

Ministry of higher education
and scientific research

University of El Manar

Faculty of Sciences of Tunis



وزارة التعليم العالي و البحث العلمي

جامعة تونس المنار

كلية العلوم بتونس

AWS Project Report

Deployment of a 3-Tier Application

Work developed by:

AROUA Rahma

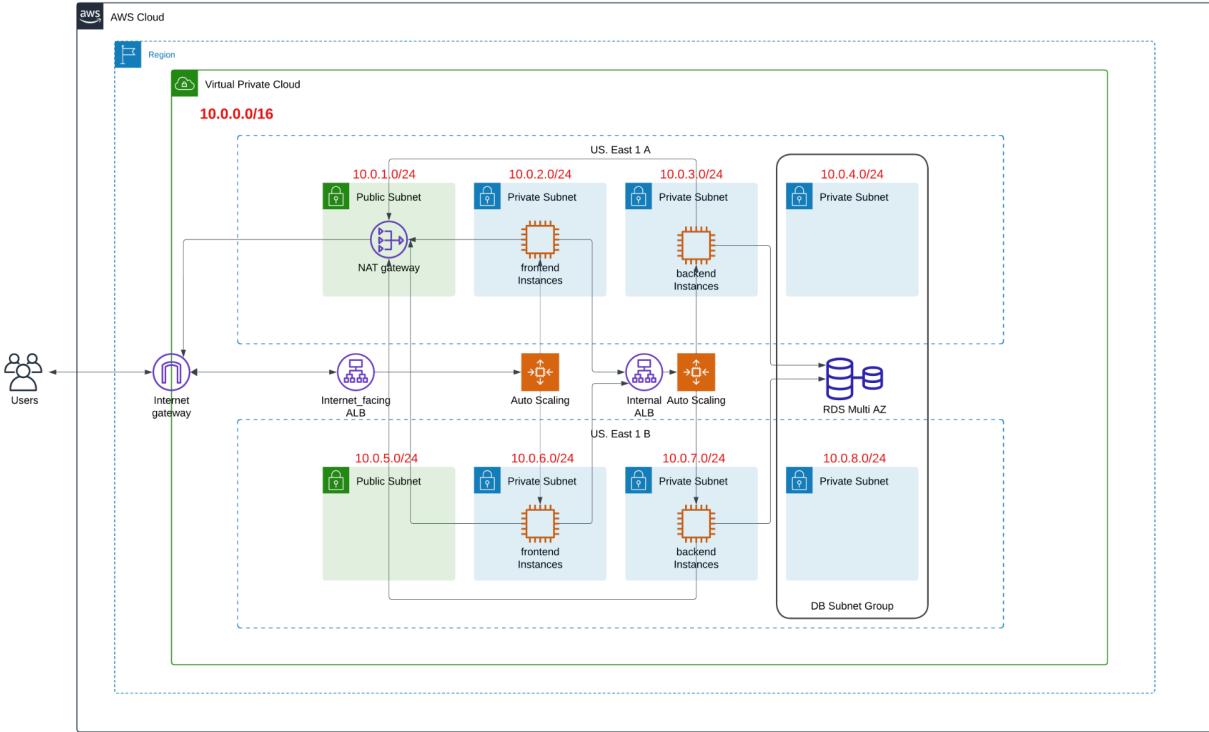
BEN SLIMANE Yasmine

Supervised by:

M. ABBES Haithem

Academic year : 2024/2025

Phase 0: Project architecture



Phase 1: Network Infrastructure

1. VPC Creation:

- A VPC was configured with a CIDR block of **10.0.0.0/16**

VPC Creation:

Your VPCs (1/2) Info							Last updated 7 minutes ago	Actions
Name		VPC ID	State	Block Public...	IPv4 CIDR	IPv6 CIDR		
<input type="checkbox"/>	-	vpc-037ab9e73fa52c4ca	Available	<input type="radio"/> Off	172.31.0.0/16	-		
<input checked="" type="checkbox"/>	awsproject-vpc	vpc-0b6fdbbb4a6bca32b	Available	<input type="radio"/> Off	10.0.0.0/16	-		

- Public and private subnets were then created in **at least two Availability Zones (AZs)** to ensure high availability.

Subnet Creation:

Subnets (8) Info							Last updated 3 minutes ago	Actions	Create subnet
Name		Subnet ID	State	VPC	Block Public...	IPv4 CIDR	IPv6 CIDR		
<input type="checkbox"/>	public-us-east-1a	subnet-070ef4aa9b1fad6fb	Available	vpc-0b6fdbbb4a6bca32b aws...	<input type="radio"/> Off	10.0.1.0/24	-		
<input type="checkbox"/>	frontend-us-east-1a	subnet-05346dc767b4556f6	Available	vpc-0b6fdbbb4a6bca32b aws...	<input type="radio"/> Off	10.0.2.0/24	-		
<input type="checkbox"/>	backend-us-east-1a	subnet-044aa7bf6adc48e96	Available	vpc-0b6fdbbb4a6bca32b aws...	<input type="radio"/> Off	10.0.3.0/24	-		
<input type="checkbox"/>	rds-us-east-1a	subnet-03cf2aa953af28ef9	Available	vpc-0b6fdbbb4a6bca32b aws...	<input type="radio"/> Off	10.0.4.0/24	-		
<input type="checkbox"/>	public-us-east-1b	subnet-027e18a2dda396ea4	Available	vpc-0b6fdbbb4a6bca32b aws...	<input type="radio"/> Off	10.0.5.0/24	-		
<input type="checkbox"/>	frontend-us-east-1b	subnet-0e6263290ff8ae235	Available	vpc-0b6fdbbb4a6bca32b aws...	<input type="radio"/> Off	10.0.6.0/24	-		
<input type="checkbox"/>	backend-us-east-1b	subnet-031d5fb02ac4a5691	Available	vpc-0b6fdbbb4a6bca32b aws...	<input type="radio"/> Off	10.0.7.0/24	-		
<input type="checkbox"/>	rds-us-east-1b	subnet-0a3af96499360fa80	Available	vpc-0b6fdbbb4a6bca32b aws...	<input type="radio"/> Off	10.0.8.0/24	-		

For the 2 public subnets, we enabled the following property:

Auto-assign IP settings [Info](#)

Enable AWS to automatically assign a public IPv4 or IPv6 address to a new primary network interface for an instance in this subnet.

Enable auto-assign public IPv4 address [Info](#)

Enable auto-assign customer-owned IPv4 address [Info](#)
Option disabled because no customer owned pools found.

2. Network Component Configuration:

- An **Internet Gateway** was created to allow public resources to access the Internet.

Internet Gateway Creation and Attachment to Our VPC:

The screenshot shows a CloudFormation stack named "igw-0e9f3e70aba335c5f / awsproject-internet-gw". A green status bar at the top indicates that the Internet gateway was successfully attached to the VPC. The main details page shows the Internet gateway ID (igw-0e9f3e70aba335c5f), state (Attached), VPC ID (vpc-0b6fdbbb4a6bca32b), and owner (172615386837). The "Actions" button is visible in the top right corner.

- Then, we configured a **NAT Gateway** to allow instances in private subnets to access the Internet without being exposed.

To create the NAT Gateway, we assigned it an **Elastic IP**:

The screenshot shows a CloudFormation stack named "nat-0c7d28f22e45d0aa1 / awsproject-nat-gw". The "Details" section displays the NAT gateway ID (nat-0c7d28f22e45d0aa1), ARN (arn:aws:ec2:us-east-1:172615386837:natgateway/nat-0c7d28f22e45d0aa1), VPC (vpc-0b6fdbbb4a6bca32b), connectivity type (Public), primary public IPv4 address (52.5.45.18), subnet (subnet-070ef4aa9b1fad6fb / public-us-east-1a), state (Available), primary private IPv4 address (10.0.1.27), created date (Tuesday, December 31, 2024 at 01:11:33 GMT+1), and primary network interface ID (eni-07011d47f59f19378).

- Configure the **routing tables** to correctly connect public and private subnets.

Routing Table for the 2 Public Subnets:

The screenshot shows a CloudFormation stack named "rtb-0b3768ae35068e59e / awsproject-public-rt". The "Routes" tab is selected, showing two routes: one for destination 0.0.0.0/0 targeting the Internet Gateway (igw-0e9f3e70aba335c5f) with status Active, and another for destination 10.0.0.0/16 targeting the local subnet with status Active. The "Edit routes" button is visible in the top right corner.

Routing Table for the 2 Private Subnets Dedicated to the Frontend:

<input checked="" type="checkbox"/> awsproject-frontend-rt	rtb-027ec8b0d7a92a20d	2 subnets	-	No	vpc-0b6...	172615386837
<hr/>						
Details	Routes	Subnet associations	Edge associations	Route propagation	Tags	
<hr/>						
Routes (2)	<input type="button" value="Both"/>	<input type="button" value="Edit routes"/>				
<hr/>						
Destination	▼ Target	▼ Status	▼ Propagated			
0.0.0.0/0	nat-0c7d28f22e45d0aa1	Active	No			
10.0.0.0/16	local	Active	No			

Routing Table for the 2 Private Subnets Dedicated to the Backend:

<input checked="" type="checkbox"/> awsproject-backend-rt	rtb-02f5e0c7fa5760e62	2 subnets	-	No	vpc-0b6...	172615386837
<hr/>						
Details	Routes	Subnet associations	Edge associations	Route propagation	Tags	
<hr/>						
Routes (2)	<input type="button" value="Both"/>	<input type="button" value="Edit routes"/>				
<hr/>						
Destination	▼ Target	▼ Status	▼ Propagated			
0.0.0.0/0	nat-0c7d28f22e45d0aa1	Active	No			
10.0.0.0/16	local	Active	No			

Routing Table for the 2 Private Subnets Dedicated to the RDS Database:

<input checked="" type="checkbox"/> awsproject-rds-rt	rtb-0772bc5779b1d787e	2 subnets	-	No	vpc-0b6...	172615386837
<hr/>						
Details	Routes	Subnet associations	Edge associations	Route propagation	Tags	
<hr/>						
Routes (1)	<input type="button" value="Both"/>	<input type="button" value="Edit routes"/>				
<hr/>						
Destination	▼ Target	▼ Status	▼ Propagated			
10.0.0.0/16	local	Active	No			

Note: The database does not need Internet access, so no route to the NAT Gateway is configured for it.

3. DNS Management with Route 53:

- We configured a **hosted zone** in Route 53 to manage the domain names for your application.

Hosted zone details

Hosted zone name: projetccloud-fst.com

Hosted zone ID: Z00091348F15C51OPCIT

Description: -

Query log: -

Type: Public hosted zone

Record count: 4

Name servers:

- ns-685.awsdns-21.net
- ns-1649.awsdns-14.co.uk
- ns-28.awsdns-03.com
- ns-1491.awsdns-58.org

- We created an **A record** or **CNAME record** to associate our domain with the public IP address of the **load balancer**.

Record details

Record name: projetccloud-fst.com

Record type: A

Value: dualstack.frontend-alb-754519508.us-east-1.elb.amazonaws.com

Alias: Yes

TTL (seconds): -

Routing policy: Simple

- We added a subdomain, **api.projetccloud-fst.com**, pointing to the backend via the **load balancer**.

Record details

Record name: api.projetccloud-fst.com

Record type: A

Value: dualstack.internal-backend-alb-1582098394.us-east-1.elb.amazonaws.com

Alias: Yes

TTL (seconds): -

Routing policy: Simple

4. Application Deployment

Frontend :

- **Technology** : React.js

- **Deployment:** Hébergé sur des instances EC2 avec NGINX comme serveur web.
- **Port and protocol :**
 - **Incoming Port :** HTTP (port 80) for user traffic.
 - **Outgoing Traffic :** HTTP (port 80) to the backend's Application Load Balancer (ALB).

Backend :

- **Technology :** Node.js with Express.js
- **Deployment :** Hosted on EC2 instances managed by an Application Load Balancer (ALB).
- **Port and protocol :**
 - **Incoming Port :** HTTP (port 3200) via the ALB.
 - **Outgoing Traffic :** MySQL (port 3306) to the RDS database instance.

