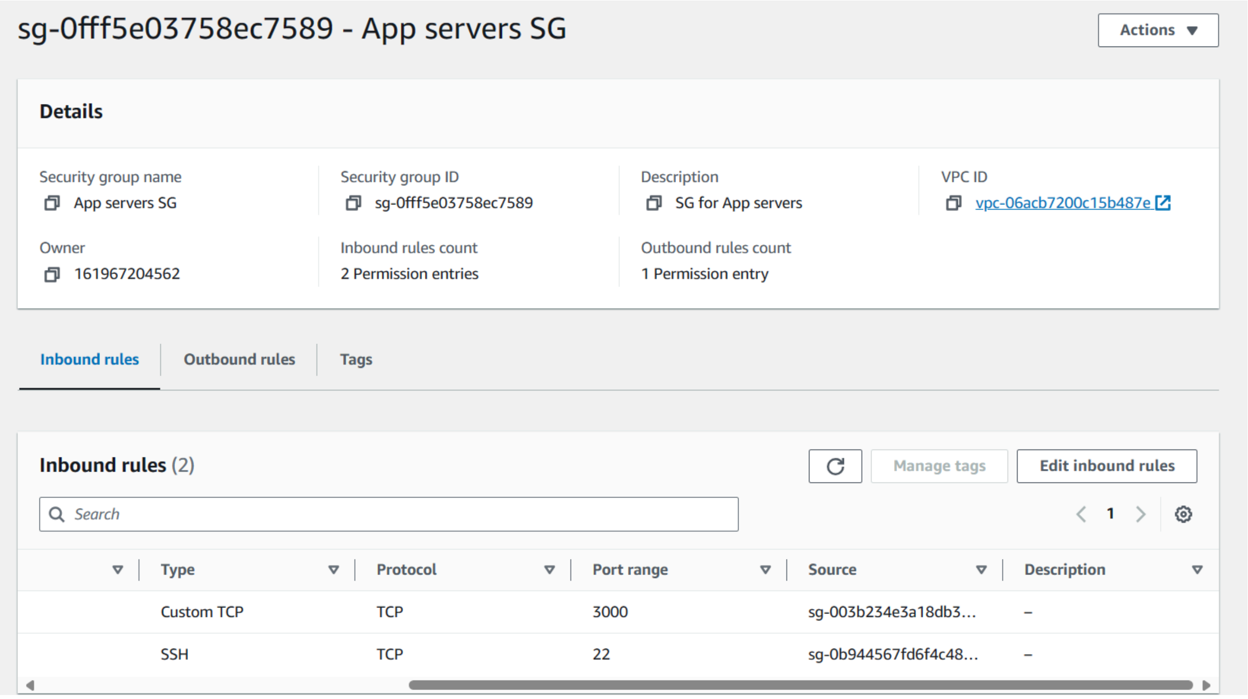
Inbound rules count: 1 Permission entry

Outbound rules count: 1 Permission entry

Outbound rules (1)

Type	Protocol	Port range	Destination	Description
Custom TCP	TCP	3000	sg-0fff5e03758ec758...	–

Then, I created one for the application servers.



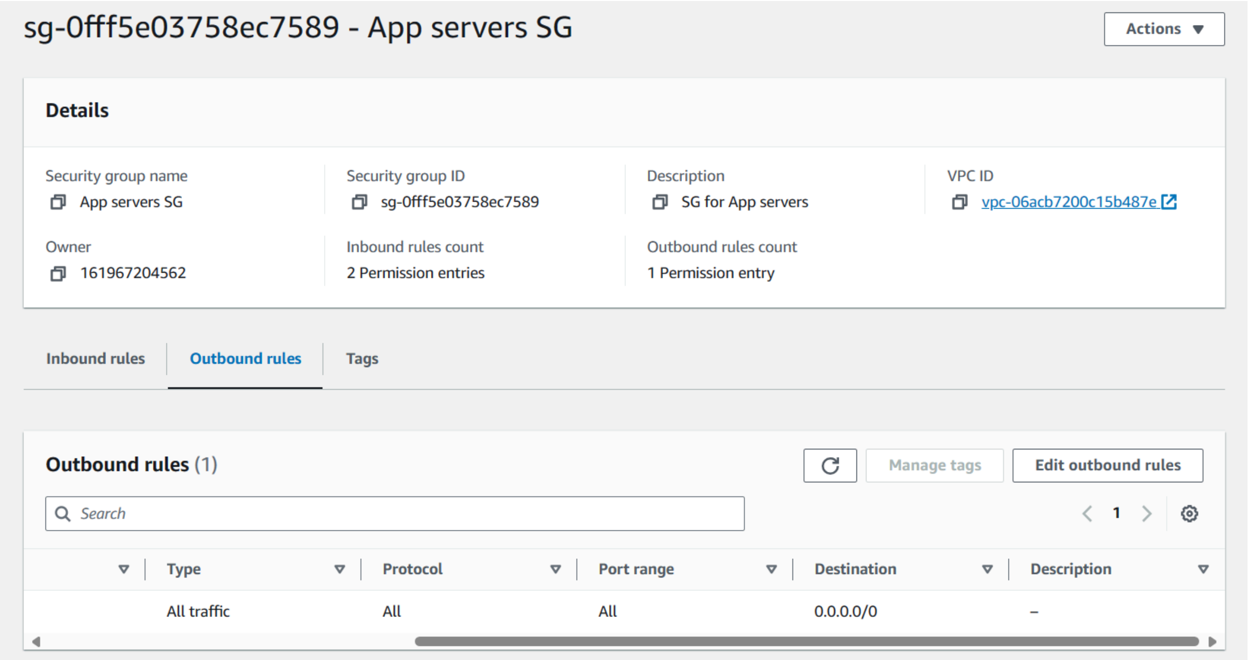
sg-0fff5e03758ec7589 - App servers SG

Details

Security group name	Security group ID	Description	VPC ID
App servers SG	sg-0fff5e03758ec7589	SG for App servers	vpc-06acb7200c15b487e
Owner	**Inbound rules count**	**Outbound rules count**	
161967204562	2 Permission entries	1 Permission entry	

Inbound rules (2)

Type	Protocol	Port range	Source	Description
Custom TCP	TCP	3000	sg-003b234e3a18db3…	–
SSH	TCP	22	sg-0b944567fd6f4c48…	–



sg-0fff5e03758ec7589 - App servers SG

Details

Security group name	Security group ID	Description	VPC ID
App servers SG	sg-0fff5e03758ec7589	SG for App servers	vpc-06acb7200c15b487e
Owner	**Inbound rules count**	**Outbound rules count**	
161967204562	2 Permission entries	1 Permission entry	

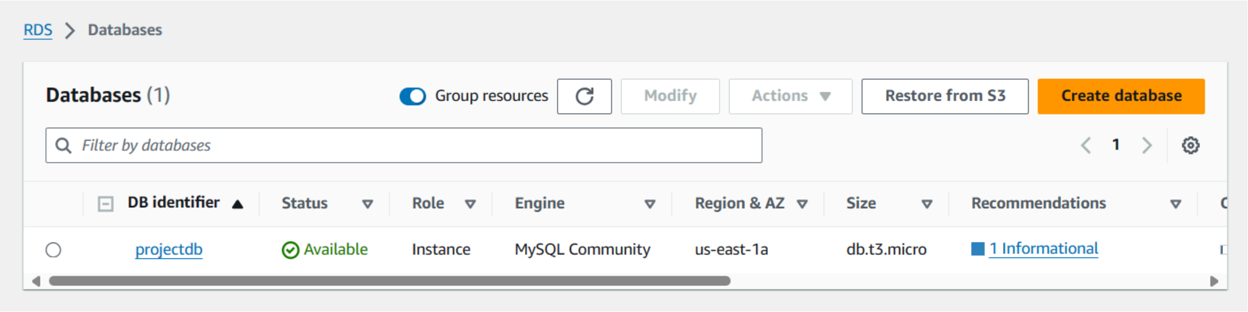
Outbound rules (1)

Type	Protocol	Port range	Destination	Description
All traffic	All	All	0.0.0.0/0	–

Then, I moved on to the next step.

RDS

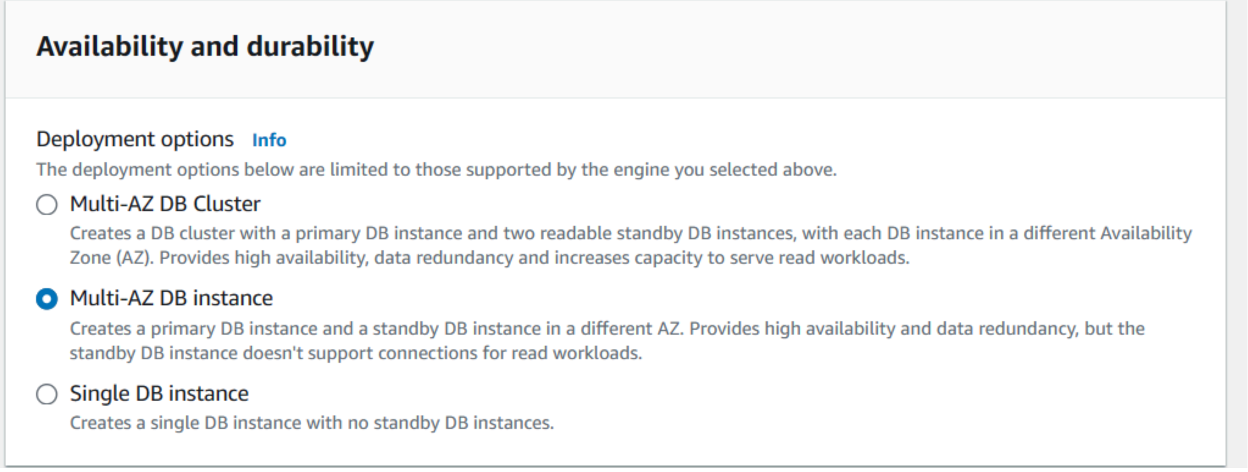
I created a MySQL database instance `db.t3.micro` with 20 GB of storage named `projectdb` in AZ1, with a standby instance in AZ2.



RDS > Databases

Databases (1)

	DB identifier ▲	Status	Role	Engine	Region & AZ	Size	Recommendations
○	projectdb	✓ Available	Instance	MySQL Community	us-east-1a	db.t3.micro	1 Informational



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

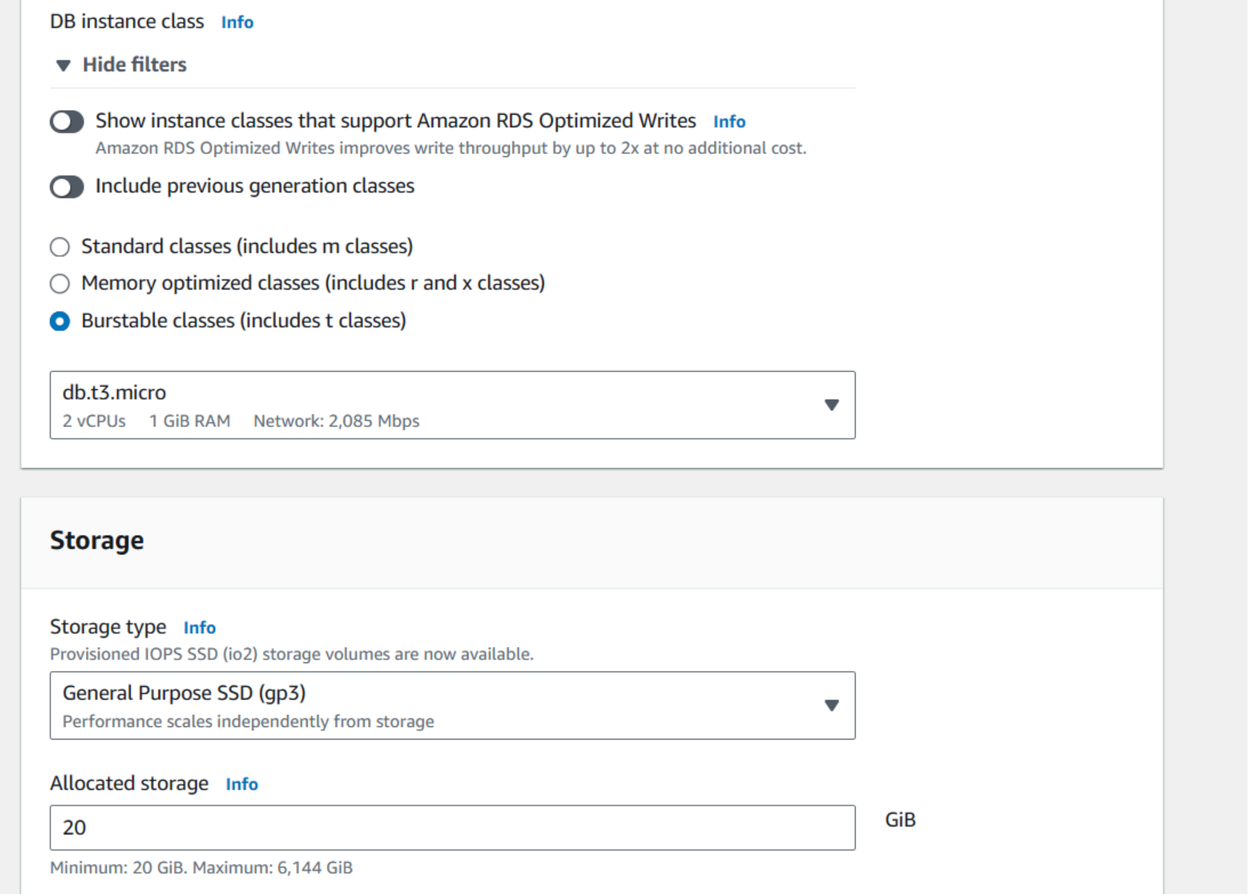
Creates a DB cluster with a primary DB instance and two readable standby DB instances, with each DB instance in a different Availability Zone (AZ). Provides high availability, data redundancy and increases capacity to serve read workloads.

● Multi-AZ DB instance

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 workloads.

○ Single DB instance

Creates a single DB instance with no standby DB instances.



DB instance class Info

▼ Hide filters

Show instance classes that support Amazon RDS Optimized Writes Info
Amazon RDS Optimized Writes improves write throughput by up to 2x at no additional cost.

Include previous generation classes

○ Standard classes (includes m classes)
○ Memory optimized classes (includes r and x classes)
● Burstable classes (includes t classes)

db.t3.micro
2 vCPUs 1 GiB RAM Network: 2,085 Mbps

Storage

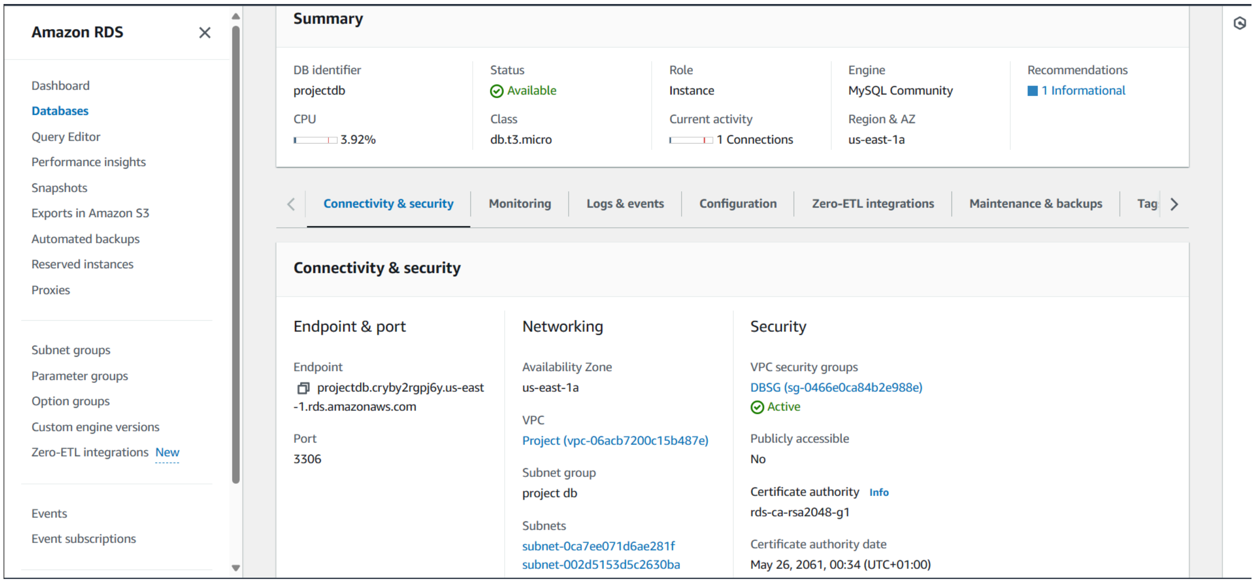
Storage type Info
Provisioned IOPS SSD (io2) storage volumes are now available.

General Purpose SSD (gp3)
Performance scales independently from storage

Allocated storage Info

20 GiB

Minimum: 20 GiB. Maximum: 6,144 GiB



Summary

DB identifier	Status	Role	Engine	Recommendations
projectdb	✓ Available	Instance	MySQL Community	1 Informational
CPU	Class	Current activity	Region & AZ	
3.92%	db.t3.micro	1 Connections	us-east-1a	

Connectivity & security

Endpoint & port

Endpoint
projectdb.cryby2rgpj6y.us-east-1.rds.amazonaws.com

Port
3306

Networking

Availability Zone
us-east-1a

VPC
Project (vpc-06acb7200c15b487e)

Subnet group
project db

Subnets
subnet-0ca7ee071d6ae281f
subnet-002d5153d5c2630ba

Security

VPC security groups
DBSG (sg-0466e0ca84b2e988e)
✓ Active

Publicly accessible
No

Certificate authority Info
rds-ca-rsa2048-g1

Certificate authority date
May 26, 2061, 00:34 (UTC+01:00)

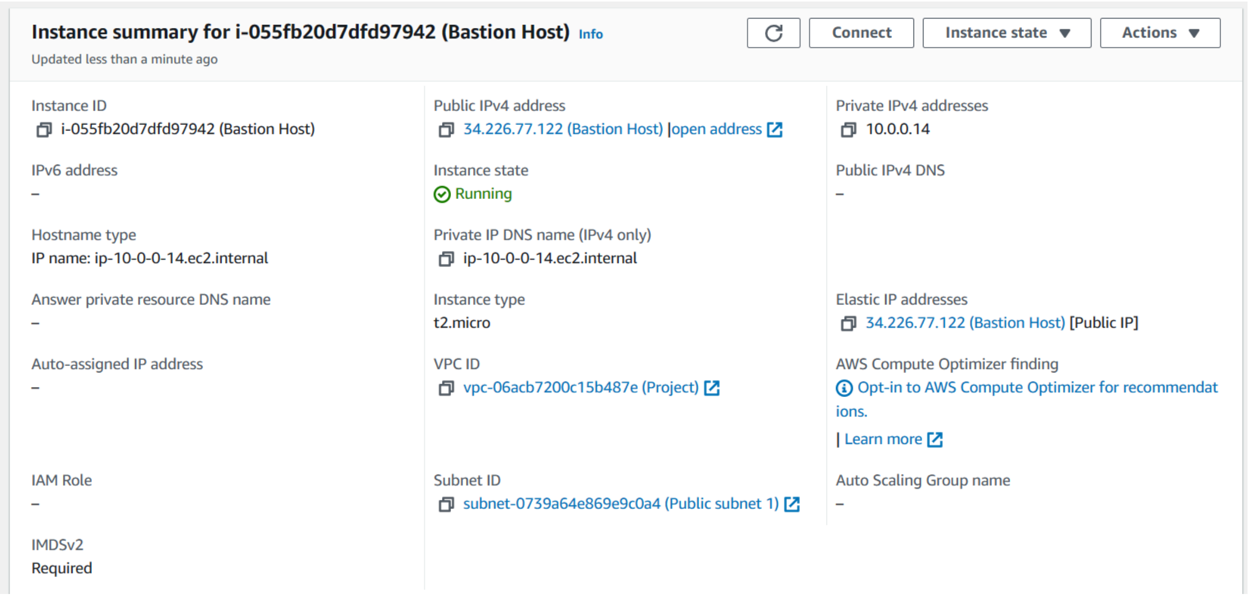
Then, I proceeded to the next step.

App Servers

I created three instances:

- An EC2 Bastion host.
- An EC2 web server.
- An EC2 application server.

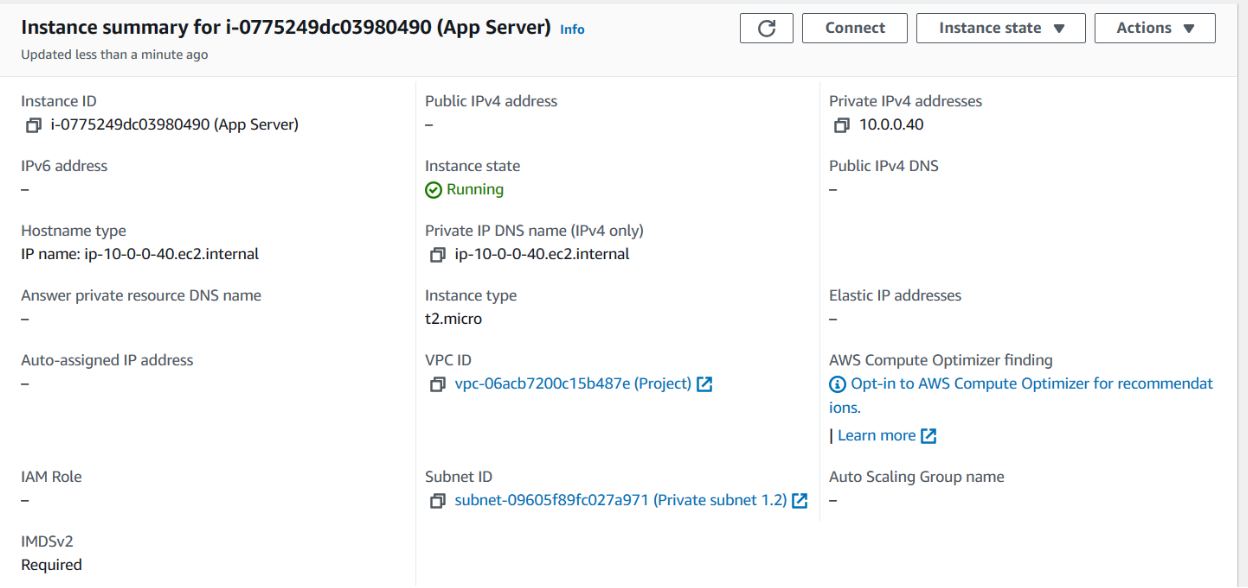
I generated an SSH key pair during the creation of the EC2 Bastion host and used the same key pair (Host.pem) for creating the other instances.



Instance summary for i-055fb20d7dfd97942 (Bastion Host)

Updated less than a minute ago

Instance ID i-055fb20d7dfd97942 (Bastion Host)	Public IPv4 address 34.226.77.122 (Bastion Host) |open address	Private IPv4 addresses 10.0.0.14
IPv6 address –	Instance state Running	Public IPv4 DNS –
Hostname type IP name: ip-10-0-0-14.ec2.internal	Private IP DNS name (IPv4 only) ip-10-0-0-14.ec2.internal	
Answer private resource DNS name –	Instance type t2.micro	Elastic IP addresses 34.226.77.122 (Bastion Host) [Public IP]
Auto-assigned IP address –	VPC ID vpc-06acb7200c15b487e (Project)	AWS Compute Optimizer finding Opt-in to AWS Compute Optimizer for recommendations. | Learn more
IAM Role –	Subnet ID subnet-0739a64e869e9c0a4 (Public subnet 1)	Auto Scaling Group name –
IMDSv2 Required		



Instance summary for i-0775249dc03980490 (App Server)

Updated less than a minute ago

Instance ID
i-0775249dc03980490 (App Server)

IPv6 address
–

Hostname type
IP name: ip-10-0-0-40.ec2.internal

Answer private resource DNS name
–

Auto-assigned IP address
–

IAM Role
–

IMDSv2
Required

Public IPv4 address
–

Instance state
Running

Private IP DNS name (IPv4 only)
ip-10-0-0-40.ec2.internal

Instance type
t2.micro

VPC ID
vpc-06acb7200c15b487e (Project)

Subnet ID
subnet-09605f89fc027a971 (Private subnet 1.2)

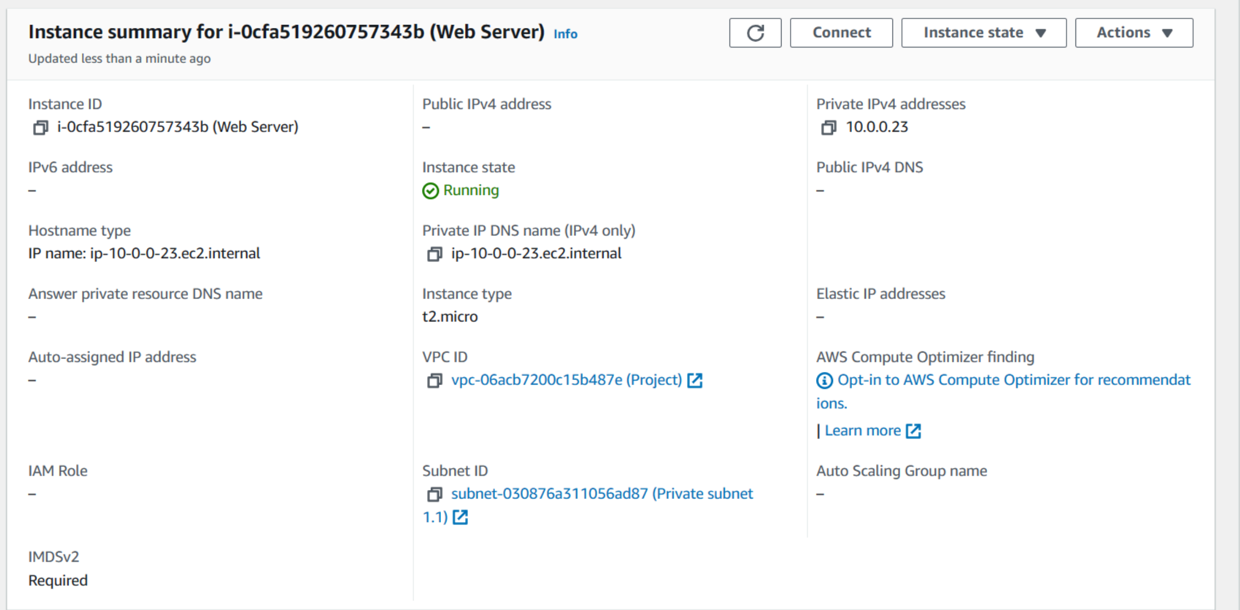
Private IPv4 addresses
10.0.0.40

Public IPv4 DNS
–

Elastic IP addresses
–

AWS Compute Optimizer finding
Opt-in to AWS Compute Optimizer for recommendations.
| Learn more