Database :

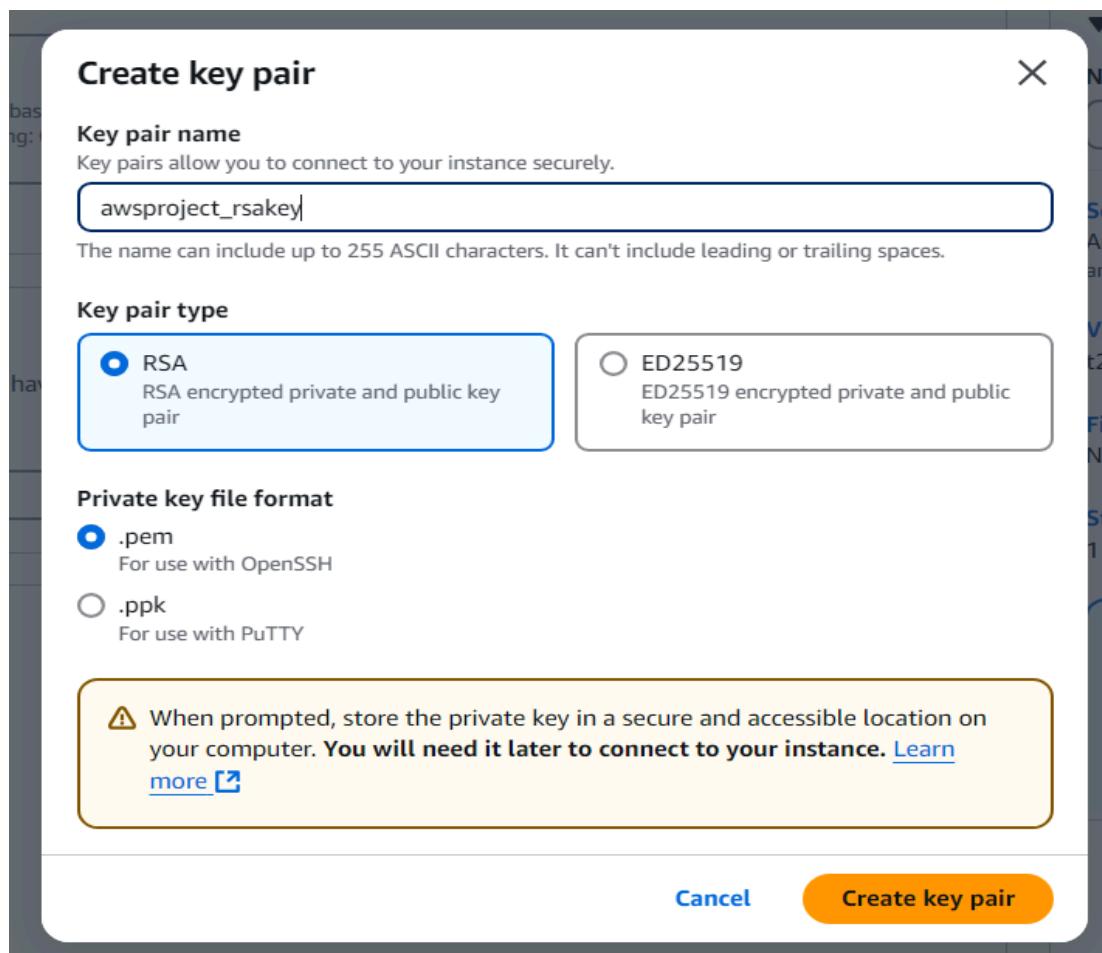
- **Technology :** MySQL on an RDS instance.
- **Port and protocole :**
 - **Incoming Port :** MySQL (port 3306) for requests from the backend.
 - **Outgoing Traffic :** Communication is only inbound from the backend.

Phase 2: Security

5. Bastion Host :

- We started by adding a **Bastion EC2 instance** in a public subnet to connect to machines in private subnets.

First, we created a **key pair** to connect to the bastion host via SSH:



Creation of a virtual machine in a public subnet

Instance summary for i-041b7170c415a7337 (Bastion Host) [Info](#)

Updated less than a minute ago

Instance ID i-041b7170c415a7337	Public IPv4 address 52.91.141.129 open address	Private IPv4 addresses 10.0.1.68
IPv6 address -	Instance state Running	Public IPv4 DNS -
Hostname type IP name: ip-10-0-1-68.ec2.internal	Private IP DNS name (IPv4 only) ip-10-0-1-68.ec2.internal	Elastic IP addresses -
Answer private resource DNS name -	Instance type t2.micro	AWS Compute Optimizer finding Opt-in to AWS Compute Optimizer for recommendations. Learn more
Auto-assigned IP address 52.91.141.129 [Public IP]	VPC ID vpc-0b6fdbbb4a6bca32b (awsproject-vpc)	Auto Scaling Group name -
IAM Role -	Subnet ID subnet-070ef4aa9b1fad6fb (public-us-east-1a)	Managed false
IMDSv2 Required	Instance ARN arn:aws:ec2:us-east-1:172615386837:instance/i-041b7170c415a7337	
Operator -		

- We restricted SSH access to the bastion host only from a specific IP address (our workstation) for security purposes.

We assigned this **security group** to the bastion host, specifying the **inbound** and **outbound rules** as follows:

sg-0e5bf86baff918a02 - bastion-host-sg

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

Inbound rules (1)

Name	Security group rule ID	IP version	Type	Protocol	Port range	Source	Description
-	sgr-07f65551d71f18597	IPv4	SSH	TCP	22	196.176.62.90/32	allow only ssh traffic from my IP

sg-0e5bf86baff918a02 - bastion-host-sg

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

Outbound rules (1)

Name	Security group rule ID	IP version	Type	Protocol	Port range	Destination	Description
-	sgr-0a04bac26a6d0253	IPv4	All traffic	All	All	0.0.0.0/0	-

We then connected to the bastion host via SSH:

```
PS C:\Users\yasm\Downloads\projet-cloud> ssh -i "awsproject_rsakey.pem" ec2-user@52.91.141.129
The authenticity of host '52.91.141.129 (52.91.141.129)' can't be established.
ED25519 key fingerprint is SHA256:UR0rg2aegElTjKDhoHxln0b5ijmdveF102s/2EgdYo.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '52.91.141.129' (ED25519) to the list of known hosts.

      _#_
     / \_###_          Amazon Linux 2023
    / \_####\_
   / \#\#
  / \#/ --> https://aws.amazon.com/linux/amazon-linux-2023
 / \V~' '-->
 / \_/
 / \_/
 / \_/
 [ec2-user@ip-10-0-1-68 ~]$
```

6. Security Groups:

- To secure traffic, we configured distinct **security groups** for each service group (frontend, backend, database) with the following rules:
- **Frontend** : Allows only **HTTP/HTTPS traffic** from the Internet via the load balancer.

Security Group for the Frontend Load Balancer:

The screenshot shows the AWS CloudFormation console for a stack named 'frontend-alb'. The 'Resources' tab is selected, displaying the 'sg-0eeb421b10a5d95d1 - frontend-alb-sg' resource. The 'Inbound rules' tab is active, showing two rules:

Name	Security group rule ID	IP version	Type	Protocol	Port range	Source
-	sgr-0a399bb874f27e370	IPv4	HTTPS	TCP	443	0.0.0.0/0
-	sgr-01981b878a1157d83	IPv4	HTTP	TCP	80	0.0.0.0/0

The 'Outbound rules' tab is also visible, showing one rule:

Name	Security group rule ID	IP version	Type	Protocol	Port range	Destination	Description
-	sgr-0215f84ab18d356d8	IPv4	All traffic	All	All	0.0.0.0/0	-

Security Group for the EC2 instances running the frontend of our application:

Inbound rules (2)

Name	Security group rule ID	IP version	Type	Protocol	Port range	Source	Description
-	sgr-0ae54b7e65439f1f5	-	HTTP	TCP	80	sg-0eeb421b10a5d95d...	allows traffic from front alb
-	sgr-03e95521a3b7dd6aa	-	SSH	TCP	22	sg-0e5bf86baff918a02...	allows ssh traffic from bastion

Outbound rules (1)

Name	Security group rule ID	IP version	Type	Protocol	Port range	Destination	Description
-	sgr-02d520640933664a6	IPv4	All traffic	All	All	0.0.0.0/0	-

- **Backend** : Allows only connections from the frontend (e.g., on port 8080).

Security Group for the Backend Load Balancer:

Inbound rules (1)

Name	Security group rule ID	IP version	Type	Protocol	Port range	Source	Description
-	sgr-088f0e219aa29a254	-	HTTP	TCP	80	sg-096b14371efe2c090...	allows http traffic from frontend ec2

Outbound rules (1)

Name	Security group rule ID	IP version	Type	Protocol	Port range	Destination	Description
-	sgr-0fae09410b6ecde19	IPv4	All traffic	All	All	0.0.0.0/0	-

Security Group for the EC2 instances running the backend of our application:

sg-0363c5f2f78579e78 - backend-ec2-sg

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

Inbound rules (2)

<input type="checkbox"/>	Name	Security group rule ID	IP version	Type	Protocol	Port range	Source	Description
<input type="checkbox"/>	-	sgr-0085bb2517ef6e327	-	SSH	TCP	22	sg-0e5bf86baff918a02...	allows ssh traffic from bastion
<input type="checkbox"/>	-	sgr-0c8b028dbbee211533	-	Custom TCP	TCP	3200	sg-0d3b0fd849975bf39...	allows traffic from backend alb

sg-0363c5f2f78579e78 - backend-ec2-sg

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

Outbound rules (1)

<input type="checkbox"/>	Name	Security group rule ID	IP version	Type	Protocol	Port range	Destination	Description
<input type="checkbox"/>	-	sgr-0863e2503c169c520	IPv4	All traffic	All	All	0.0.0.0/0	-

- **Database:** Allows only connections from the backend on port 3306.

sg-03b4740ddcd5cbef0 - rds-sg

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

Inbound rules (2)

<input type="checkbox"/>	Name	Security group rule ID	IP version	Type	Protocol	Port range	Source	Description
<input type="checkbox"/>	-	sgr-06b6865a60475523	-	MySQL/Aurora	TCP	3306	sg-0e5bf86baff918a02...	allows traffic from the bastion
<input type="checkbox"/>	-	sgr-0240485f2429e0ee5	-	MySQL/Aurora	TCP	3306	sg-0363c5f2f78579e78...	allows traffic from the backend ec2

sg-03b4740ddcd5cbef0 - rds-sg

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

Outbound rules (1)

<input type="checkbox"/>	Name	Security group rule ID	IP version	Type	Protocol	Port range	Destination	Description
<input type="checkbox"/>	-	sgr-0d233dfade517310c	IPv4	All traffic	All	All	0.0.0.0/0	-

Phase 3 : Instance Deployment

7. RDS Database:

- We configured an **RDS MySQL database** in private subnets.

Created a **subnet group**

The screenshot shows the 'Subnet group details' section with the following information:

- VPC ID: vpc-0b6fd8bb4a6bca52b
- ARN: arn:aws:rds:us-east-1:172615386837:subgrp:db-subnet-group
- Supported network types: IPv4
- Description: db-subnet-group

Below this is the 'Subnets (2)' section, which lists two subnets:

Availability zone	Subnet name	Subnet ID	CIDR block
us-east-1a	rds-us-east-1a	subnet-03cf2aa953af28ef9	10.0.4.0/24
us-east-1b	rds-us-east-1b	subnet-03af9649360fa80	10.0.8.0/24

Set up the database instance.

db-instance

The screenshot shows the 'Summary' section of the DB instance configuration:

DB identifier	Status	Role	Engine
db-instance	Available	Instance	MySQL Community
CPU	Class	Current activity	Region & AZ
3.46%	db.t4g.micro	0 Connections	us-east-1a

- Enabled **multi-AZ replication** for high availability:

Instance class

Instance class

db.t4g.micro

vCPU

2

RAM

1 GB

Availability

Master username

admin

Master password

IAM DB authentication

Not enabled

Multi-AZ

Yes

Secondary Zone

us-east-1b

- Configured automatic backups and manual snapshots.

Backup

Automated backups	Enabled (1 Day)	Latest restore time	December 31, 2024, 03:30 (UTC+01:00)	Replicate to Region	-
Copy tags to snapshots	Enabled	Backup window	04:08-04:38 UTC (GMT)	Replicated automated backup	-
Backup target	AWS Cloud (US East (N. Virginia))				

Snapshots (1)

Actions		Restore	Remove	Take snapshot
<input type="checkbox"/> Snapshot name		▲ Snapshot creation time	▼ Status	▼ Snapshot type
rds:db-instance-2024-12-31-02-29		December 31, 2024, 03:30 (UTC+01:00)	Available	Automated

Manual Snapshot:

Snapshots

Manual System Shared with me Public Backup service Exports in Amazon S3

Manual snapshots (1)						Actions	Take snapshot
<input type="checkbox"/> Filter by manual snapshots							
	Snapshot name	Engine version	DB instance or cluster	Snapshot creation time	DB Instance created time		
	my-db-snapshot	8.0.39	db-instance	December 31, 2024, 03:37 (UTC+01:00)	December 31, 2024, 03:29 (UTC+01:00)		

- To access RDS from the bastion host :

We installed MySQL on the bastion using the following commands:

To install the MySQL repository package:

```
➤ wget  
https://dev.mysql.com/get/mysql80-community-release-el9-1.noarch.  
rpm
```

To verify the installation:

```
➤ ls -lrt
```

To install the repository package:

```
➤ sudo dnf install -y mysql80-community-release-el9-1.noarch.rpm
```

To import the GPG key :

```
➤ sudo rpm --import  
https://repo.mysql.com/RPM-GPG-KEY-mysql-2023
```

To update the package index:

```
➤ sudo dnf update -y
```

To install the MySQL server:

```
➤ sudo dnf install -y mysql-community-server
```

To start the MySQL service:

```
➤ sudo systemctl start mysqld
```

To enable MySQL at startup :

```
➤ sudo systemctl enable mysqld
```

And there you go—MySQL is successfully installed on the bastion host :

```
[ec2-user@ip-10-0-1-68 ~]$ mysql --version
mysql Ver 8.0.40 for Linux on x86_64 (MySQL Community Server - GPL)
```

Next, we connected to RDS :

```
[ec2-user@ip-10-0-1-68 ~]$ mysql -h db-instance.c5dm2g8sjnly.us-east-1.rds.amazonaws.com -P 3306 -u admin -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 11
Server version: 8.0.39 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> |
```

Note:

We used the **master password** configured during the creation of the RDS instance.