Auto Scaling Group name
–



Instance summary for i-0cfa519260757343b (Web Server)

Updated less than a minute ago

Instance ID
i-0cfa519260757343b (Web Server)

IPv6 address
–

Hostname type
IP name: ip-10-0-0-23.ec2.internal

Answer private resource DNS name
–

Auto-assigned IP address
–

IAM Role
–

IMDSv2
Required

Public IPv4 address
–

Instance state
Running

Private IP DNS name (IPv4 only)
ip-10-0-0-23.ec2.internal

Instance type
t2.micro

VPC ID
vpc-06acb7200c15b487e (Project)

Subnet ID
subnet-030876a311056ad87 (Private subnet 1.1)

Private IPv4 addresses
10.0.0.23

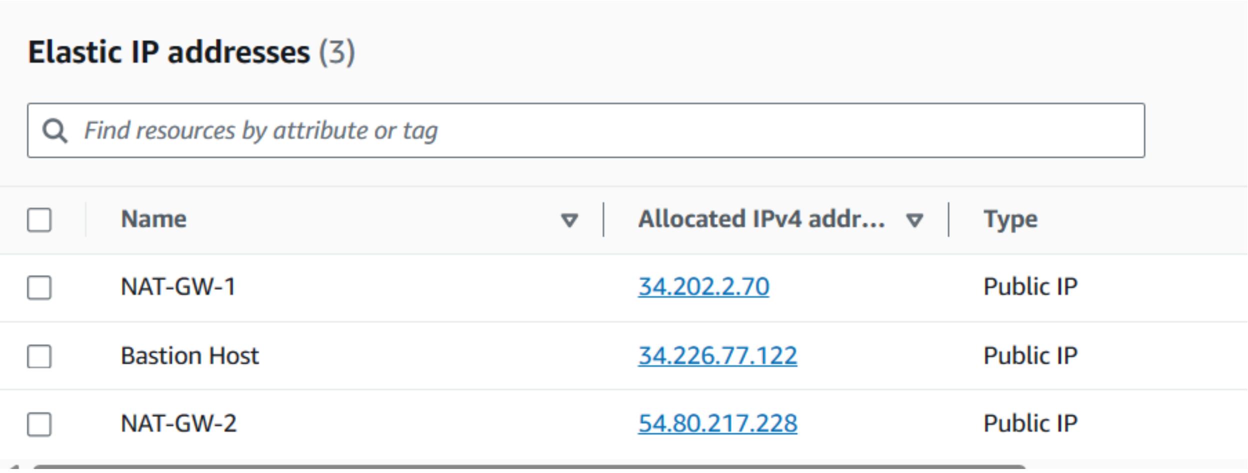
Public IPv4 DNS
–

Elastic IP addresses
–

AWS Compute Optimizer finding
Opt-in to AWS Compute Optimizer for recommendations.
| Learn more

Auto Scaling Group name
–

I also created an Elastic IP address for the Bastion host to facilitate easy access to it.

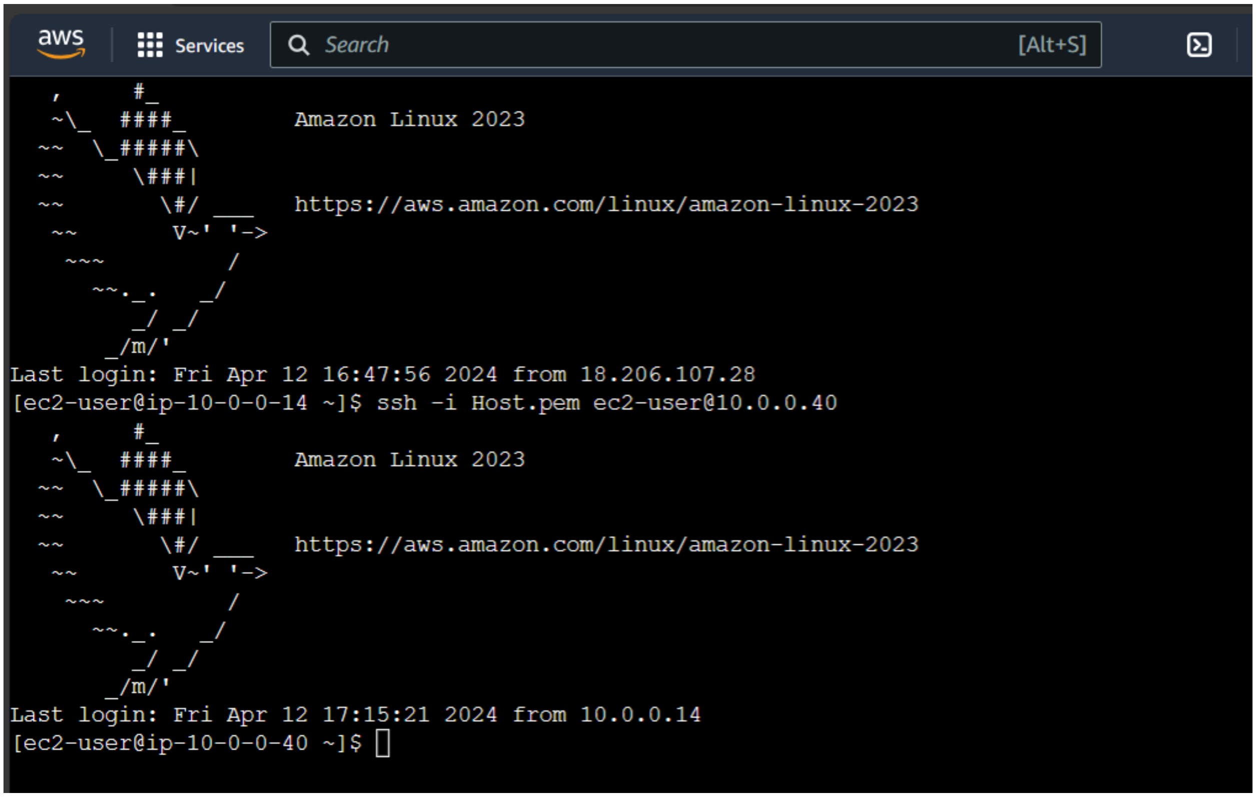


Elastic IP addresses (3)

Find resources by attribute or tag

□	Name	Allocated IPv4 addr...	Type
□	NAT-GW-1	34.202.2.70	Public IP
□	Bastion Host	34.226.77.122	Public IP
□	NAT-GW-2	54.80.217.228	Public IP

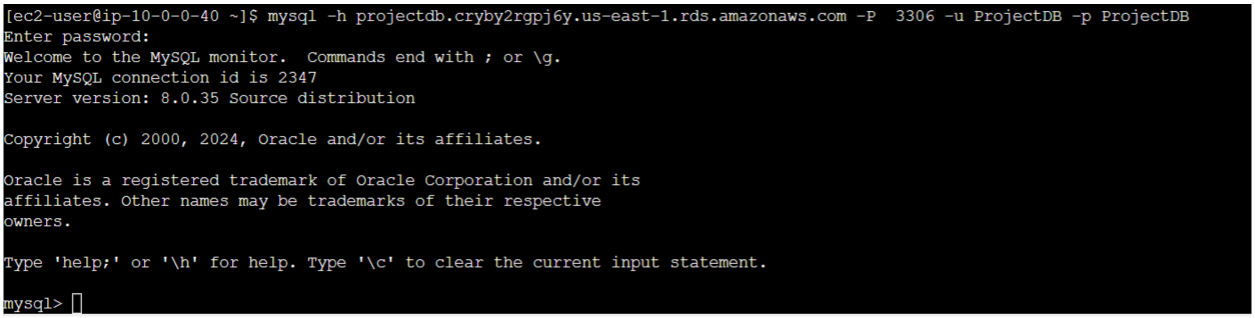
Next, I connected to the Bastion host and created a file containing the private key, then connected to the application server using SSH.



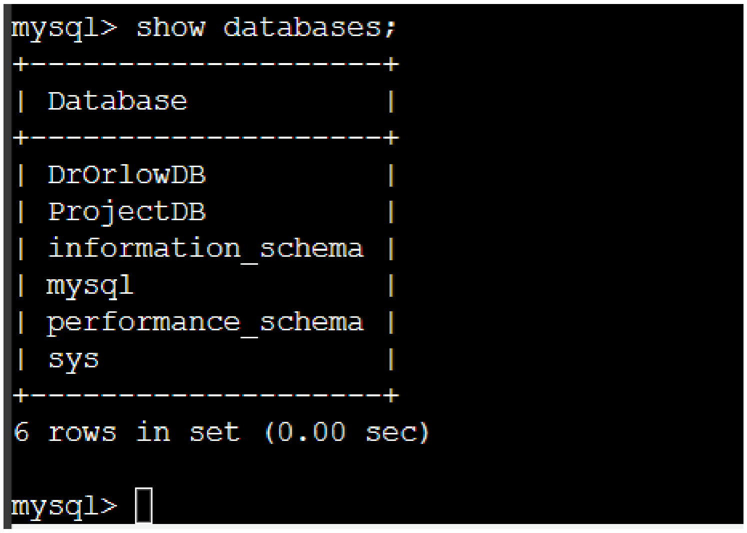
```
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Fri Apr 12 16:47:56 2024 from 18.206.107.28
[ec2-user@ip-10-0-0-14 ~]$ ssh -i Host.pem ec2-user@10.0.0.40
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Fri Apr 12 17:15:21 2024 from 10.0.0.14
[ec2-user@ip-10-0-0-40 ~]$ 
```

First, I downloaded MySQL and connected to the database using these commands:

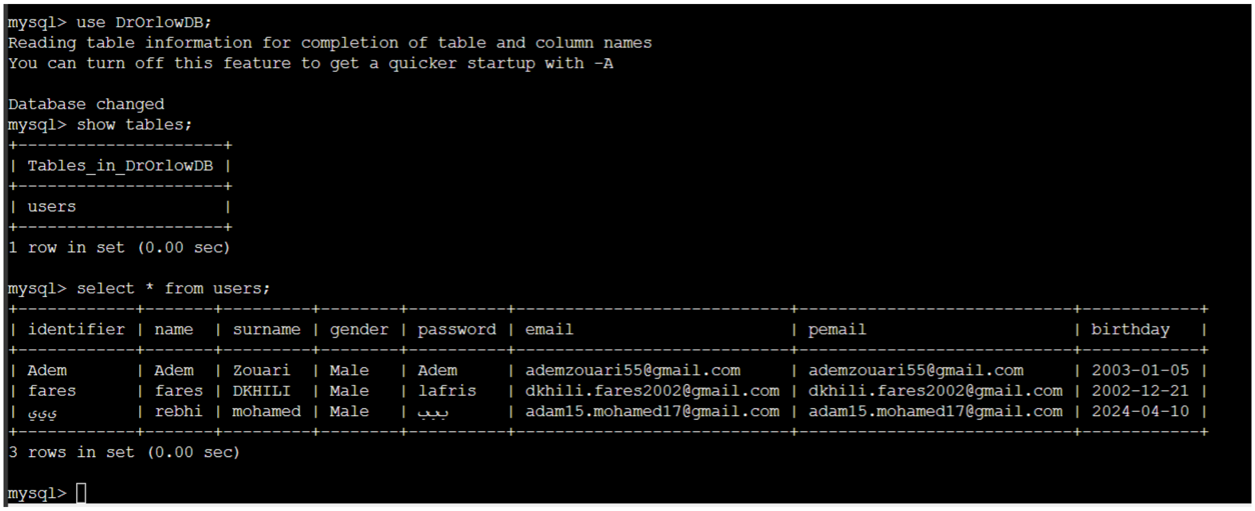
```
-install mysql

-sudo yum install https://dev.mysql.com/get/mysql80-community-release-el9-5.noarch.rpm

-sudo yum install mysql-community-server

-sudo systemctl enable --now mysqld

mysql -h projectdb.cryby2rgpj6y.us-east-1.rds.amazonaws.com -P  3306 -u ProjectDB -p ProjectDB
```



```
[ec2-user@ip-10-0-0-40 ~]$ mysql -h projectdb.cryby2rgpj6y.us-east-1.rds.amazonaws.com -P  3306 -u ProjectDB -p ProjectDB
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2347
Server version: 8.0.35 Source distribution

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> 
```



```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| DrOrlowDB          |
| ProjectDB          |
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
6 rows in set (0.00 sec)

mysql> 
```

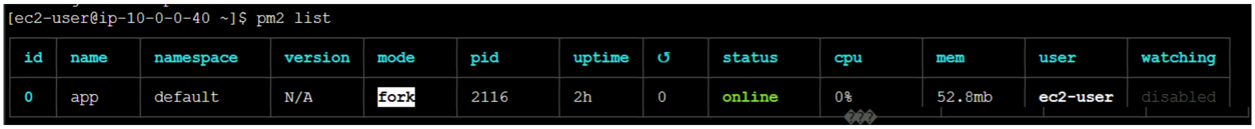


```
mysql> use DrOrlowDB;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+---------------------+
| Tables_in_DrOrlowDB |
+---------------------+
| users               |
+---------------------+
1 row in set (0.00 sec)

mysql> select * from users;
+------------+-------+---------+--------+----------+-----------------------------+-----------------------------+------------+
| identifier | name  | surname | gender | password | email                       | pemail                      | birthday   |
+------------+-------+---------+--------+----------+-----------------------------+-----------------------------+------------+
| Adem       | Adem  | Zouari  | Male   | Adem     | ademzouari55@gmail.com      | ademzouari55@gmail.com      | 2003-01-05 |
| fares      | fares | DKHILI  | Male   | lafris   | dkhili.fares2002@gmail.com | dkhili.fares2002@gmail.com | 2002-12-21 |
| ييي        | rebhi | mohamed | Male   | بيب      | adam15.mohamed17@gmail.com  | adam15.mohamed17@gmail.com  | 2024-04-10 |
+------------+-------+---------+--------+----------+-----------------------------+-----------------------------+------------+
3 rows in set (0.00 sec)

mysql> 
```

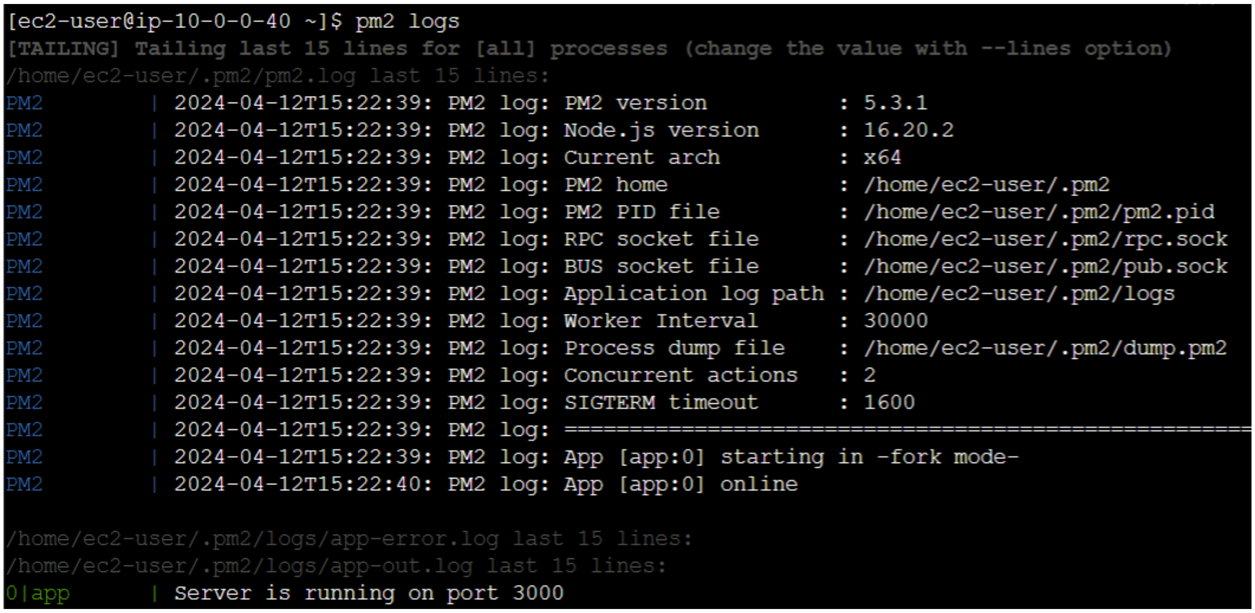
Then, I executed these commands to prepare the server.

```
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.38.0/install.sh | bash
source ~/.bashrc
nvm install 16
nvm use 16
npm install -g pm2
sudo yum install git
git clone https://github.com/Adem-Zouari/DrOrlow.git app-tier
cd app-tier
rm -r public
cd server
npm install mysql express body-parser
pm2 start app.js
pm2 save
```

I created a public GitHub repository containing the frontend and backend files, then downloaded the files from this repository.

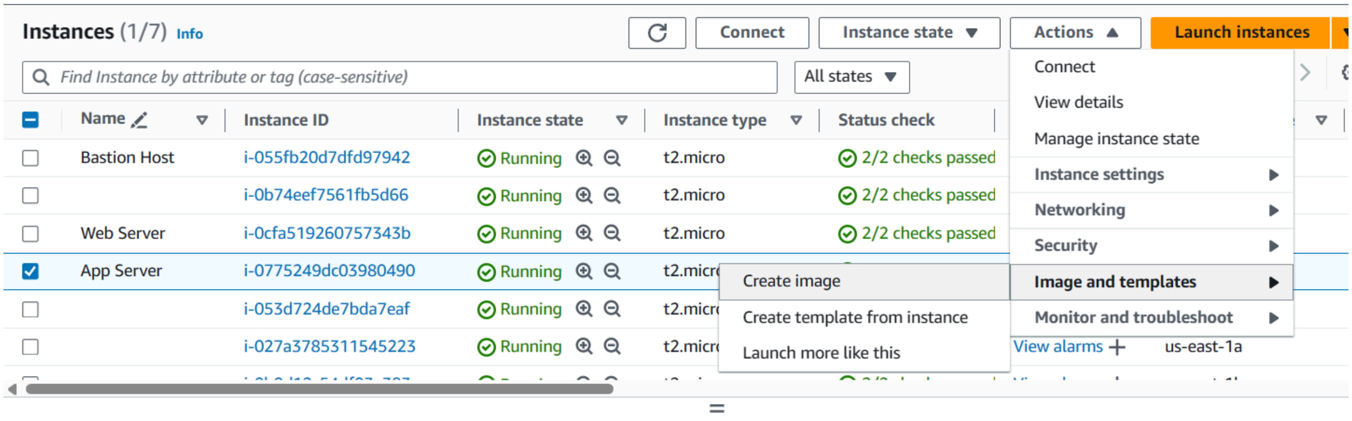


```
[ec2-user@ip-10-0-0-40 ~]$ pm2 list
│ id │ name │ namespace │ version │ mode │ pid  │ uptime │ ↺ │ status │ cpu │ mem    │ user     │ watching │
│ 0  │ app  │ default   │ N/A     │ fork │ 2116 │ 2h     │ 0 │ online │ 0%  │ 52.8mb │ ec2-user │ disabled │
```



```
[ec2-user@ip-10-0-0-40 ~]$ pm2 logs
[TAILING] Tailing last 15 lines for [all] processes (change the value with --lines option)
/home/ec2-user/.pm2/pm2.log last 15 lines:
PM2         | 2024-04-12T15:22:39: PM2 log: PM2 version          : 5.3.1
PM2         | 2024-04-12T15:22:39: PM2 log: Node.js version      : 16.20.2
PM2         | 2024-04-12T15:22:39: PM2 log: Current arch         : x64
PM2         | 2024-04-12T15:22:39: PM2 log: PM2 home             : /home/ec2-user/.pm2
PM2         | 2024-04-12T15:22:39: PM2 log: PM2 PID file         : /home/ec2-user/.pm2/pm2.pid
PM2         | 2024-04-12T15:22:39: PM2 log: RPC socket file      : /home/ec2-user/.pm2/rpc.sock
PM2         | 2024-04-12T15:22:39: PM2 log: BUS socket file      : /home/ec2-user/.pm2/pub.sock
PM2         | 2024-04-12T15:22:39: PM2 log: Application log path : /home/ec2-user/.pm2/logs
PM2         | 2024-04-12T15:22:39: PM2 log: Worker Interval      : 30000
PM2         | 2024-04-12T15:22:39: PM2 log: Process dump file    : /home/ec2-user/.pm2/dump.pm2
PM2         | 2024-04-12T15:22:39: PM2 log: Concurrent actions   : 2
PM2         | 2024-04-12T15:22:39: PM2 log: SIGTERM timeout      : 1600
PM2         | 2024-04-12T15:22:39: PM2 log: ==========================================================
PM2         | 2024-04-12T15:22:39: PM2 log: App [app:0] starting in -fork mode-
PM2         | 2024-04-12T15:22:40: PM2 log: App [app:0] online

/home/ec2-user/.pm2/logs/app-error.log last 15 lines:
/home/ec2-user/.pm2/logs/app-out.log last 15 lines:
0|app       | Server is running on port 3000
```

For the next step, I created an image of the application server.



	Name	Instance ID	Instance state	Instance type	Status check
□	Bastion Host	i-055fb20d7dfd97942	Running	t2.micro	2/2 checks passed
□		i-0b74eef7561fb5d66	Running	t2.micro	2/2 checks passed
□	Web Server	i-0cfa519260757343b	Running	t2.micro	2/2 checks passed
☑	App Server	i-0775249dc03980490	Running	t2.micr	
□		i-053d724de7bda7eaf	Running	t2.micr	
□		i-027a3785311545223	Running	t2.micr	

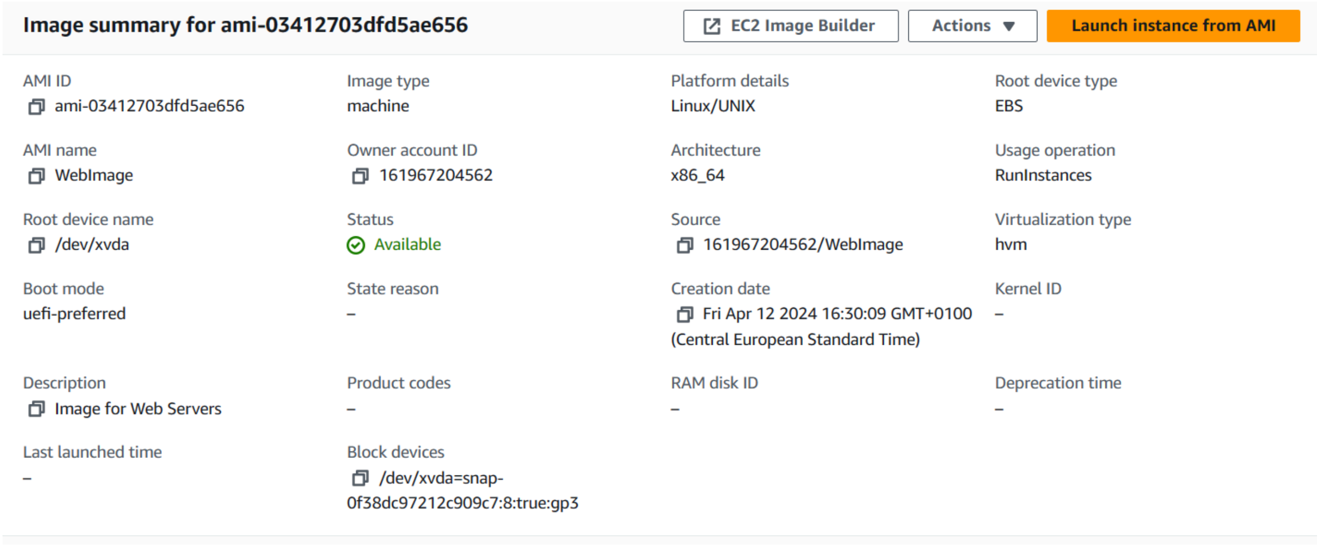


Image summary for ami-03412703dfd5ae656

AMI ID
ami-03412703dfd5ae656

Image type
machine

Platform details
Linux/UNIX

Root device type
EBS

AMI name
WebImage

Owner account ID
161967204562

Architecture
x86_64

Usage operation
RunInstances

Root device name
/dev/xvda

Status
Available

Source
161967204562/WebImage

Virtualization type
hvm

Boot mode
uefi-preferred

State reason
–

Creation date
Fri Apr 12 2024 16:30:09 GMT+0100 (Central European Standard Time)

Kernel ID
–

Description
Image for Web Servers

Product codes
–

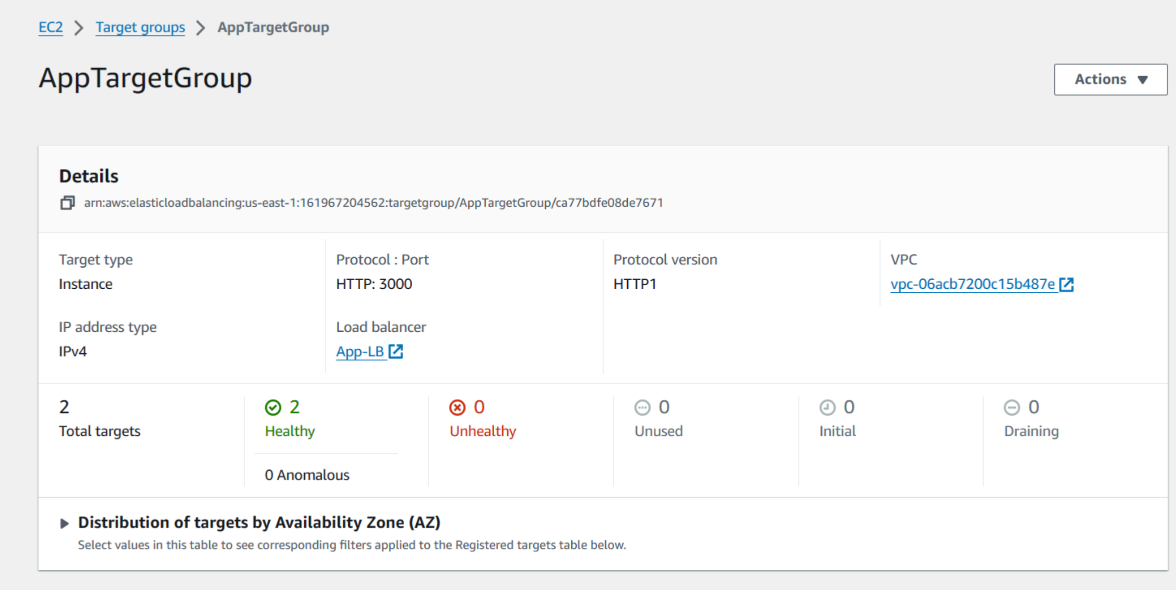
RAM disk ID
–

Deprecation time
–

Last launched time
–

Block devices
/dev/xvda=snap-0f38dc97212c909c7:8:true:gp3

Then, the AppTargetGroup.