- Database Preparation and Table Population: We created the database tables and populated them with sample data

Table Creation and Sample Data:

```

mysql> USE react_node_app;
    int NOT NULL AUTO_INCREMENT,
    `title` varchar(255) NOT NULL,
    `releaseDate` date NOT NULL,
    `description` text NOT NULL,
    `pages` int NOT NULL,
    `createdAt` date NOT NULL,
    `updatedAt` date NOT NULL,
    `authorId` int DEFAULT NULL,
    PRIMARY KEY (`id`),
    KEY `FK_66a4f0f47943a0d99c16ecf90b2` (`authorId`),
    CONSTRAINT `FK_66a4f0f47943a0d99c16ecf90b2` FOREIGN KEY (`authorId`) REFERERDatabase changed
mysql> -- Create Tables
mysql> CREATE TABLE `author` (
    -> `id` int NOT NULL AUTO_INCREMENT,
    -> `name` varchar(255) NOT NULL,
    -> `birthday` date NOT NULL,
    -> `bio` text NOT NULL,
    -> `createdAt` date NOT NULL,
    -> `updatedAt` date NOT NULL,
    -> PRIMARY KEY (`id`)
    -> ) ENGINE=InnoDB AUTO_INCREMENT=8 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
NCES `author` (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=10 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

```

```

mysql>
mysql> CREATE TABLE `book` (
    -> `id` int NOT NULL AUTO_INCREMENT,
    -> `title` varchar(255) NOT NULL,
    -> `releaseDate` date NOT NULL,
    -> `description` text NOT NULL,
    -> `pages` int NOT NULL,
    -> `createdAt` date NOT NULL,
    -> `updatedAt` date NOT NULL,
    -> `authorId` int DEFAULT NULL,
    -> PRIMARY KEY (`id`),
    -> KEY `FK_66a4f0f47943a0d99c16ecf90b2` (`authorId`),
    -> CONSTRAINT `FK_66a4f0f47943a0d99c16ecf90b2` FOREIGN KEY (`authorId`) REFERENCES `author` (`id`)
    -> ) ENGINE=InnoDB AUTO_INCREMENT=10 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
Query OK, 0 rows affected (0.04 sec)

```

Here is the display of the sample data we inserted :

```
mysql> select * from author;
+----+-----+-----+-----+-----+
| id | name | birthday | bio | createdAt | updatedAt |
+----+-----+-----+-----+-----+
| 1 | J.K. Rowling (Joanne Kathleen Rowling) | 1965-07-31 | J.K. Rowling is a British author best known for writing the Harry Potter fantasy series. The series has won multiple awards and sold over 500 million copies, becoming the best-selling book series in history. Rowling has also written other novels, including The Casual Vacancy and the Cormoran Strike crime series under the pen name Robert Galbraith. | 2024-05-29 | 2024-05-29 |
| 3 | Jane Austen | 1775-12-16 | Jane Austen was an English novelist known for her wit, social commentary, and romantic stories. Her six major novels, which explore themes of love, marriage, and money, have earned her a place as one of the greatest writers in the English language. | 2024-05-29 | 2024-05-29 |
| 4 | Harper Lee | 1926-07-11 | Harper Lee was an American novelist best known for her Pulitzer Prize-winning novel To Kill a Mockingbird. The novel explores themes of racial injustice and the importance of compassion. Lee published a sequel, Go Set a Watchman, in 2015. | 2024-05-29 | 2024-05-29 |
| 5 | J.R.R. Tolkien | 1954-07-29 | J.R.R. Tolkien was a British philologist and writer best known for his fantasy novels The Hobbit and The Lord of the Rings. Tolkien's works have had a profound influence on the fantasy genre and popular culture. | 2024-05-29 | 2024-05-29 |
| 6 | Mary Shelley | 1818-03-03 | Mary Shelley was a British novelist, playwright, and short story writer, the daughter of Mary Wollstonecraft Godwin and the wife of poet Percy Bysshe Shelley. Frankenstein, or, The Modern Prometheus (1818) is her most famous work. | 2024-05-29 | 2024-05-29 |
| 7 | Douglas Adams | 1979-10-12 | Douglas Adams was an English science fiction writer, satirist, humorist, dramatist, screenwriter, and occasional actor. He is best known for the Hitchhiker's Guide to the Galaxy comedy series, which inspired a radio comedy, several books, stage shows, comic books, a 1981 TV series, a 1984 video game, a 2005 feature film, and a 2008 sequel film. | 2024-05-29 | 2024-05-29 |
+----+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

```
mysql> select * from book;
+----+-----+-----+-----+-----+
| id | title | releasedate | description | pages | createdAt | updatedAt | authorId |
+----+-----+-----+-----+-----+
| 1 | Harry Potter and the Sorcerer's Stone | 1997-07-26 | On his birthday, Harry Potter discovers that he is the son of two well-known wizards, from whom he has inherited magical powers. He must attend a famous school of magic and sorcery, where he establishes a friendship with two young men who will become his companions on his adventure. During his first year at Hogwarts, he discovers that a malevolent and powerful wizard named Voldemort is in search of a philosopher's stone that prolongs the life of its owner. | 223 | 2024-05-29 | 2024-05-29 | 1 |
| 3 | Harry Potter and the Chamber of Secrets | 1998-07-02 | Harry Potter and the sophomore investigate a malevolent threat to their Hogwarts classmates, a menacing beast that hides within the castle. | 223 | 2024-05-29 | 2024-05-29 | 1 |
| 4 | Pride and Prejudice | 1813-01-28 | An English novel of manners by Jane Austen, first published in 1813. The story centres on the relationships among the Bennet sisters, in particular Elizabeth Bennet, the middle daughter, and the wealthy Mr. Darcy. It satirizes the social classes of the English gentry through a witty and ironic narrative voice. | 251 | 2024-05-29 | 2024-05-29 | 1 |
| 5 | Harry Potter and the Prisoner of Azkaban | 1999-07-08 | Harry's third year of studies at Hogwarts is threatened by Sirius Black's escape from Azkaban prison. Apparently, it is a dangerous wizard who was an accomplice of Lord Voldemort and who will try to take revenge on Harry Potter. | 234 | 2024-05-29 | 2024-05-29 | 1 |
| 6 | Harry Potter and the Goblet of Fire | 2000-07-08 | Hogwarts prepares for the Triwizard Tournament, in which three schools of wizardry will compete. To everyone's surprise, Harry Potter is chosen to participate in the competition, in which he must fight dragons, travel the water, and face his greatest fears. | 317 | 2024-05-29 | 2024-05-29 | 1 |
| 7 | The Hitchhiker's Guide to the Galaxy | 1979-10-12 | A comic science fiction comedy series created by Douglas Adams. The story follows the comedic misadventures of Arthur Dent, a hapless Englishman, following the destruction of the Earth by the Vogons, a race of unpleasant bureaucratic aliens. Arthur escapes with his friend Ford Prefect, who reveals himself to be an undercover researcher for the titular Hitchhiker's Guide to the Galaxy, a galactic encyclopedia containing information about anything and everything. | 184 | 2024-05-29 | 2024-05-29 | 7 |
| 8 | Victor Frankenstein; or, The Modern Prometheus | 1818-03-03 | A gothic novel by Mary Shelley that tells the story of Victor Frankenstein, a young scientist who creates a grotesque creature in an unorthodox scientific experiment. Frankenstein is horrified by his creation and abandons it, but the creature seeks revenge. The novel explores themes of scientific responsibility, creation versus destruction, and the nature of good and evil. | 636 | 2024-05-29 | 2024-05-29 | 1 |
| 9 | The Lord of the Rings: The Fellowship of the Ring | 1954-07-29 | The first book in J.R.R. Tolkien's epic fantasy trilogy, The Lord of the Rings. The Fellowship of the Ring follows hobbit Frodo Baggins as he inherits the One Ring, an evil artifact of power created by the Dark Lord Sauron. Frodo embarks on a quest to destroy the Ring in the fires of Mount Doom, accompanied by a fellowship of eight companions. | 482 | 2024-05-29 | 2024-05-29 | 5 |
+----+-----+-----+-----+-----+
8 rows in set (0.00 sec)
```

8. Load Balancer and Auto Scaling :

- We configured an **Application Load Balancer (ALB)** to manage traffic for the frontend and another one for the backend, based on requirements.

Backend Deployment: Launch Template

The screenshot shows the AWS Lambda console interface for a launch template named 'backend-launch-template'. The top navigation bar includes 'Actions' and 'Delete template' buttons. Below the title, there's a 'Launch template details' section with fields for 'Launch template ID' (lt-0f6f38aacb227d975), 'Launch template name' (backend-launch-template), 'Default version' (2), and 'Owner' (arn:aws:sts::172615386837:assumed-role/vocabs/user3562905-yasmine.benslimane@etudiant-fst.utm.tn). A 'Details' tab is selected, showing tabs for 'Details', 'Versions', and 'Template tags'. The 'Launch template version details' section displays the default version (2) with a date created of 2024-12-31T10:57:21.000Z. It also lists 'Instance details', 'Storage', 'Resource tags', 'Network interfaces', and 'Advanced details' tabs. Under 'Instance details', it shows the AMI ID (ami-01816d07b1128cd2d), Instance type (t2.micro), Availability Zone (-), and Key pair name (awsproject_rsakey). Security groups are listed as '-'.

Here is the user data script we configured for the EC2 instances running the backend :

```

#!/bin/bash
# Update package list and install required packages
sudo yum update -y
sudo yum install -y git

# Install Node.js (use NodeSource for the latest version)
curl -fsSL https://rpm.nodesource.com/setup_18.x | sudo bash -
sudo yum install -y nodejs

# Install PM2 globally
sudo npm install -g pm2

# Define variables
REPO_URL="https://github.com/learnItRightWay01/react-node-mysql-app.git"
BRANCH_NAME="feature/add-logging"
REPO_DIR="/home/ec2-user/react-node-mysql-app/backend"
ENV_FILE="$REPO_DIR/.env"

# Clone the repository
cd /home/ec2-user
sudo -u ec2-user git clone $REPO_URL
cd react-node-mysql-app

# Checkout to the specific branch
sudo -u ec2-user git checkout $BRANCH_NAME
cd backend

# Define the log directory and ensure it exists
LOG_DIR="/home/ec2-user/react-node-mysql-app/backend/logs"
mkdir -p $LOG_DIR
sudo chown -R ec2-user:ec2-user $LOG_DIR

# Append environment variables to the .env file
echo "LOG_DIR=$LOG_DIR" >> "$ENV_FILE"
echo "DB_HOST=\"db-instance.c5dm2g8sjnly.us-east-1.rds.amazonaws.com\"" >> "$ENV_FILE"
echo "DB_PORT=3306" >> "$ENV_FILE"
echo "DB_USER=admin" >> "$ENV_FILE"
echo "DB_PASSWORD=aws_fst2025" >> "$ENV_FILE"
echo "DB_NAME=react_node_app" >> "$ENV_FILE"