AppTargetGroup

Details

arn:aws:elasticloadbalancing:us-east-1:161967204562:targetgroup/AppTargetGroup/ca77bdfe08de7671

Target type
Instance

Protocol : Port
HTTP: 3000

Protocol version
HTTP1

VPC
vpc-06acb7200c15b487e

IP address type
IPv4

Load balancer
App-LB

2
Total targets

✓ 2
Healthy

0 Anomalous

⊗ 0
Unhealthy

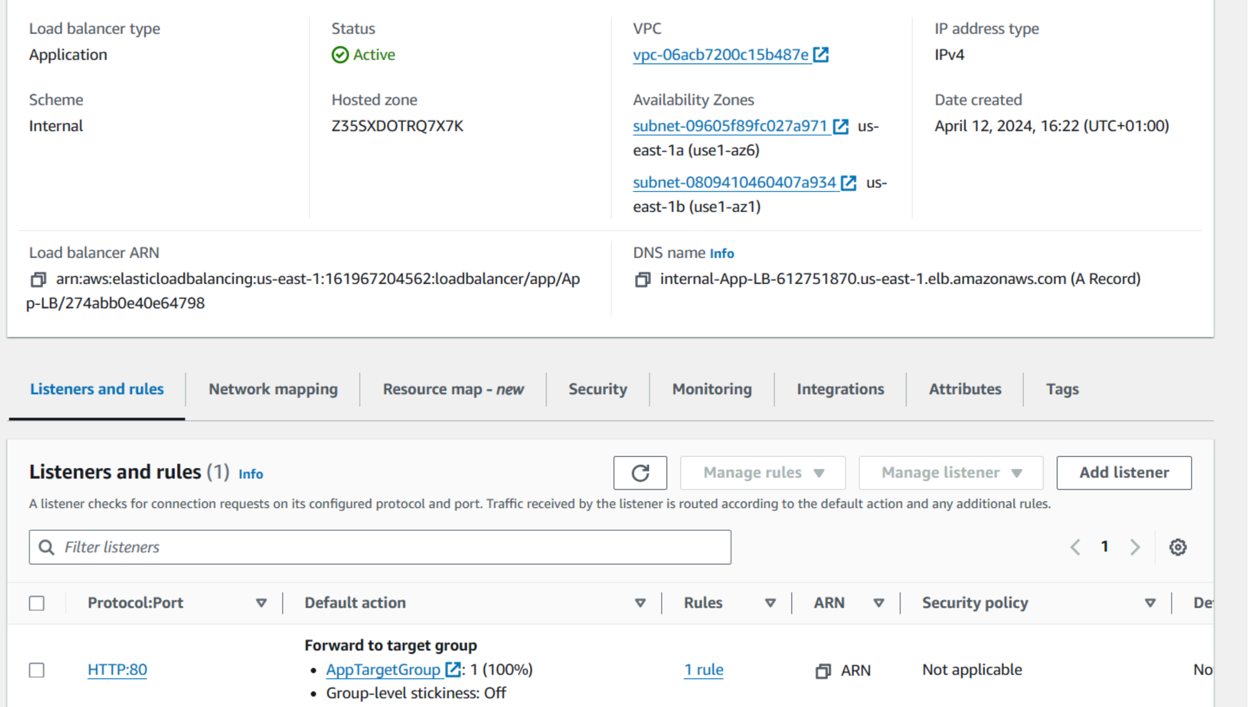
0
Unused

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.

Then, the internal load balancer.



Load balancer type
Application

Status
Active

VPC
vpc-06acb7200c15b487e

IP address type
IPv4

Scheme
Internal

Hosted zone
Z35SXDOTRQ7X7K

Availability Zones
subnet-09605f89fc027a971 us-east-1a (use1-az6)
subnet-0809410460407a934 us-east-1b (use1-az1)

Date created
April 12, 2024, 16:22 (UTC+01:00)

Load balancer ARN
arn:aws:elasticloadbalancing:us-east-1:161967204562:loadbalancer/app/App-LB/274abb0e40e64798

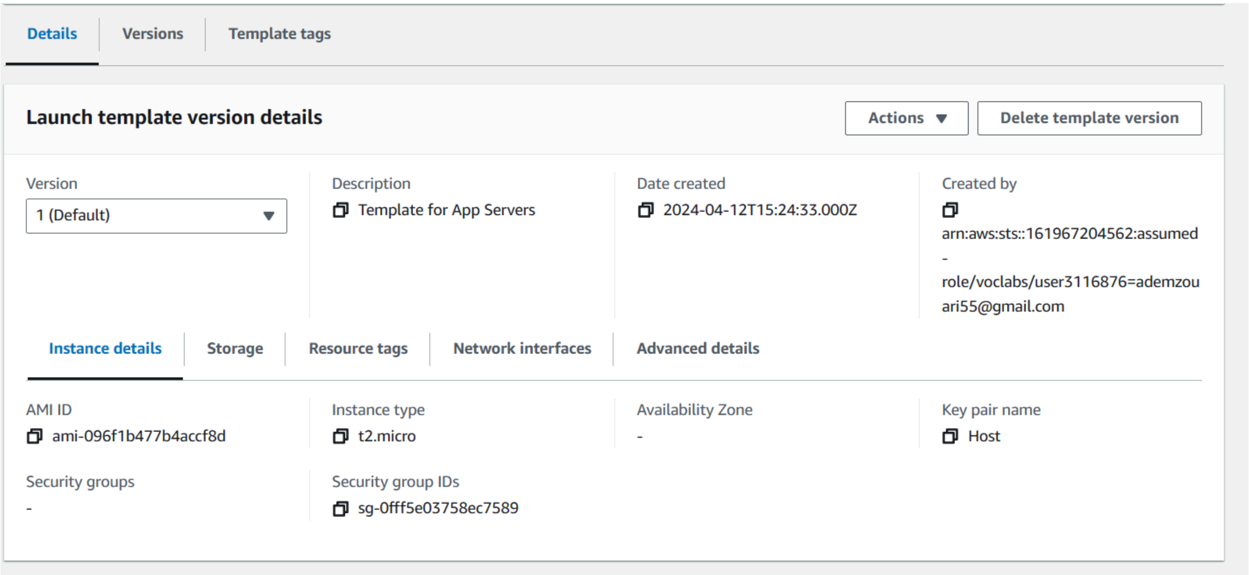
DNS name Info
internal-App-LB-612751870.us-east-1.elb.amazonaws.com (A Record)

Listeners and rules (1) Info

A listener checks for connection requests on its configured protocol and port. Traffic received by the listener is routed according to the default action and any additional rules.

□	Protocol:Port	Default action	Rules	ARN	Security policy	De
□	HTTP:80	**Forward to target group** • AppTargetGroup: 1 (100%) • Group-level stickiness: Off	1 rule	ARN	Not applicable	No

Then, I created AppTemplate using the image.



Launch template version details

Version
1 (Default)

Description
Template for App Servers

Date created
2024-04-12T15:24:33.000Z

Created by
arn:aws:sts::161967204562:assumed-role/voclabs/user3116876=ademzouari55@gmail.com

AMI ID
ami-096f1b477b4accf8d

Instance type
t2.micro

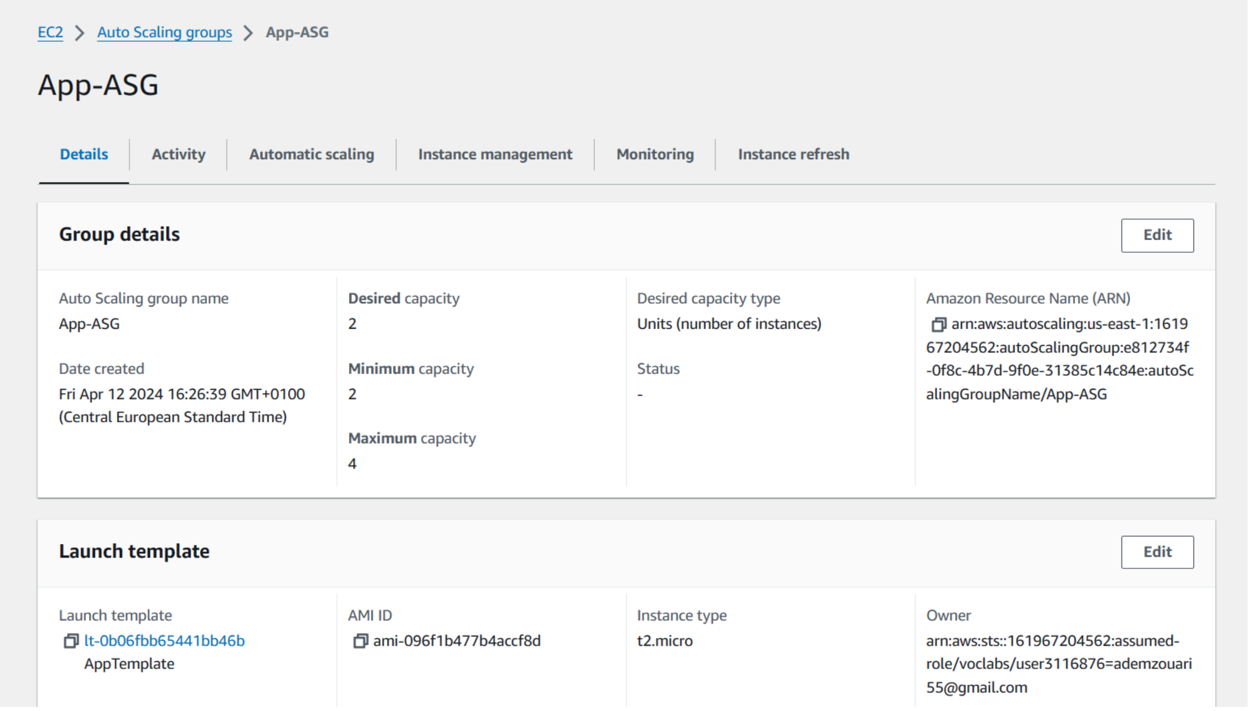
Availability Zone
-

Key pair name
Host

Security groups
-

Security group IDs
sg-0fff5e03758ec7589

And finally: App Auto Scaling Group



App-ASG

Group details

Auto Scaling group name
App-ASG

Date created
Fri Apr 12 2024 16:26:39 GMT+0100 (Central European Standard Time)

Desired capacity
2

Minimum capacity
2

Maximum capacity
4

Desired capacity type
Units (number of instances)

Status
-

Amazon Resource Name (ARN)
arn:aws:autoscaling:us-east-1:161967204562:autoScalingGroup:e812734f-0f8c-4b7d-9f0e-31385c14c84e:autoScalingGroupName/App-ASG

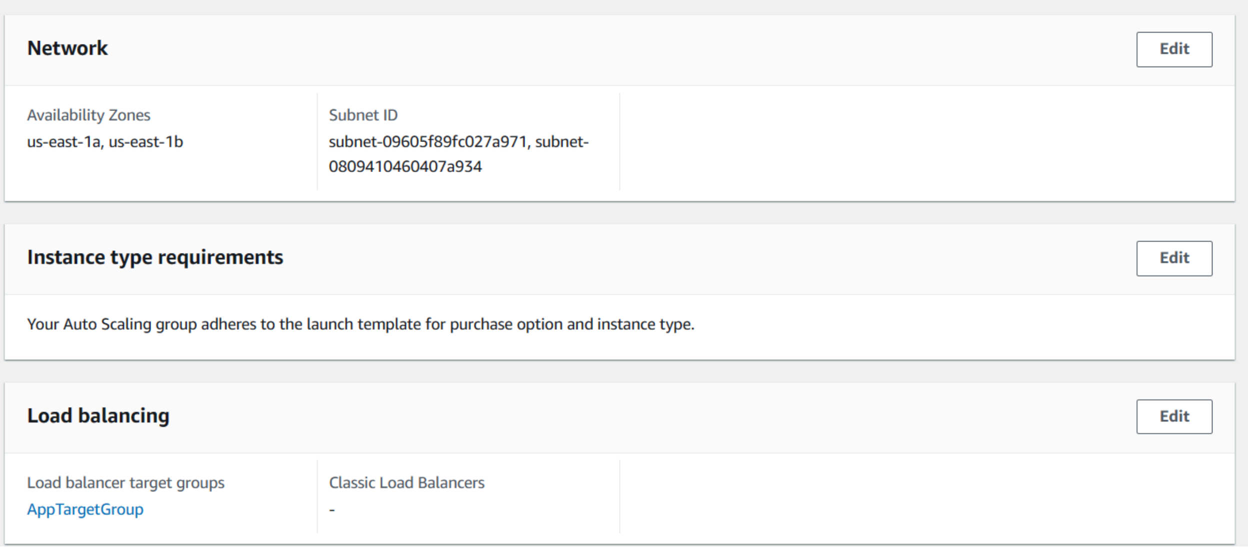
Launch template

Launch template
lt-0b06fbb65441bb46b
AppTemplate

AMI ID
ami-096f1b477b4accf8d

Instance type
t2.micro

Owner
arn:aws:sts::161967204562:assumed-role/voclabs/user3116876=ademzouari55@gmail.com



Network

Availability Zones
us-east-1a, us-east-1b

Subnet ID
subnet-09605f89fc027a971, subnet-0809410460407a934

Instance type requirements

Your Auto Scaling group adheres to the launch template for purchase option and instance type.