# Install Node.js dependencies as ec2-user
sudo -u ec2-user npm install

# Start the application using PM2 as ec2-user
sudo -u ec2-user npm run serve

```

Target Group for the Backend :

backend-target-grp

Actions ▾

Details

arn:aws:elasticloadbalancing:us-east-1:172615386837:targetgroup/backend-target-grp/8a8e20dfa5c1b545

Target type Instance	Protocol : Port HTTP: 3200	Protocol version HTTP1	VPC vpc-0b6fdbbb4a6bca32b					
IP address type IPv4	Load balancer backend-alb							
2 Total targets	<table border="1"> <tr> <td>2 Healthy</td> <td>0 Unhealthy</td> </tr> <tr> <td>0 Anomalous</td> <td></td> </tr> </table>	2 Healthy	0 Unhealthy	0 Anomalous		0 Unused	0 Initial	0 Draining
2 Healthy	0 Unhealthy							
0 Anomalous								

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

Zone	Total targets	Healthy	Unhealthy	Unused	Initial
us-east-1a (use1-az4)	1	1	0	0	0
us-east-1b (use1-az6)	1	1	0	0	0

Last fetched seconds ago

Targets | Monitoring | **Health checks** | Attributes | Tags

Health check settings

Edit

Protocol HTTP	Path /health	Port Traffic port	Healthy threshold 5 consecutive health check successes
Unhealthy threshold 2 consecutive health check failures	Timeout 5 seconds	Interval 30 seconds	Success codes 200

We also set up the backend load balancer :

backend-alb

Actions ▾

▼ Details

Load balancer type Application	Status Active	VPC vpc-0b6fdbbb4a6bca32b	Load balancer IP address type IPv4
Scheme Internal	Hosted zone Z35SXDOTRQ7XK	Availability Zones subnet-044aa7bf6adc48e96 us-east-1a (use1-az4) subnet-031d5fb02ac4a5691 us-east-1b (use1-az6)	Date created December 31, 2024, 11:33 (UTC+01:00)
Load balancer ARN arn:aws:elasticloadbalancing:us-east-1:172615386837:loadbalancer/app/backend-alb/41329f751f845b0a	DNS name Info internal-backend-alb-1582098394.us-east-1.elb.amazonaws.com (A Record)		

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

Listeners and rules (1) [Info](#)

[C](#)

Manage rules ▾

Manage listener ▾

Add listener

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.

Filter listeners		Protocol:Port	Default action	Rules	ARN	Security policy	Default SSL/TLS certificate
<input type="checkbox"/>	HTTP:80		Forward to target group <ul style="list-style-type: none"> backend-target-grp: 1 (100%) Target group stickiness: Off 	1 rule	ARN	Not applicable	Not applicable

Frontend Deployment:

Launch template

frontend-launch-template (lt-050f45c5863ce6403)

Actions Delete template

Launch template details

Launch template ID lt-050f45c5863ce6403	Launch template name frontend-launch-template	Default version 1	Owner arn:aws:sts::172615386837:assumed-role/voclabs/user3562905:yasmine.benslimane@etudiant-fst.utm.tn
--	--	----------------------	--

Details Versions Template tags

Launch template version details

Version 1 (Default)	Description frontend-launch-template	Date created 2024-12-31T11:20:56.000Z	Created by arn:aws:sts::172615386837:assumed-role/voclabs/user3562905:yasmine.benslimane@etudiant-fst.utm.tn
Instance details Storage Resource tags Network interfaces Advanced details			
AMI ID ami-01816d07b1128cd2d	Instance type t2.micro	Availability Zone -	Key pair name awsproject_rsakey
Security groups -	Security group IDs sg-096b14371efe2c090		

User data script for frontend template launch :

In the variable **APP_TIER_ALB_URL**, we configured the DNS name of the backend ALB created earlier to ensure communication between the frontend and the deployed backend.

```

#!/bin/bash
# Update package list and install required packages
sudo yum update -y
sudo yum install -y git

# Install Node.js (use NodeSource for the latest version)
curl -fsSL https://rpm.nodesource.com/setup_18.x | sudo bash -
sudo yum install -y nodejs

# Install NGINX
sudo yum install -y nginx

# Start and enable NGINX
sudo systemctl start nginx
sudo systemctl enable nginx

# Define variables
REPO_URL="https://github.com/learnItRightWay01/react-node-mysql-app.git"
BRANCH_NAME="feature/add-logging"
REPO_DIR="/home/ec2-user/react-node-mysql-app/frontend"
ENV_FILE="$REPO_DIR/.env"
APP_TIER_ALB_URL="http://internal-backend-alb-1582098394.us-east-1.elb.amazonaws.com"
API_URL="/api"

# Clone the repository as ec2-user
cd /home/ec2-user
sudo -u ec2-user git clone $REPO_URL
cd react-node-mysql-app

# Checkout to the specific branch
sudo -u ec2-user git checkout $BRANCH_NAME
cd frontend

```

```
# Ensure ec2-user owns the directory
sudo chown -R ec2-user:ec2-user /home/ec2-user/react-node-mysql-app

# Create .env file with the API_URL
echo "VITE_API_URL=\"$API_URL\"" >> "$ENV_FILE"

# Install Node.js dependencies as ec2-user
sudo -u ec2-user npm install

# Build the frontend application as ec2-user
sudo -u ec2-user npm run build

# Copy the build files to the NGINX directory
sudo cp -r dist /usr/share/nginx/html/

# Update NGINX configuration
NGINX_CONF="/etc/nginx/nginx.conf"
SERVER_NAME="_" # Replace with your actual domain name

# Backup existing NGINX configuration
sudo cp $NGINX_CONF ${NGINX_CONF}.bak

# Write new NGINX configuration
sudo tee $NGINX_CONF > /dev/null <<EOL
user nginx;
worker_processes auto;

error_log /var/log/nginx/error.log warn;
pid /run/nginx.pid;

events {
    worker_connections 1024;
}
```

```

http {
    include /etc/nginx/mime.types;
    default_type application/octet-stream;

    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;
    types_hash_max_size 2048;

    include /etc/nginx/conf.d/*.conf;
}

EOL

# Create a separate NGINX configuration file
sudo tee /etc/nginx/conf.d/presentation-tier.conf > /dev/null <<EOL
server {
    listen 80;
    server_name $SERVER_NAME;
    root /usr/share/nginx/html/dist;
    index index.html index.htm;

    #health check
    location /health {
        default_type text/html;
        return 200 "<!DOCTYPE html><p>Health check endpoint</p>\n";
    }

    location / {
        try_files $uri /index.html;
    }
}

```

```

location /api/ {
    proxy_pass $APP_TIER_ALB_URL;
    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;
}
}