Load balancing

Load balancer target groups
AppTargetGroup

Classic Load Balancers
-

Then, I proceeded to the next step.

Web Server

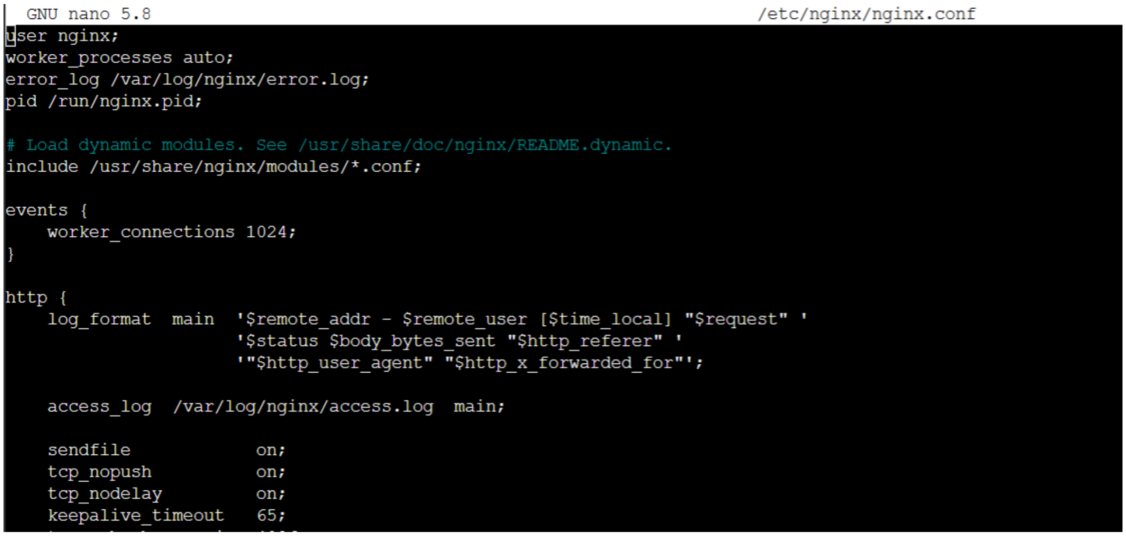
I connected to the web server using the Bastion host, then entered these commands:

```
-curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.38.0/install.sh | bash

-source ~/.bashrc

-nvm install 16

-nvm use 16
```

```
-sudo yum install git
-git clone https://github.com/Adem-Zouari/DrOrlow.git web-tier
-cd web-tier
-rm -r server
-sudo yum install nginx
-sudo truncate -s 0 /etc/nginx/nginx.conf
-chmod -R 755 /home/ec2-user
-sudo chkconfig nginx on
```

Then, I used the command `sudo nano /etc/nginx/nginx.conf` and added a health check and a submit endpoint :

```
        #health check
        location /health {
            default_type text/html;
            return 200 "<!DOCTYPE html><p>Web Tier Health Check</p>\n";
        }
        #front end files
        location / {
            root    /home/ec2-user/web-tier/public;
            index Home.html "About me.html";
            try_files $uri /Home.html;
        }
     # New location block to handle submit endpoint
        location /submit {
            proxy_pass http://internal-App-LB-612751870.us-east-1.elb.amazonaws.com:80;
            proxy_set_header Host $host;
            proxy_set_header X-Real-IP $remote_addr;
            proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
            proxy_set_header X-Forwarded-Proto $scheme;
           }
    }
```

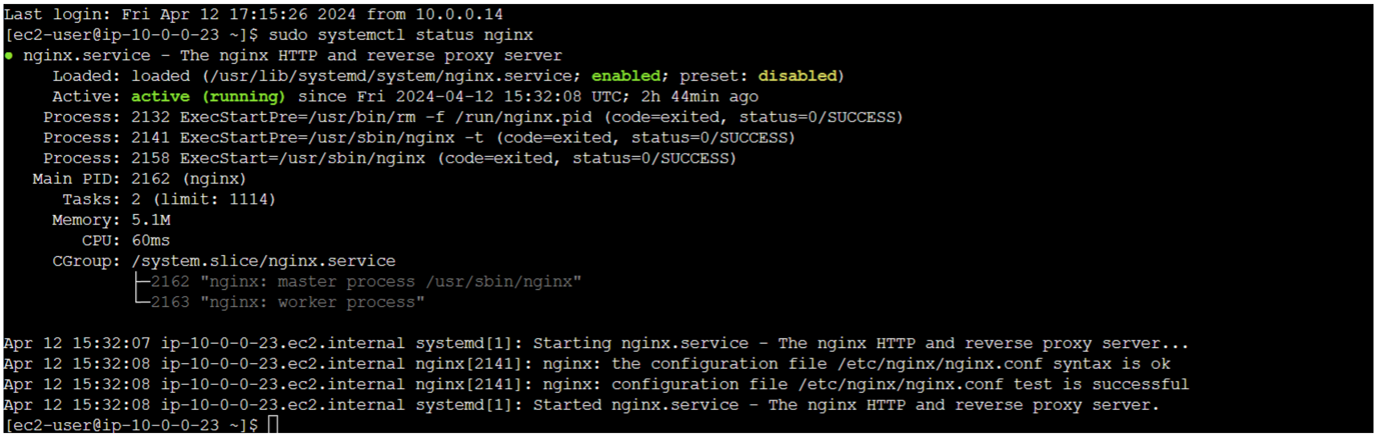


```
  GNU nano 5.8                                                   /etc/nginx/nginx.conf
user nginx;
worker_processes auto;
error_log /var/log/nginx/error.log;
pid /run/nginx.pid;

# Load dynamic modules. See /usr/share/doc/nginx/README.dynamic.
include /usr/share/nginx/modules/*.conf;

events {
    worker_connections 1024;
}

http {
    log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
                      '$status $body_bytes_sent "$http_referer" '
                      '"$http_user_agent" "$http_x_forwarded_for"';

    access_log  /var/log/nginx/access.log  main;

    sendfile            on;
    tcp_nopush          on;
    tcp_nodelay         on;
    keepalive_timeout   65;
```

And then

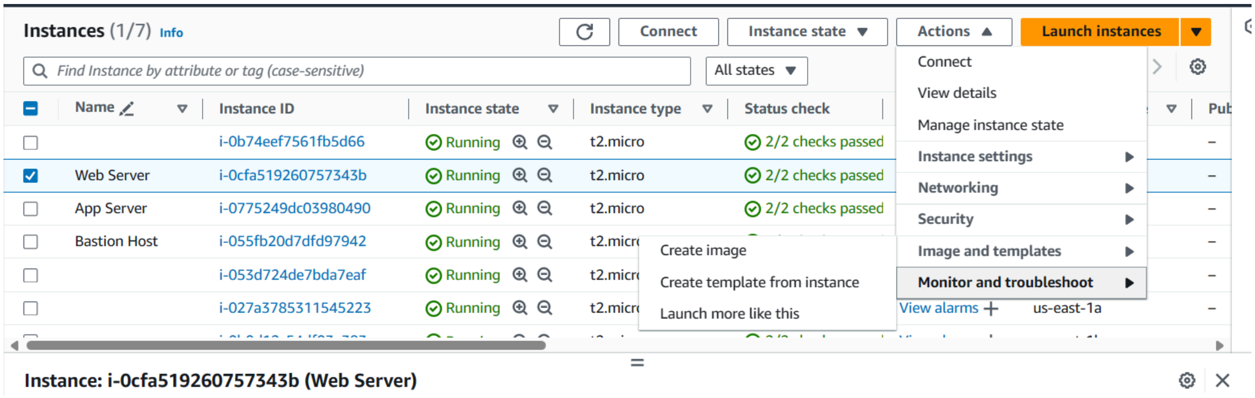
-sudo systemctl start nginx

-sudo systemctl status nginx



```
Last login: Fri Apr 12 17:15:26 2024 from 10.0.0.14
[ec2-user@ip-10-0-0-23 ~]$ sudo systemctl status nginx
● nginx.service - The nginx HTTP and reverse proxy server
     Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: disabled)
     Active: active (running) since Fri 2024-04-12 15:32:08 UTC; 2h 44min ago
    Process: 2132 ExecStartPre=/usr/bin/rm -f /run/nginx.pid (code=exited, status=0/SUCCESS)
    Process: 2141 ExecStartPre=/usr/sbin/nginx -t (code=exited, status=0/SUCCESS)
    Process: 2158 ExecStart=/usr/sbin/nginx (code=exited, status=0/SUCCESS)
   Main PID: 2162 (nginx)
      Tasks: 2 (limit: 1114)
     Memory: 5.1M
        CPU: 60ms
     CGroup: /system.slice/nginx.service
             ├─2162 "nginx: master process /usr/sbin/nginx"
             └─2163 "nginx: worker process"

Apr 12 15:32:07 ip-10-0-0-23.ec2.internal systemd[1]: Starting nginx.service - The nginx HTTP and reverse proxy server...
Apr 12 15:32:08 ip-10-0-0-23.ec2.internal nginx[2141]: nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
Apr 12 15:32:08 ip-10-0-0-23.ec2.internal nginx[2141]: nginx: configuration file /etc/nginx/nginx.conf test is successful
Apr 12 15:32:08 ip-10-0-0-23.ec2.internal systemd[1]: Started nginx.service - The nginx HTTP and reverse proxy server.
[ec2-user@ip-10-0-0-23 ~]$ 
```

And finally, I set up the Web Auto Scaling Group and internet-facing load balancer using the image and WebTemplate for the web servers.



	Name	Instance ID	Instance state	Instance type	Status check			Pub
□		i-0b74eef7561fb5d66	Running	t2.micro	2/2 checks passed			–
☑	Web Server	i-0cfa519260757343b	Running	t2.micro	2/2 checks passed			–
□	App Server	i-0775249dc03980490	Running	t2.micro	2/2 checks passed			–
□	Bastion Host	i-055fb20d7dfd97942	Running	t2.micro				–
□		i-053d724de7bda7eaf	Running	t2.micro				–
□		i-027a3785311545223	Running	t2.micro		View alarms +	us-east-1a	–

Instance: i-0cfa519260757343b (Web Server)

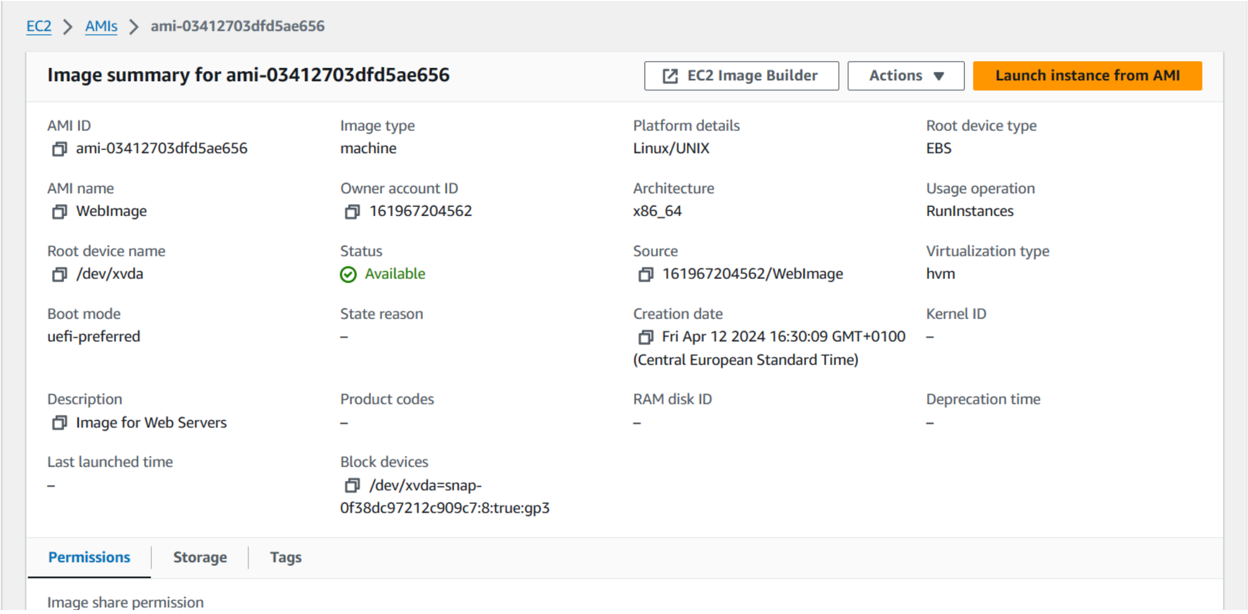


Image summary for ami-03412703dfd5ae656

AMI ID
ami-03412703dfd5ae656

Image type
machine

Platform details
Linux/UNIX

Root device type
EBS

AMI name
WebImage

Owner account ID
161967204562

Architecture
x86_64

Usage operation
RunInstances

Root device name
/dev/xvda

Status
Available

Source
161967204562/WebImage

Virtualization type
hvm

Boot mode
uefi-preferred

State reason
–

Creation date
Fri Apr 12 2024 16:30:09 GMT+0100 (Central European Standard Time)