EOL

# Restart NGINX to apply the new configuration
sudo systemctl restart nginx

```

Target Group for the Frontend :

frontend-target-grp Actions ▾

Details

arn:aws:elasticloadbalancing:us-east-1:172615386837:targetgroup/frontend-target-grp/5a8d4b72585c8f7b

Target type Instance	Protocol : Port HTTP: 80	Protocol version HTTP1	VPC vpc-0b6fdbbb4a6bca32b
IP address type IPv4	Load balancer frontend-alb		
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.

Last fetched seconds ago

Zone	Total targets	Healthy	Unhealthy	Unused	Initial
us-east-1a (use1-az4)	1	1	0	0	0
us-east-1b (use1-az6)	1	1	0	0	0

[Targets](#)
[Monitoring](#)
[Health checks](#)
[Attributes](#)
[Tags](#)

Health check settings

Protocol HTTP	Path /health	Port Traffic port	Healthy threshold 5 consecutive health check successes
Unhealthy threshold 2 consecutive health check failures	Timeout 5 seconds	Interval 30 seconds	Success codes 200

We also set up the frontend load balancer :

frontend-alb

Details

Load balancer type Application	Status Active	VPC vpc-0b6fdbbb4a6bca32b	Load balancer IP address type IPv4
Scheme Internet-facing	Hosted zone Z35SXDOTRQ7XK	Availability Zones subnet-027e18a2dda396ea4 us-east-1b (use1-az6) subnet-070ef4aa9b1fad6fb us-east-1a (use1-az4)	Date created December 31, 2024, 12:42 (UTC+01:00)
Load balancer ARN arn:aws:elasticloadbalancing:us-east-1:172615386837:loadbalancer/app/frontend-alb/6b225e950f9c6a39	DNS name Info frontend-alb-754519508.us-east-1.elb.amazonaws.com (A Record)		

Listeners and rules [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.

<input type="checkbox"/> Protocol:Port	<input type="checkbox"/> Default action	<input type="checkbox"/> Rules	<input type="checkbox"/> ARN	<input type="checkbox"/> Security policy	<input type="checkbox"/> Default SSL/TLS certificate
<input type="checkbox"/> HTTP:80	Forward to target group • frontend-target-grp : 1 (100%) • Target group stickiness: Off	1 rule	ARN	Not applicable	Not applicable

Note :

The frontend load balancer was configured as internet-facing and placed in the public subnets of Availability Zones (AZ) A and B to ensure it is accessible externally

- We proceeded to set up **Auto Scaling Groups** for the web servers and application servers.

Backend Auto Scaling Group

We configured the backend auto scaling group as follows:

backend-autoscaling-group Capacity overview

arn:aws:autoscaling:us-east-1:172615386837:autoScalingGroup:b19864e3-f43d-4d19-83f3-000d518d7d51:autoScalingGroupName/backend-autoscaling-group

Desired capacity 2	Scaling limits (Min - Max) 1 - 3	Desired capacity type Units (number of instances)	Status -
-----------------------	-------------------------------------	--	-------------

Date created
Tue Dec 31 2024 11:47:15 GMT+0100 (West Africa Standard Time)

Edit

Details | Integrations - new | Automatic scaling | **Instance management** | Instance refresh | Activity | Monitoring

Instances (2)

Instance ID	Lifecycle	Instance ...	Weighted...	Launch t...	Availabil...	Health st...	Protecte...
i-014be5047dee882ef	InService	t2.micro	-	backend-launch-t	us-east-1b	Healthy	
i-07b66f54aa67f2463	InService	t2.micro	-	backend-launch-t	us-east-1a	Healthy	

Actions ▾

The desired number of instances was created by the auto scaling group.

We configured the auto-scaling group to create the instances on the backend subnets of the **us-east-1a** and **us-east-1b** availability zones.

Network

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

Subnet ID
subnet-044aa7bf6adc48e96, subnet-031d5fb02ac4a5691

Availability Zone distribution
Balanced best effort

Edit

We configured the auto-scaling group with the backend launch template created previously:

Launch template			
Launch template lt-0f6f38aacb227d975 backend-launch-template	AMI ID ami-01816d07b1128cd2d	Instance type t2.micro	Owner arn:aws:sts::172615386837:assumed-role/voclabs/user3562905=yasmine.benslimane@etudiant-fst.utm.tn
Version Default	Security groups -	Security group IDs sg-0363c5f2f78579e78	Create time Tue Dec 31 2024 11:37:21 GMT+0100 (West Africa Standard Time)
Description -	Storage (volumes) -	Key pair name awsproject_rsakey	Request Spot Instances No
View details in the launch template console			

We connected the auto-scaling group to the backend load balancer:

Load balancing		
Load balancer target groups backend-target-grp	Classic Load Balancers -	Edit

And we configured this **dynamic scaling policy**:

Dynamic scaling policies (1) [Info](#)

[Actions ▾](#) [Create dynamic scaling policy](#)

< 1 >

Target Tracking Policy	
Policy type	Target tracking scaling
Enabled or disabled	Enabled
Execute policy when	As required to maintain Average CPU utilization at 50
Take the action	Add or remove capacity units as required
Instances need	300 seconds to warm up before including in metric
Scale in	Enabled

We enabled the **ELB health checks**:

Health checks

[Edit](#)

Health check type	Health check grace period
EC2, ELB	300

And we also enabled monitoring with CloudWatch for this auto-scaling group:

backend-autoscaling-group

backend-autoscaling-group Capacity overview

[Edit](#)

Desired capacity	Scaling limits (Min - Max)	Desired capacity type	Status
1	1 - 5	Units (number of instances)	-

Date created
Tue Dec 31 2024 11:47:15 GMT+0100 (West Africa Standard Time)

[Details](#) [Integrations - new](#) [Automatic scaling](#) [Instance management](#) [Instance refresh](#) [Activity](#) [Monitoring](#)

CloudWatch monitoring details

[Auto Scaling](#) [EC2](#)

Auto Scaling group metrics collection:
 Enable
All times shown are in UTC.
[View all CloudWatch metrics](#)

[Alarm recommendations](#)

1h 3h 12h 1d 5d 1w Custom Local timezone [C](#) [Add to dashboard](#)

Minimum Group Size (Count)	Maximum Group Size (Count)	Desired Capacity (Count)	In Service Instances (Count)
2	3	2	2
1	2	1.5	1
0	1	1	0
09:12	12:11	12:11	12:11

We can visualize the different metrics throughout the deployment.

Frontend Auto-scaling Group

We configured the frontend auto-scaling group as follows:

frontend-autoscaling-group

frontend-autoscaling-group Capacity overview

arn:aws:autoscaling:us-east-1:172615386837:autoScalingGroup:41c74199-0e61-49ce-a75d-5c44e6b10c76:autoScalingGroupName/frontend-autoscaling-group

Desired capacity	Scaling limits (Min - Max)	Desired capacity type	Status
2	1 - 3	Units (number of instances)	-

Date created
Tue Dec 31 2024 12:51:20 GMT+0100 (West Africa Standard Time)

Details Integrations - new Automatic scaling Instance management Instance refresh Activity Monitoring

Instances (2)

Instance ID	Lifecycle	Instance Type	Weighted...	Launch t...	Availability...	Health st...	Protecte...
i-03f04c3daf9b21dc6	InService	t2.micro	-	frontend-launch-t	us-east-1a	Healthy	
i-0b25c34867f610150	InService	t2.micro	-	frontend-launch-t	us-east-1b	Healthy	

And as we can see, the desired number of instances was created by the auto-scaling group.

We configured the auto-scaling group to create the instances on the frontend subnets of the **us-east-1a** and **us-east-1b** availability zones:

Network

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

Subnet ID
subnet-05346dc767b4556f6, subnet-0e6263290ff8ae235

Availability Zone distribution
Balanced best effort

On configuré le asg avec le launch template du frontend créé précédemment :

Launch template			
Launch template lt-050f45c5863ce6403 frontend-launch-template	AMI ID ami-01816d07b1128cd2d	Instance type t2.micro	Owner arn:aws:sts::172615386837:assumed-role/vocabs/user3562905=yasmine.benslimane@etudiant-fst.utm.tn
Version Default	Security groups -	Security group IDs sg-096b14371efe2c090	Create time Tue Dec 31 2024 12:20:56 GMT+0100 (West Africa Standard Time)
Description frontend-launch-template	Storage (volumes) -	Key pair name awsproject_rsakey	Request Spot Instances No
View details in the launch template console			

We connected the auto-scaling group to the frontend load balancer:

Load balancing	
Load balancer target groups frontend-target-grp	Classic Load Balancers -

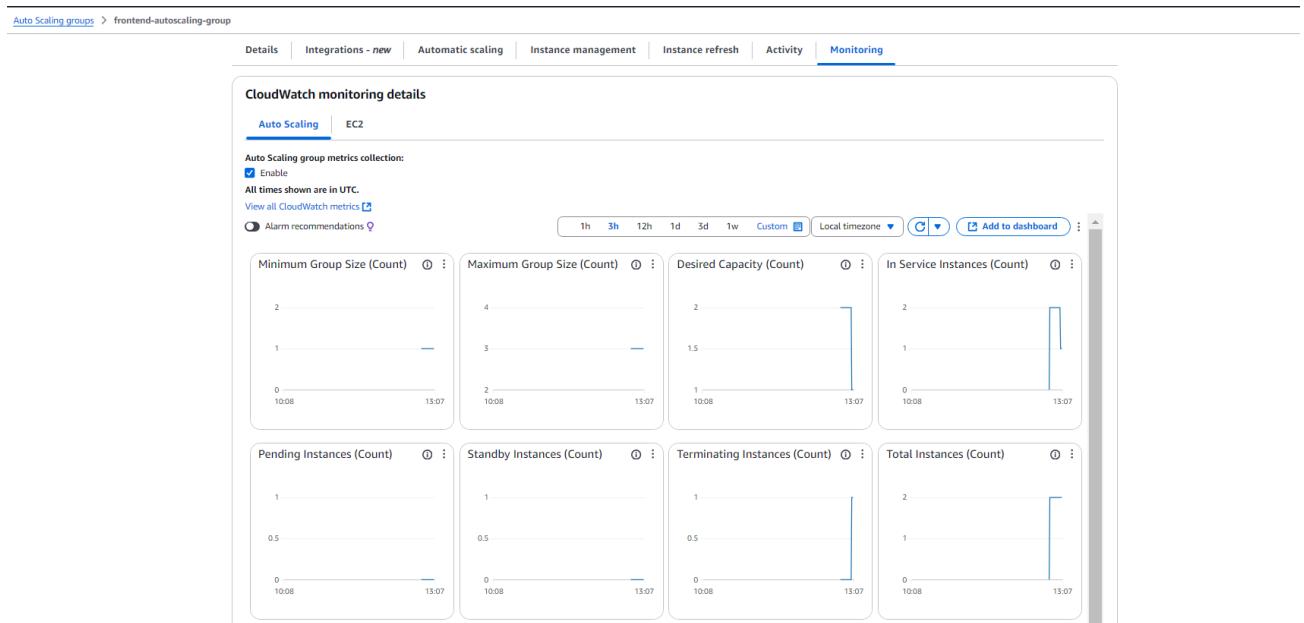
And we configured this dynamic scaling policy:

Dynamic scaling policies (1) Info	
Target Tracking Policy <p>Policy type Target tracking scaling</p> <p>Enabled or disabled Enabled</p> <p>Execute policy when As required to maintain Average CPU utilization at 50</p> <p>Take the action Add or remove capacity units as required</p> <p>Instances need 300 seconds to warm up before including in metric</p> <p>Scale in Enabled</p>	Actions ▾ Create dynamic scaling policy ◀ 1 ▶

We enabled the ELB health checks:

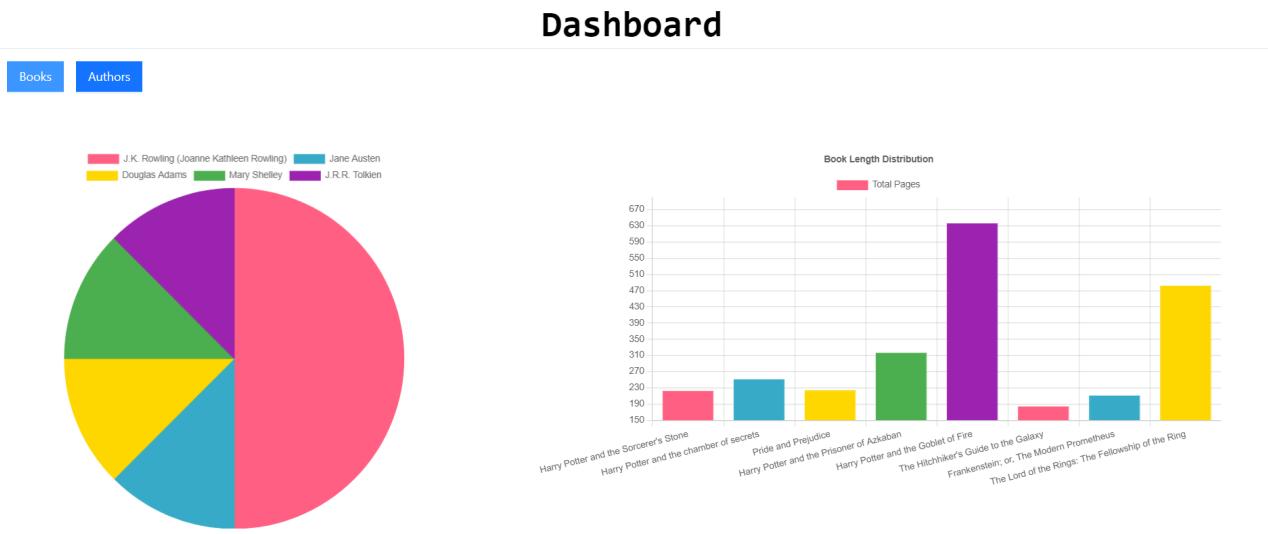
The screenshot shows the 'Health checks' configuration for an auto-scaling group. It includes fields for 'Health check type' (set to EC2, EBS) and 'Health check grace period' (set to 300). A blue 'Edit' button is located in the top right corner.

And we also enabled monitoring with CloudWatch for this auto-scaling group:



9. Testing the Deployed Application:

Here is our deployed application:



Here is the author module on which we can perform CRUD operations:

[Dashboard](#)

MANAGE AUTHORS

[+ Add Author](#)

ID	Author	Birthday	Description	Created Date	Updated Date	Actions
1	J.K. Rowling (Joanne Kathleen Rowling)	1965-07-31T00:00:00.000Z	J.K. Rowling is a British author best known for writing the Harry Potter fantasy series. The series has won multiple awards and sold over 500 million copies, becoming the best-selling book series in history. Rowling has also written other novels, including The Casual Vacancy and the Cormoran Strike crime series under the pen name Robert Galbraith.	2024-05-29T00:00:00.000Z	2024-05-29T00:00:00.000Z	
3	Jane Austen	1775-12-16T00:00:00.000Z	Jane Austen was an English novelist known for her wit, social commentary, and romantic stories. Her six major novels, which explore themes of love, marriage, and money, have earned her a place as one of the greatest writers in the English language.	2024-05-29T00:00:00.000Z	2024-05-29T00:00:00.000Z	
4	Harper Lee	1960-07-11T00:00:00.000Z	Harper Lee was an American novelist best known for her Pulitzer Prize-winning novel To Kill a Mockingbird. The novel explores themes of racial injustice and the importance of compassion. Lee published a sequel, Go Set a Watchman, in 2015.	2024-05-29T00:00:00.000Z	2024-05-29T00:00:00.000Z	
5	J.R.R. Tolkien	1954-07-29T00:00:00.000Z	J.R.R. Tolkien was a British philologist and writer best known for his fantasy novels The Hobbit and The Lord of the Rings. Tolkien's works have had a profound influence on the fantasy genre and popular culture.	2024-05-29T00:00:00.000Z	2024-05-29T00:00:00.000Z	
6	Mary Shelley	1818-03-03T00:00:00.000Z	Mary Shelley was a British novelist, playwright, and short story writer; the daughter of Mary Wollstonecraft Godwin and the wife of poet Percy Bysshe Shelley. Frankenstein, or, The Modern Prometheus (1818) is her most famous work.	2024-05-29T00:00:00.000Z	2024-05-29T00:00:00.000Z	
7	Douglas Adams	1979-10-12T00:00:00.000Z	Douglas Adams was an English science fiction writer, satirist, humorist, dramatist, screenwriter, and occasional actor. He is best known for the Hitchhiker's Guide to the Galaxy comedy series, which inspired a radio comedy, several books, stage shows, comic books, a 1981 TV series, a 1984 video game, a 2005 feature film, and a 2008 sequel film.	2024-05-29T00:00:00.000Z	2024-05-29T00:00:00.000Z	

< 1 >

Here is the book module on which we can perform CRUD operations:

[Dashboard](#)

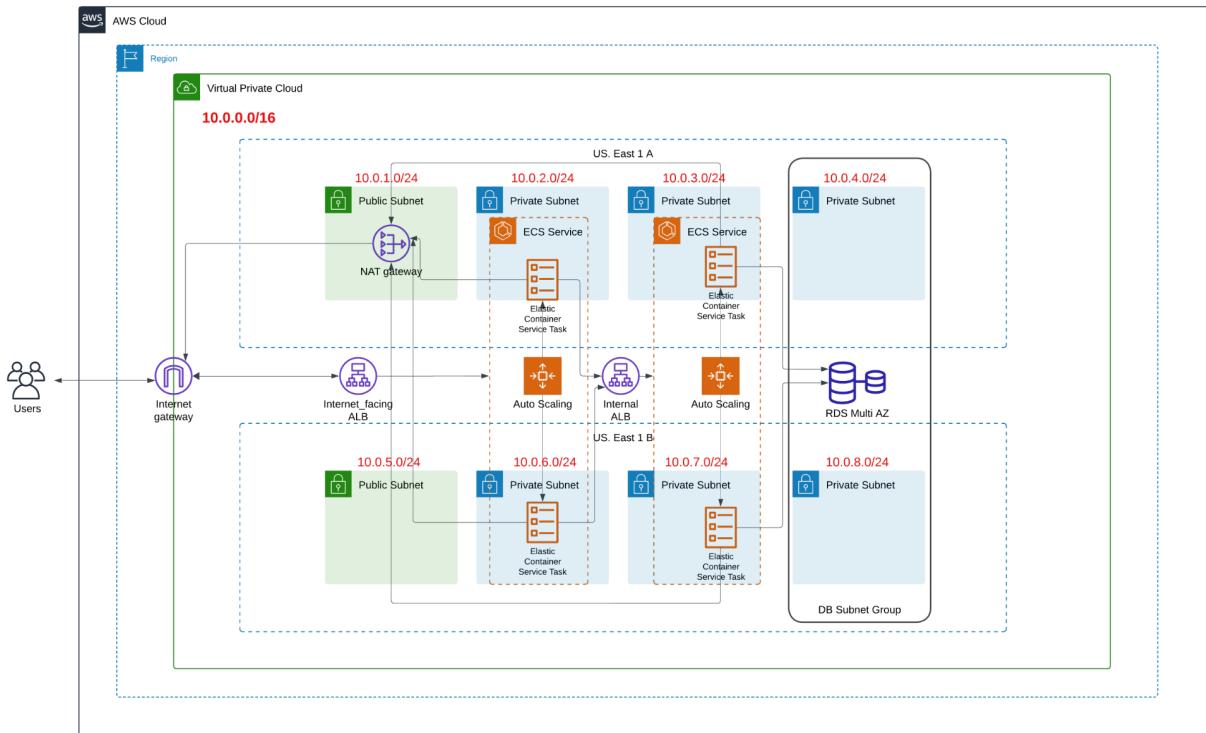
MANAGE BOOKS

[+ Add Book](#)

ID	Title	Description	Release Date	Author	Created Date	Updated Date	Actions
1	Harry Potter and the Sorcerer's Stone	On his birthday, Harry Potter discovers that he is the son of two well-known wizards, from whom he has inherited magical powers. He must attend a famous school of magic and sorcery, where he establishes a friendship with two young men who will become his companions on his adventure. During his first year at Hogwarts, he discovers that a malevolent and powerful wizard named Voldemort is in search of a philosopher's stone that prolongs the life of its owner.	1997-07-26T00:00:00.000Z	J.K. Rowling (Joanne Kathleen Rowling)	2024-05-29T00:00:00.000Z	2024-05-29T00:00:00.000Z	
3	Harry Potter and the chamber of secrets	Harry Potter and the sophomores investigate a malevolent threat to their Hogwarts classmates, a menacing beast that hides within the castle.	1998-07-02T00:00:00.000Z	J.K. Rowling (Joanne Kathleen Rowling)	2024-05-29T00:00:00.000Z	2024-05-29T00:00:00.000Z	
4	Pride and Prejudice	An English novel of manners by Jane Austen, first published in 1813. The story centres on the relationships among the Bennet sisters, in particular Elizabeth Bennet the middle daughter, and the wealthy Mr. Darcy. Austen satirizes the social classes of the English gentry through a witty and ironic narrative voice.	1813-01-28T00:00:00.000Z	Jane Austen	2024-05-29T00:00:00.000Z	2024-05-29T00:00:00.000Z	
5	Harry Potter and the Prisoner of Azkaban	Harry's third year of studies at Hogwarts is threatened by Sirius Black's escape from Azkaban prison. Apparently, it is a dangerous wizard who was an accomplice of Lord Voldemort and who will try to take revenge on Harry Potter.	1999-07-08T00:00:00.000Z	J.K. Rowling (Joanne Kathleen Rowling)	2024-05-29T00:00:00.000Z	2024-05-29T00:00:00.000Z	
6	Harry Potter and the Goblet of Fire	Hogwarts prepares for the Triwizard Tournament, in which three schools of wizardry will compete. To everyone's surprise, Harry Potter is chosen to participate in the competition, in which he must fight dragons, enter the water and face his greatest fears.	2000-07-08T00:00:00.000Z	J.K. Rowling (Joanne Kathleen Rowling)	2024-05-29T00:00:00.000Z	2024-05-29T00:00:00.000Z	

Phase 4 : Refactoring with Containers

10. Migration to ECS :



- We replicated the EC2 architecture using containers:

First, we began by configuring the Cluster, Task Definitions, and then the services for ECS.

Next, we configured the scaling of the containers based on CloudWatch metrics.

Cluster creation :

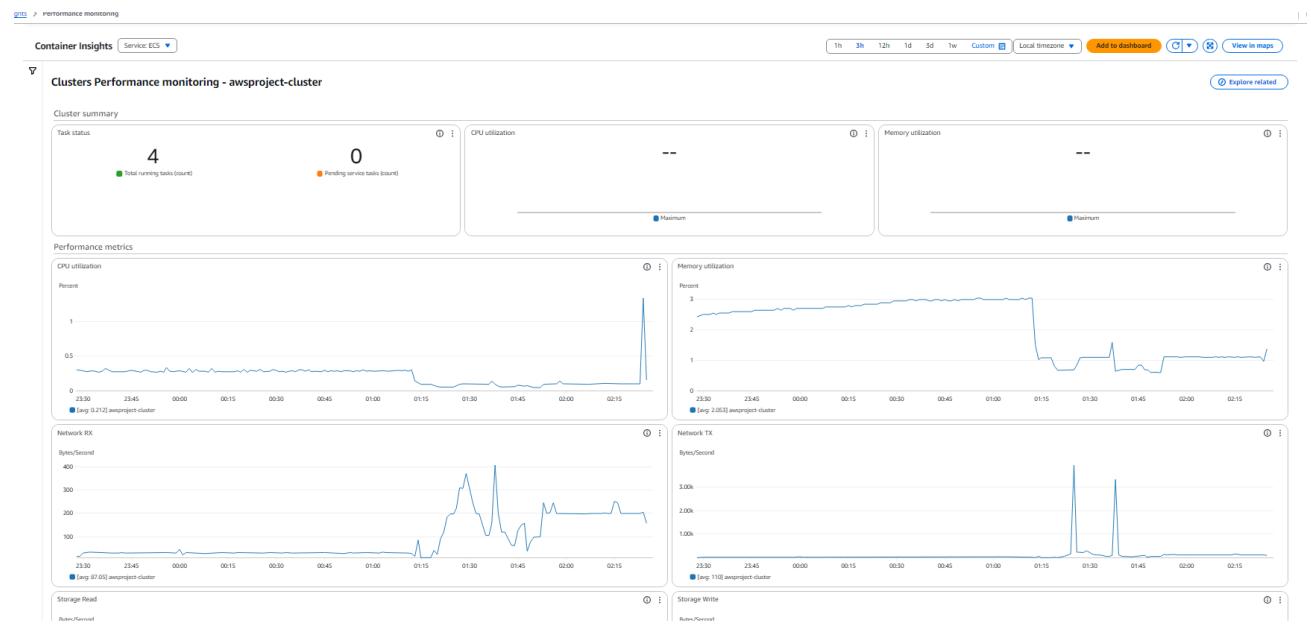
awsproject-cluster

Last updated December 31, 2024 at 19:57 (UTC+100) [Update cluster](#) [Delete cluster](#)

Cluster overview

ARN arn:aws:ecs:us-east-1:172615386837:cluster/awsproj-ect-cluster	Status Active	CloudWatch monitoring Container Insights View in CloudWatch	Registered container instances -
Services		Tasks	
Draining -	Active	Pending	Running
Encryption			
Managed storage -	Fargate ephemeral storage		

And we enabled the "Container Insights" option for cluster monitoring with CloudWatch.



For the backend deployment:

1- Dockerization of the Backend:

Before creating the backend task definition, we must first convert our application into a Docker image, which we will upload to Docker Hub.

To do this:

We navigated to the project's **backend** folder and added this **Dockerfile** (containing the endpoint of our RDS database):

```

backend > 📄 Dockerfile
 1  # Use Node.js 18 as the base image
 2  FROM node:18
 3
 4  # Set the working directory
 5  WORKDIR /usr/src/app
 6
 7  # Install PM2 globally
 8  RUN npm install -g pm2
 9
10 # Copy package.json and install dependencies
11 COPY package.json package-lock.json ./
12 RUN npm install
13
14 # Copy application files
15 COPY . .
16
17 # Set environment variables (can also be provided during container runtime)
18 ENV LOG_DIR=/usr/src/app/logs
19 ENV DB_HOST=db-instance.c5dm2g8sjnly.us-east-1.rds.amazonaws.com
20 ENV DB_PORT=3306
21 ENV DB_USER=admin
22 ENV DB_PASSWORD=aws_fst2025
23 ENV DB_NAME=react_node_app
24
25 # Ensure log directory exists
26 RUN mkdir -p $LOG_DIR && chown -R node:node $LOG_DIR
27
28 # Expose application port
29 EXPOSE 3200
30
31 # Start the application with PM2
32 CMD ["pm2-runtime", "start", "server.js", "--interpreter", "node"]

```

- And added this `.dockerignore` file:

```

backend > 📄 .dockerignore
 1  node_modules
 2  npm-debug.log
 3  .env
 4

```

- Then, we started the Docker engine and executed the following command:

```
docker build -t yasminebenslimane123/backend-awsproject
```

```
PS C:\Users\yasm\Downloads\AWS_project\lirw-react-node-mysql-app\backend> docker build -t yasminebenslimane123/backend-awsproject .
[+] Building 4.0s (13/13) FINISHED                                            docker:desktop-linux
=> [internal] load build definition from Dockerfile                         0.0s
=> => transferring dockerfile: 842B                                         0.0s
=> [internal] load metadata for docker.io/library/node:18                   2.3s
=> [auth] library/node:pull token for registry-1.docker.io                  0.0s
=> [internal] load .dockerrcignore                                         0.0s
=> => transferring context: 75B                                           0.0s
=> [internal] load build context                                         0.1s
=> => transferring context: 63.10kB                                       0.1s
=> [1/7] FROM docker.io/library/node:18@sha256:7f31a1eb14c61719b8bb0eaa029310cc33851f71d3578cc422b390f8096977c5 0.0s
=> CACHED [2/7] WORKDIR /usr/src/app                                      0.0s
=> CACHED [3/7] RUN npm install -g pm2                                     0.0s
=> CACHED [4/7] COPY package.json package-lock.json ./                     0.0s
=> CACHED [5/7] RUN npm install                                         0.0s
=> [6/7] COPY . .                                                       0.1s
=> [7/7] RUN mkdir -p /usr/src/app/logs && chown -R node:node /usr/src/app/logs 1.1s
=> exporting to image                                                     0.1s
=> => exporting layers                                                    0.1s
=> => writing image sha256:b23308f72ea0b8c45996d0ae637849aa8d3f52f6eee405934b870ea22aecf5ae 0.0s
=> => naming to docker.io/yasminebenslimane123/backend-awsproject          0.0s

What's next:
  View a summary of image vulnerabilities and recommendations → docker scout quickview
PS C:\Users\yasm\Downloads\AWS_project\lirw-react-node-mysql-app\backend> |
```

- We connected to Docker Hub :

```
PS C:\Users\yasma\Downloads\AWS_project\lirw-react-node-mysql-app\backend> docker login -u yasminebenslimane123  
Password:  
Login Succeeded
```

- We pushed the image to Docker Hub :