Kernel ID
–

Description
Image for Web Servers

Product codes
–

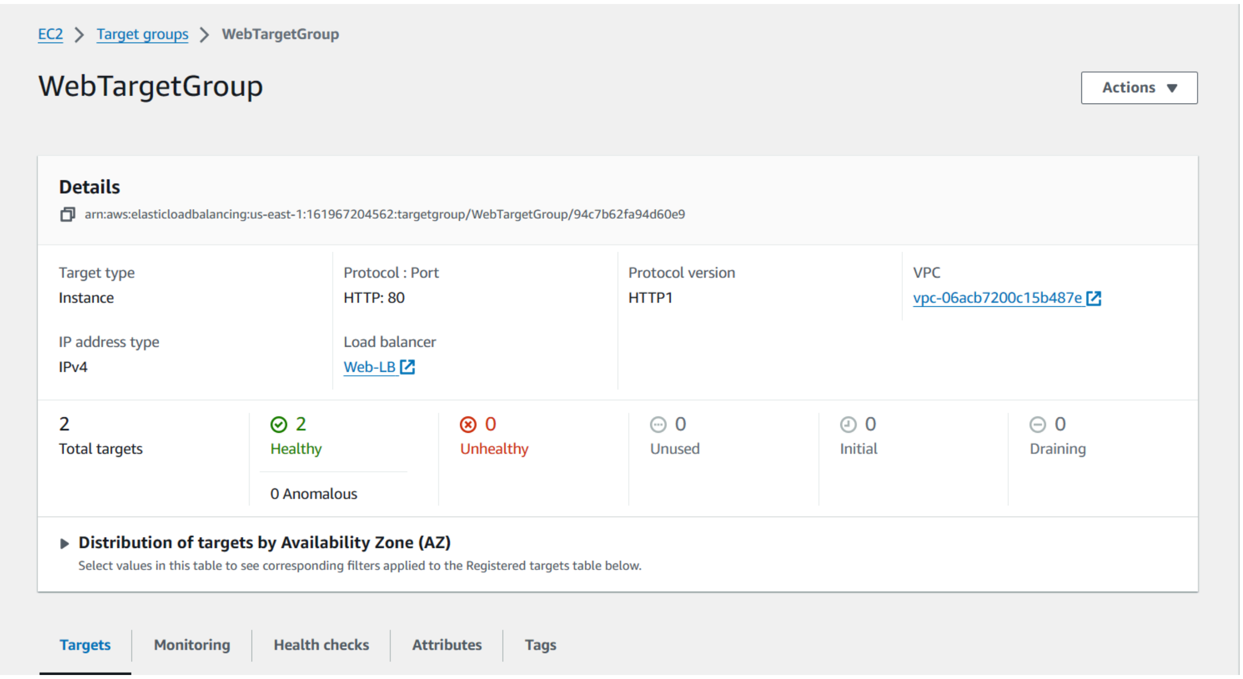
RAM disk ID
–

Deprecation time
–

Last launched time
–

Block devices
/dev/xvda=snap-0f38dc97212c909c7:8:true:gp3

Image share permission


EC2 > Target groups > WebTargetGroup
Actions ▼
Distribution of targets by Availability Zone (AZ)
Targets
Monitoring
Health checks
Attributes
Tags

WebTargetGroup

Details
arn:aws:elasticloadbalancing:us-east-1:161967204562:targetgroup/WebTargetGroup/94c7b62fa94d60e9

Target type
Instance

Protocol : Port
HTTP: 80

Protocol version
HTTP1

VPC
vpc-06acb7200c15b487e

IP address type
IPv4

Load balancer
Web-LB

2 Total targets

2 Healthy

0 Anomalous

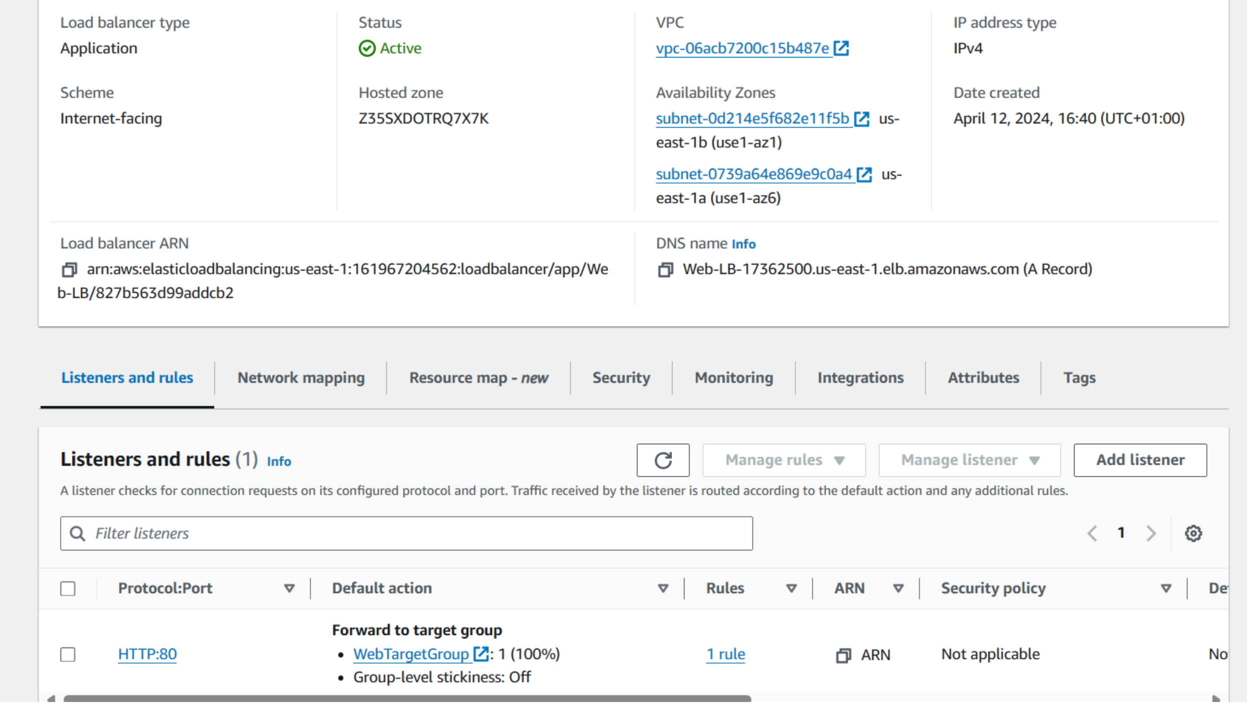
0 Unhealthy

0 Unused

0 Initial

0 Draining

Select values in this table to see corresponding filters applied to the Registered targets table below.



Load balancer type
Application

Status
Active

VPC
vpc-06acb7200c15b487e

IP address type
IPv4

Scheme
Internet-facing

Hosted zone
Z35SXDOTRQ7X7K

Availability Zones
subnet-0d214e5f682e11f5b us-east-1b (use1-az1)
subnet-0739a64e869e9c0a4 us-east-1a (use1-az6)

Date created
April 12, 2024, 16:40 (UTC+01:00)

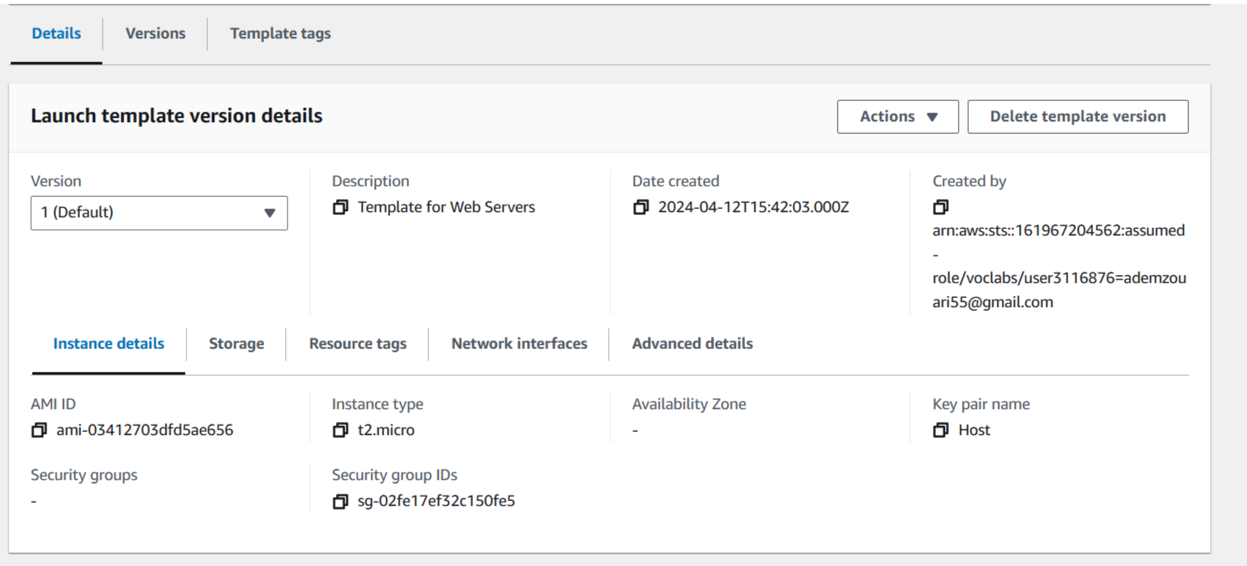
Load balancer ARN
arn:aws:elasticloadbalancing:us-east-1:161967204562:loadbalancer/app/Web-LB/827b563d99addcb2

DNS name Info
Web-LB-17362500.us-east-1.elb.amazonaws.com (A Record)

Listeners and rules (1) Info

A listener checks for connection requests on its configured protocol and port. Traffic received by the listener is routed according to the default action and any additional rules.

□	Protocol:Port	Default action	Rules	ARN	Security policy	De
□	HTTP:80	**Forward to target group** • WebTargetGroup: 1 (100%) • Group-level stickiness: Off	1 rule	ARN	Not applicable	No



Launch template version details

Version
1 (Default)

Description
Template for Web Servers

Date created
2024-04-12T15:42:03.000Z

Created by
arn:aws:sts::161967204562:assumed-role/voclabs/user3116876=ademzouari55@gmail.com

AMI ID
ami-03412703dfd5ae656

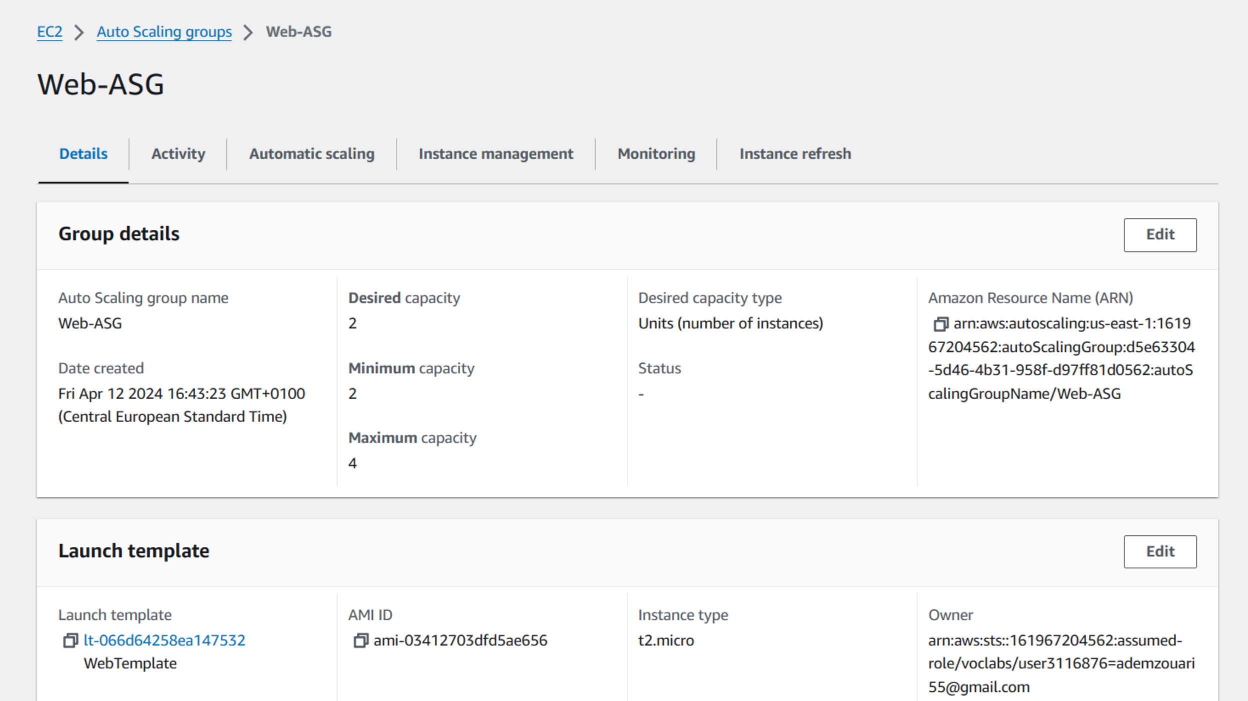
Instance type
t2.micro

Availability Zone
-

Key pair name
Host

Security groups
-

Security group IDs
sg-02fe17ef32c150fe5



Web-ASG

Group details

Auto Scaling group name
Web-ASG

Date created
Fri Apr 12 2024 16:43:23 GMT+0100 (Central European Standard Time)