```
PS C:\Users\yasma\Downloads\AWS_project\lirw-react-node-mysql-app\backend> docker push yasminebenslimane123/backend
project
Using default tag: latest
The push refers to repository [docker.io/yasminebenslimane123/backend-awsproject]
c2078536b23a: Pushed
93abaf3a3fe3: Pushed
cd5b1bf05515: Pushed
9c0121b41943: Pushed
a0721d7e5b8f: Pushed
1d652418bc69: Pushed
40689c24a66a: Mounted from yasminebenslimane123/backend-app
fef83a3a8b40: Mounted from yasminebenslimane123/backend-app
57edb89ea088: Mounted from yasminebenslimane123/backend-app
6edb22a949de: Mounted from yasminebenslimane123/backend-app
6d58389117c3: Mounted from yasminebenslimane123/backend-app
85c6f0cfb532: Mounted from yasminebenslimane123/backend-app
a4fd1e7df47e: Mounted from yasminebenslimane123/backend-app
85f51e104e85: Mounted from yasminebenslimane123/backend-app
```



The screenshot shows a Docker Hub repository page. At the top, there's a navigation bar with links for 'Explore', 'Repositories' (which is underlined), 'Organizations', and 'Usage'. To the right of the navigation is a search bar with the placeholder 'Search Docker Hub' and a 'ctrl+K' keyboard shortcut. Further right are icons for account, notifications, settings, and a three-dot menu, followed by a yellow 'Y' icon.

The main header displays the repository path: 'yasminebenslimane123 / Repositories / backend-awsproject / General'. To the right, it says 'Using 0 of 1 private repositories.'

Below the header is a navigation bar with tabs: 'General' (underlined), 'Tags', 'Builds', 'Collaborators', 'Webhooks', and 'Settings'.

The main content area shows the repository details: 'yasminebenslimane123/backend-awsproject'. It indicates 'Last pushed 8 minutes ago'. There are buttons to 'Add a description' (with an 'INCOMPLETE' status) and 'Add a category' (also 'INCOMPLETE').

To the right, under 'Docker commands', is a box containing the command 'docker push yasminebenslimane123/backend-awsproject:tagname'. A 'Public view' button is located at the bottom right of this box.

And here it is in the Docker Hub repository.

2-Creation of backend task definition

The screenshot shows the AWS Task Definition creation interface. At the top, a green success message states: "Task definition successfully created backend-task-definition:1 has been successfully created. You can use this task definition to deploy a service or run a task." Below the message are three buttons: "View task definition", "Deploy", "Actions", and "Create new revision". The main section is titled "backend-task-definition:1". It contains several tabs: "Overview" (selected), "JSON", "Task placement", "Volumes (0)", "Requires attributes", and "Tags". The "Overview" tab displays the following details:

ARN	Status	Time created	App environment	
arn:aws:ecs:us-east-1:172615386837:task-definition/backend-task-definition:1	ACTIVE	December 31, 2024 at 20:06 (UTC+1:00)	Fargate	
Task role	Task execution role	Operating system/Architecture	Network mode	
LabRole	LabRole	Linux/x86_64	awsvpc	
Fault injection	Turned off			

Below this are two charts: "Task size" and "Task memory". The "Task size" chart shows Task CPU (1,024 units) and Task CPU maximum allocation for containers (1,000 units). The "Task memory" chart shows Task memory (2,048 MB) and Task memory maximum allocation for container memory reservation (2,000 MB).

The "Containers" tab shows one container named "backend-container" with the following details:

Container name	Image	Private registry	Essential	CPU	Memory hard/soft limit	GPU
backend-container	yasminebenslimane123/backend-awsproject	-	Yes	1 vCPU	2 GB/1 GB	-

3- Creation of the Backend Service

When creating a service in an ECS cluster, we noticed that when configuring a load balancer via the service creation interface, it is automatically set as an "internet-facing" load balancer. However, for a backend, it should be configured as an "internal" load balancer. Therefore, I had to configure this manually.

ecs-backend-alb

[Actions ▾](#)

▼ Details			
Load balancer type	Status	VPC	Load balancer IP address type
Application	Active	vpc-0b6fdbbb4a6bca32b	IPv4
Scheme	Hosted zone	Availability Zones	Date created
Internal	Z35SXDOTRQ7X7K	subnet-044aa7bf6adc48e96 us-east-1a (use1-az4)	December 31, 2024, 22:13 (UTC+01:00)
Load balancer ARN			DNS name Info
arn:aws:elasticloadbalancing:us-east-1:172615386837:loadbalancer/app/ecs-backend-alb/33f6f55f834e29c5			internal-ecs-backend-alb-1547380488.us-east-1.elb.amazonaws.com (A Record)

[Listeners and rules](#) [Network mapping](#) [Resource map - new](#) [Security](#) [Monitoring](#) [Integrations](#) [Attributes](#) [Capacity - new](#) [Tags](#)

Listeners and rules (1) Info		Manage rules ▾	Manage listener ▾	Add listener
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.				
<input type="checkbox"/> Protocol:Port	Default action	Rules	ARN	Security policy
<input type="checkbox"/> HTTP:80	Forward to target group • ecs-backend-tg : 1 (100%) • Target group stickiness: Off	1 rule	ARN	Not applicable

For this load balancer, we defined an IP type target group and configured the targets as follows:

- **Port:** 3200 — the port on which our application is listening
- **IP:** 10.0.3 and 10.0.7 (corresponding to the backend subnets we configured)

ecs-backend-tg

[Actions ▾](#)

Details					
arn:aws:elasticloadbalancing:us-east-1:172615386837:targetgroup/ecs-backend-tg/5ac6df481930ff6a	Protocol : Port	Protocol version			
Target type	HTTP: 3200	HTTP1			
IP	Load balancer	VPC			
IP address type	ecs-backend-alb	vpc-0b6fdbbb4a6bca32b			
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.					
Last fetched 4 minutes ago					
Zone	Total targets	Healthy	Unhealthy	Unused	Initial
us-east-1a (use1-az4)	1	1	0	0	0
us-east-1b (use1-az6)	1	1	0	0	0

Here is the backend service:

backend-service [Info](#)

Last updated December 31, 2024 at 23:15 (UTC+1:00) [Edit](#) [Update service](#) [Delete service](#)

Service overview [Info](#)

Status Active	Tasks (2 Desired)	Task definition: revision backend-task-definition:1	Deployment status Success
0 Pending 2 Running			

Health and metrics [Tasks](#) [Logs](#) [Deployments](#) [Events](#) [Configuration and networking](#) [Service auto scaling](#) [Tags](#)

Service configuration [Info](#)

Service ARN arn:aws:ecs:us-east-1:1172615386837:service/aws-project-cluster/backend-service	Task definition: revision backend-task-definition:1	Service type REPLICA	Created by arn:aws:iam::1172615386837:role/vocabs
Platform version LATEST	Platform family Linux	Amazon ECS managed tags Turned on	Propagate tags from None
Availability Zone rebalancing Turned on	CloudFormation stack backend-service		

Capacity provider strategy (1)

Capacity provider	Base	Weight
FARGATE	0	1

Load balancers

Load balancer type Application Load Balancer	Listener protocol:port HTTP:80	Target group name:protocol ecs-backend-tg:HTTP	Health check path /health
--	--	--	---

Fargate ephemeral storage

Here is the network configuration of our service: We configured it on the backend subnets of the availability zones **us-east-1a** and **us-east-1b**.

Network configuration

Network vpc-0b6fdbbb4a6bca32b	Security groups sg-04987b6f5edec682	Service role AWSServiceRoleForECS	Load balancers ecs-backend-alb
Subnets subnet-044aa7bf6adc48e96 subnet-031d5fb02ac4a5691	Auto-assign public IP Turned off	Health check grace period 300	DNS names internal-ecs-backend-alb-1547380488.us-east-1.elb.amazonaws.com open address

The number of running tasks matches the desired count we specified .

Here is an excerpt from the task logs:

Logs (73+) Info		View in CloudWatch Edit	
Filter logs - press Enter to search		Filter container	Filter date time range
Timestamp (UTC+01:00)	Message	Filter container	Filter date time range
December 31, 2024 at 23:02 (UTC+1:00)	2024-12-31 22:02:31 [INFO]: Connected to MySQL Database	backend-container	Since 1 hour ago
December 31, 2024 at 23:02 (UTC+1:00)	Server is running on port http://localhost:3200	backend-container	
December 31, 2024 at 23:02 (UTC+1:00)	2024-12-31T22:02:31: PM2 log: App [server:0] online	backend-container	
December 31, 2024 at 23:02 (UTC+1:00)	2024-12-31T22:02:31: PM2 log: App [server:0] starting in -fork mode-	backend-container	
December 31, 2024 at 23:02 (UTC+1:00)	2024-12-31T22:02:31: PM2 log: Launching in no daemon mode	backend-container	

As we can observe, our application is listening on port 3200 and has successfully connected to the database we created earlier.

4- Autoscaling and Monitoring of the Backend Service:

We defined the number of running tasks to be between 1 and 3 and specified a scaling policy as follows: CPU usage must be maintained at 70% for this service.

The screenshot shows the AWS Service Auto Scaling console. At the top, there are tabs: Health and metrics, Tasks, Logs, Deployments, Events, Configuration and networking, Service auto scaling (which is selected), and Tags. Below the tabs, a section titled "How to use service auto scaling" is shown. Under "Task count", it says "Number of tasks to run for backend-service" with a dropdown menu showing "1 - 3". There are buttons for "Suspend/Unsuspend scaling", "Edit", and "Set the number of tasks". Below this, there are three tabs: "Scaling policies" (selected), "Scheduled actions", and "Recommendations". The "Scaling policies" tab shows one policy named "back-service-auto-scaling-policy" with a status of "Active". The "Scaling mechanism" is set to "Target Tracking" and the "Target metric" is "ECSServiceAverageCPUUtilization (70%)". There are buttons for "Actions" and "Create scaling policy".

The screenshot shows the details of a scaling policy named "back-service-auto-scaling-policy". It includes the ARN: arn:aws:iam::172615386837:role/aws-service-role/ecs.application-autoscaling.amazonaws.com/AWSServiceRoleForApplicationAutoScaling_ECSService. The policy was created on December 31, 2024 at 21:21 (UTC). It has a resource name "backend-service" and a scalable dimension "ecsservice:DesiredCount". The policy type is "Target Tracking". The target is set to "ECSServiceAverageCPUUtilization (70%)". Additional settings include a scale in cooldown period of 300 and a scale out cooldown period of 300. The status is "Active". There is a "Close" button at the bottom right.

For the deployment of the frontend :

1-Dockerisation du frontend:

Avant de créer la task definition du frontend, on doit tout d'abord transformer notre application en image docker, qu'on va mettre sur Dockerhub

To do this:

- We navigated to the project's **frontend** folder and added this **Dockerfile**:

```

frontend > 📄 Dockerfile
1  # Use a Node.js image for building the app
2  FROM node:18 AS builder
3
4  # Set the working directory inside the container
5  WORKDIR /app
6
7  # Copy package files and install dependencies
8  COPY package.json package-lock.json ./"
9  RUN npm install
10
11 # Copy the rest of the source code
12 COPY . .
13
14 # Build the application
15 RUN npm run build
16
17 # Use an NGINX image for serving the app
18 FROM nginx:1.21
19
20 # Copy the build output from the builder stage to NGINX's web root
21 COPY --from=builder /app/dist /usr/share/nginx/html
22
23 # Copy NGINX configuration
24 COPY nginx.conf /etc/nginx/conf.d/default.conf
25
26 # Expose port 80 for the container
27 EXPOSE 80
28
29 # Start NGINX
30 CMD ["nginx", "-g", "daemon off;"]
31

```

- And added this `nginx.conf` file (containing the DNS name of our backend ALB, which we configured for ECS):

```

frontend > ⚙ nginx.conf
  1 ✓ server {
  2     listen 80;
  3     server_name _;
  4
  5     root /usr/share/nginx/html;
  6     index index.html index.htm;
  7
  8     # Health check endpoint
  9     location /health {
 10         default_type text/html;
 11         return 200 "<!DOCTYPE html><p>Health check endpoint</p>\n";
 12     }
 13
 14     location / {
 15         try_files $uri /index.html;
 16     }
 17
 18     location /api/ {
 19         proxy_pass http://internal-ecs-backend-alb-1547380488.us-east-1.elb.amazonaws.com;
 20         proxy_set_header Host $host;
 21         proxy_set_header X-Real-IP $remote_addr;
 22         proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
 23         proxy_set_header X-Forwarded-Proto $scheme;
 24     }
 25 }
 26

```

- Then , we started docker engine and executed the next command :

docker build -t yasminebenslimane123/frontend-awsproject .

```

PS C:\Users\yasm\Downloads\AWS_project\lirw-react-node-mysql-app\frontend> docker build -t yasminebenslimane123/frontend-awsproject .
[+] Building 2.2s (17/17) FINISHED
=> [internal] load build definition from Dockerfile
=> [internal] load metadata for docker.io/library/nginx:1.21
=> [internal] load metadata for docker.io/library/node:18
=> [auth] library/node:pull token for registry-1.docker.io
=> [auth] library/nginx:pull token for registry-1.docker.io
=> [internal] load .dockerignore
=> => transferring context: 28
=> [stage-1 1/3] FROM docker.io/library/nginx:1.21@sha256:2bcabc23b45489fb0885d69a06bal648aeda973fae7bb981bafbb884165e514
=> [builder 1/6] FROM docker.io/library/node:18@sha256:7f31aleb14c61719b8bb0eaa029310cc33851f71d3578cc422b390f8096977c5
=> [internal] load build context
=> => transferring context: 1.77kB
=> CACHED [builder 2/6] WORKDIR /app
=> CACHED [builder 3/6] COPY package.json package-lock.json .
=> CACHED [builder 4/6] RUN npm install
=> CACHED [builder 5/6] COPY .
=> CACHED [builder 6/6] RUN npm run build
=> CACHED [stage-1 2/3] COPY --from=builder /app/dist /usr/share/nginx/html
=> CACHED [stage-1 3/3] COPY nginx.conf /etc/nginx/conf.d/default.conf
=> => exporting to image
=> => exporting layers
=> => writing image sha256:2f6724d76d0fd4ba348881c9289bce8b32e22c2a4fa46087f40a30974c1e5d5
=> => naming to docker.io/yasminebenslimane123/frontend-awsproject

What's next:
  View a summary of image vulnerabilities and recommendations → docker scout quickview

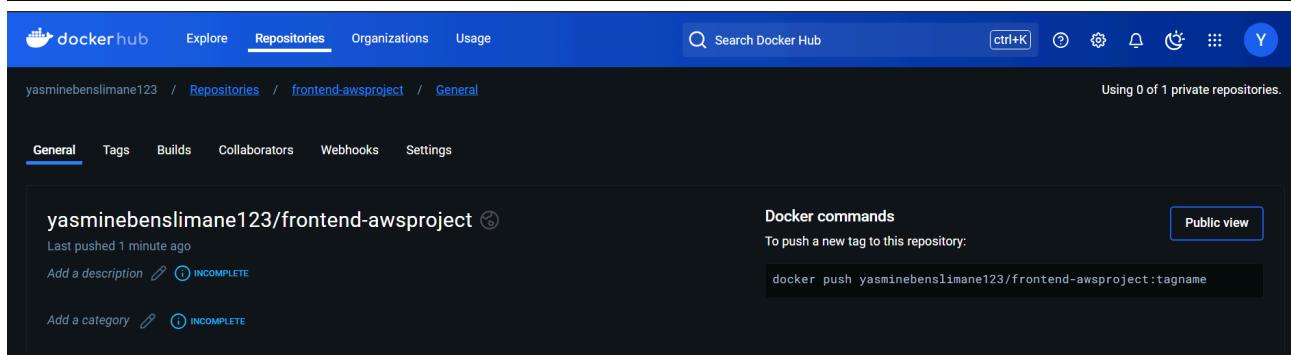
```

- We connected to docker hub :

```
PS C:\Users\yasmi\Downloads\AWS_project\lirw-react-node-mysql-app\backend> docker login -u yasminebenslimane123
Password:
Login Succeeded
```

- We pushed the image in docker hub :