Desired capacity
2

Minimum capacity
2

Maximum capacity
4

Desired capacity type
Units (number of instances)

Status
-

Amazon Resource Name (ARN)
arn:aws:autoscaling:us-east-1:161967204562:autoScalingGroup:d5e63304-5d46-4b31-958f-d97ff81d0562:autoScalingGroupName/Web-ASG

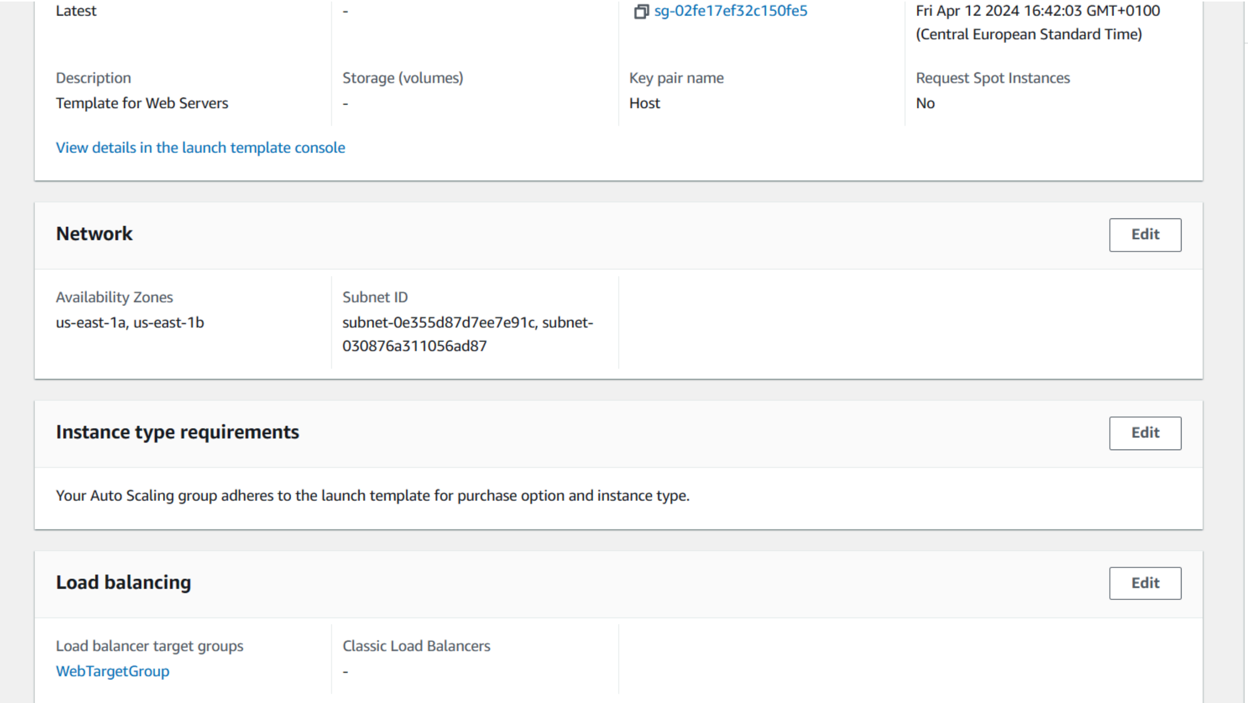
Launch template

Launch template
lt-066d64258ea147532
WebTemplate

AMI ID
ami-03412703dfd5ae656

Instance type
t2.micro

Owner
arn:aws:sts::161967204562:assumed-role/voclabs/user3116876=ademzouari55@gmail.com


Latest

-

sg-02fe17ef32c150fe5

Fri Apr 12 2024 16:42:03 GMT+0100 (Central European Standard Time)

Description
Template for Web Servers

Storage (volumes)
-

Key pair name
Host

Request Spot Instances
No

View details in the launch template console

Network

Availability Zones
us-east-1a, us-east-1b

Subnet ID
subnet-0e355d87d7ee7e91c, subnet-030876a311056ad87

Instance type requirements

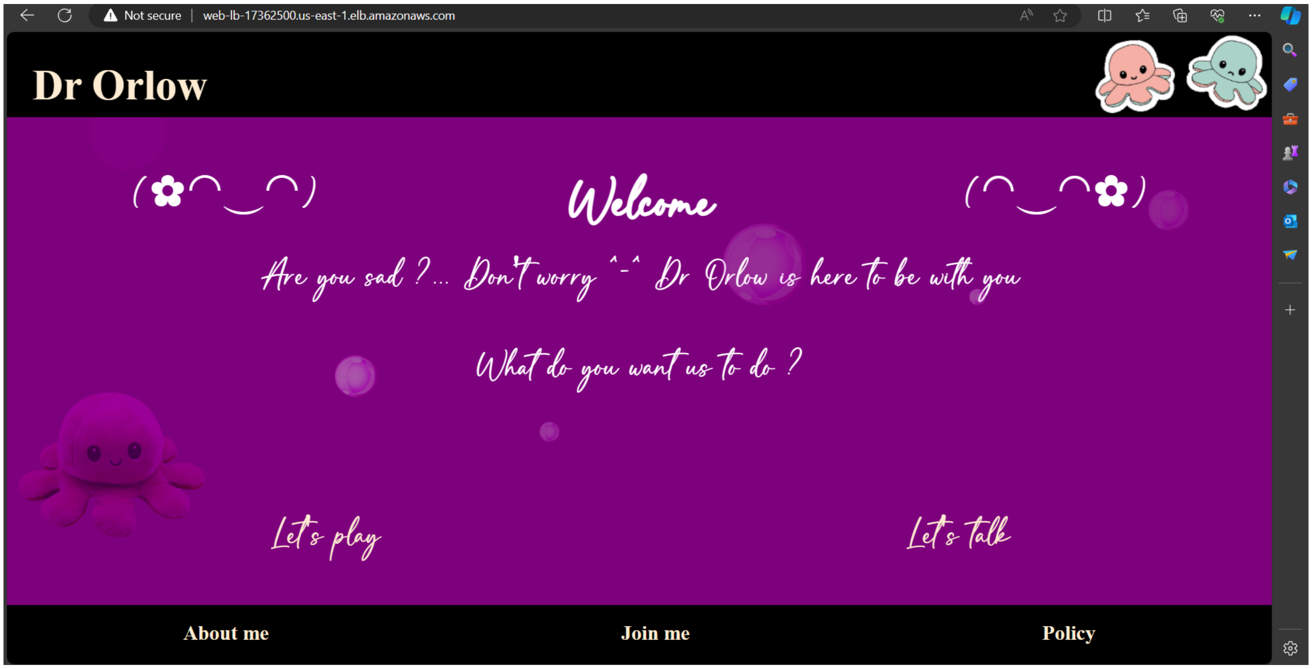
Your Auto Scaling group adheres to the launch template for purchase option and instance type.

Load balancing

Load balancer target groups
WebTargetGroup

Classic Load Balancers
-

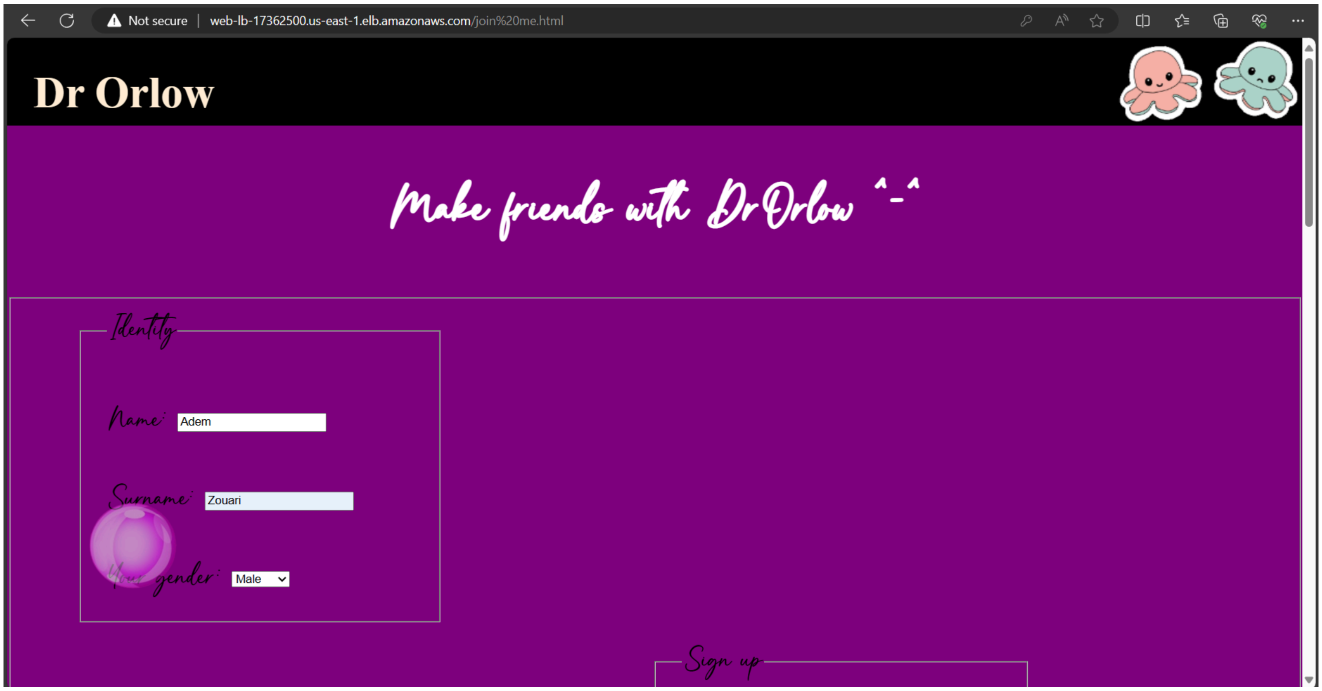
Finally, here is the website.



Welcome

Are you sad ?... Don't worry ^-^ Dr Orlow is here to be with you

What do you want us to do ?


Not secure | web-lb-17362500.us-east-1.elb.amazonaws.com/join%20me.html
Dr Orlow
Sign up

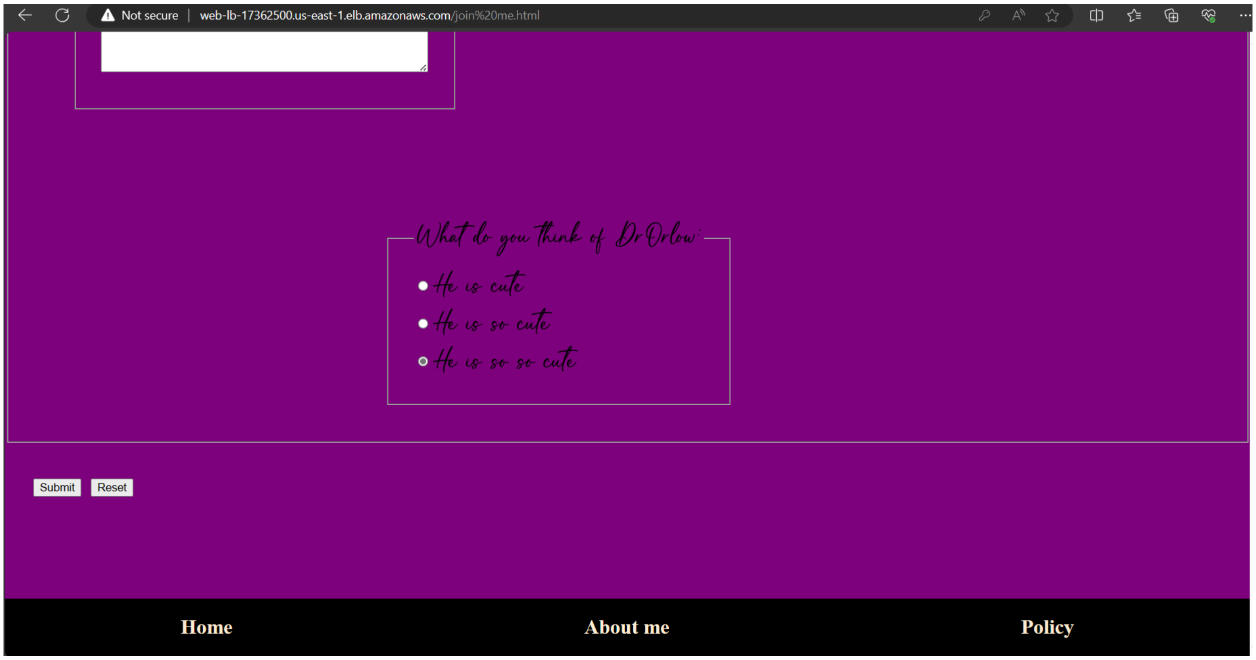
Make friends with DrOrlow ^-^

Identity

Name: Adem

Surname: Zouari

Your gender: Male



What do you think of Dr Orlow:

○ He is cute

○ He is so cute

○ He is so so cute

Home About me Policy