```
PS C:\Users\yasmi\Downloads\AWS_project\lirw-react-node-mysql-app\frontend> docker push yasminebenslimane123/frontend-awsproject
Using default tag: latest
The push refers to repository [docker.io/yasminebenslimane123/frontend-awsproject]
2fe16b89292b: Pushed
344cee0eec9: Mounted from yasminebenslimane123/frontend-app
33e3df466e11: Pushed
747b7a567071: Pushed
57d3fc88cb3f: Pushed
53ae81198b64: Pushed
58354abe5f0e: Mounted from yasminebenslimane123/frontend-app
ad6562704f37: Mounted from yasminebenslimane123/frontend-app
latest: digest: sha256:ee4f84066d85279e09de8947ab3cf12002f8319307ecbec8b57e4c19e4f8fa18 size: 1988
PS C:\Users\yasmi\Downloads\AWS_project\lirw-react-node-mysql-app\frontend> |
```



And voila here is the Docker Hub repository.

2-Création de la task definition du frontend

Task definition successfully created
frontend-task-definition:1 has been successfully created. You can use this task definition to deploy a service or run a task.

[View task definition](#)

frontend-task-definition:1

[Deploy](#) [Actions](#) [Create new revision](#)

Overview Info	
ARN arn:aws:ecs-us-east-1:172615386837:task-definition/frontend-task-definition:1	Status ACTIVE
Task role LabRole	Task execution role LabRole
Fault injection Turned off	Time created January 01, 2025 at 01:05 (UTC+1:00)
Operating system/Architecture Linux/X86_64	App environment Fargate
Network mode awsvpc	

[Containers](#) [JSON](#) [Task placement](#) [Volumes \(0\)](#) [Requires attributes](#) [Tags](#)

Task size

Task CPU		Task memory	
1,024 units (1 vCPU)		2,048 MB (2 GB)	
Task CPU maximum allocation for containers		Task memory maximum allocation for container memory reservation	
CPU (unit)	frontend-container	Memory (MB)	frontend-container Shared task memory
0	100	0	1000
50	200	100	1100
100	300	200	1200
150	400	300	1300
200	500	400	1400
250	600	500	1500
300	700	600	1600
350	800	700	1700
400	900	800	1800
450	1000	900	1900
500	1100	1000	2000

Containers [Info](#)

Container name	Image	Private registry	Essential	CPU	Memory hard/soft limit	GPU
frontend-container	yasminebenslimane123/frontend-awsproject	-	Yes	1 vCPU	2 GB/1 GB	-

3-Création du service du frontend

Voici le service du front end:

frontend-service [Info](#)

Last updated January 01, 2025 at 01:27 (UTC+1:00) [Update service](#) [Delete service](#)

Service overview Info	
Status Active	Tasks (2 Desired) 0 Pending 2 Running
Task definition: revision frontend-task-definition:1	Deployment status In progress

[Health and metrics](#) [Tasks](#) [Logs](#) [Deployments](#) [Events](#) [Configuration and networking](#) [Service auto scaling](#) [New](#) [Tags](#)

Service configuration [Info](#)

Service ARN arn:aws:ecs-us-east-1:172615386837:service/awsprojecct-cluster/frontend-service	Task definition: revision frontend-task-definition:1	Service type REPLICA	Created by arn:aws:iam::172615386837:role/voclabs
Platform version LATEST	Platform family Linux	Amazon ECS managed tags Turned on	Propagate tags from None
Availability Zone rebalancing Turned on	CloudFormation stack frontend-service		

Capacity provider strategy (1)

Capacity provider	Base	Weight
FARGATE	0	1

Load balancers

Load balancer type Application Load Balancer	Listener protocol:port HTTP:80	Target group name:protocol ecs-awspro-frontend-service:HTTP	Health check path /health
--	--------------------------------	---	---------------------------

Fargate ephemeral storage

Voici la configuration réseau de notre service: On l'a configuré sur les subnets frontend des AZ us-east-1a et us-east-1b

Network configuration

Network
vpc-0b6fdbbb4a6bca32b

Subnets
subnet-044aa7bf6adc48e96
subnet-031d5fb02ac4a5691

Security groups
sg-04987b6f3edec682

Auto-assign public IP
Turned off

Health check grace period
300

Service role
AWSServiceRoleForECS

Load balancers
ecs-backend-alb

DNS names
internal-ecs-backend-alb-1547380488.us-east-1.elb.amazonaws.com | open address

Target groups
ecs-backend-tg

Lors de la création du front end service dans le cluster ECS, on a constaté que, lorsque l'on configure un load balancer via l'interface de création de service, celui-ci est automatiquement défini dans les frontend subnets , alors que ces subnets sont privés et ne lui permettent pas d'être accessible via internet, donc on doit changer les configuration des subnets de l'ALB créé vers les subnets publiques des deux AZ.

Network mapping [Info](#)

For internet-facing load balancers, the IPv4 addresses of the nodes are assigned by AWS. For internal load balancers, the IPv4 addresses are assigned from the subnet CIDR.

VPC
vpc-0b6fdbbb4a6bca32b
IPv4 VPC CIDR: 10.0.0.0/16
IPv6: -

Load balancer IP address type
IPv4

Mappings [Info](#)
Select at least two Availability Zones and one subnet per zone. The load balancer routes traffic to targets in these Availability Zones only. Availability Zones that are not supported by the load balancer or the VPC are not available for selection.

Availability Zones

- us-east-1a (use1-az4)

Subnet

subnet-070ef4aa9b1fad6fb	public-us-east-1a
IPv4 subnet CIDR: 10.0.1.0/24	

IPv4 address
Assigned by AWS

- us-east-1b (use1-az6)

Subnet

subnet-027e18a2dda396ea4	public-us-east-1b
IPv4 subnet CIDR: 10.0.5.0/24	

IPv4 address
Assigned by AWS

On a aussi remarqué que l'ALB du front end, a été configuré automatiquement avec le security group dédié au tasks du frontend service , donc on a dû les changer pour qu'il puisse recevoir le traffic de l'internet

Load balancer details: ecs-front-alb

Security groups
A security group is a set of firewall rules that control the traffic to your load balancer. Select an existing security group, or you can [create a new security group](#).

Security groups

Select up to 5 security groups

ecs-front-alb-sg sg-0e42e2c5c5ae7775f VPC: vpc-0b6fdbbb4a6bca32b

Cancel Save changes

Le nombre des tâches en cours d'exécution est le nombre désiré qu'on a spécifié

frontend-service [Info](#)

Last updated January 01, 2025 at 01:50 (UTC+1:00)

Service overview [Info](#)

Status Active

Tasks (2 Desired) 0 Pending | 2 Running

Task definition: revision frontend-task-definition:1

Deployment status Success

Health and metrics **Tasks** Logs Deployments Events Configuration and networking Service auto scaling [New](#) Tags

Tasks (1/2)

Task	Last status	Desired status	Task definition	Health status	Started by	Started at	Container instan...
5968058eb44e460eb220cc0fc...	Running	Running	frontend-task-definition:1	Unknown	ecs-svc/10509970068...	1 minute ago	-
a280c098613d43009c5b0030f6...	Running	Running	frontend-task-definition:1	Unknown	ecs-svc/10509970068...	2 minutes ago	-

Voici un extrait des logs des tasks :

Health and metrics Tasks **Logs** Deployments Events Configuration and networking Service auto scaling [New](#) Tags

Logs (256+) Info

You can use the filter bar below to search for and match terms, phrases, or values in your log events. [Learn more about filter patterns](#)

View in CloudWatch [CloudWatch Metrics](#)

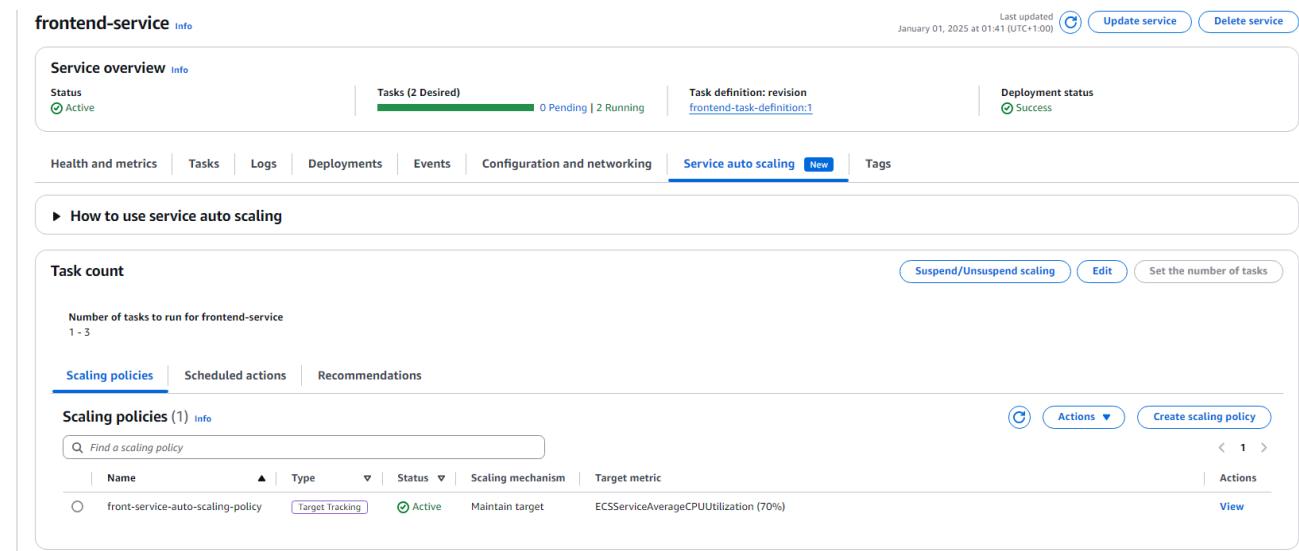
Filter container frontend-container Filter date time range Since 1 hour ago

Timestamp (UTC+01:00)	Message	Task	Container
January 01, 2025 at 01:31 (UTC+1:00)	10.0.6.30 - - [01/Jan/2025:00:31:57 +0000] "GET /health HTTP/1.1" 200 44 "-" "ELB-HealthChecker/2.0" "-"	8b0609f066534906b8	frontend-container
January 01, 2025 at 01:31 (UTC+1:00)	10.0.2.94 - - [01/Jan/2025:00:31:57 +0000] "GET /health HTTP/1.1" 200 44 "-" "ELB-HealthChecker/2.0" "-"	8b0609f066534906b8	frontend-container
January 01, 2025 at 01:31 (UTC+1:00)	10.0.1.254 - - [01/Jan/2025:00:31:41 +0000] "GET /health HTTP/1.1" 200 44 "-" "ELB-HealthChecker/2.0" "-"	8b0609f066534906b8	frontend-container
January 01, 2025 at 01:31 (UTC+1:00)	10.0.5.104 - - [01/Jan/2025:00:31:39 +0000] "GET /health HTTP/1.1" 200 44 "-" "ELB-HealthChecker/2.0" "-"	8b0609f066534906b8	frontend-container

Here are the health check requests for our application, which it successfully responds to.

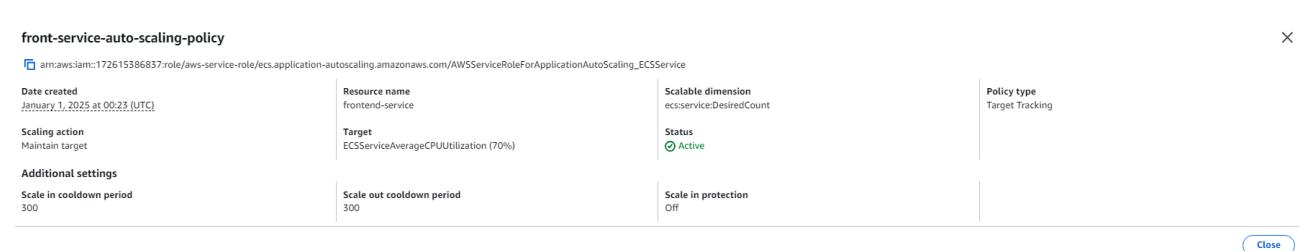
4- Autoscaling and Monitoring of the Frontend Service:

We defined the number of running tasks to be between 1 and 3 and specified the following scaling policy: CPU usage must be maintained at 70% within the service.



The screenshot shows the AWS CloudWatch Metrics Insights interface. A query is being run against CloudWatch Metrics Insights metrics over a period from January 1, 2025, to January 1, 2025. The results are displayed in a table with columns for Time, Metric Name, and Value. The table shows a single row with a value of 1.0. The 'Time' column includes a timestamp of 2025-01-01T00:00:00Z.

Time	Metric Name	Value
2025-01-01T00:00:00Z	CloudWatch Metrics Insights Metrics	1.0



The screenshot shows the AWS CloudWatch Metrics Insights interface. A query is being run against CloudWatch Metrics Insights metrics over a period from January 1, 2025, to January 1, 2025. The results are displayed in a table with columns for Time, Metric Name, and Value. The table shows a single row with a value of 1.0. The 'Time' column includes a timestamp of 2025-01-01T00:00:00Z.

Time	Metric Name	Value
2025-01-01T00:00:00Z	CloudWatch Metrics Insights Metrics	1.0

Phase 5 : Intégration DNS et Sécurité Avancée

11. Monitoring et Alertes :

- We configured CloudWatch to monitor performance metrics (CPU, memory, HTTP requests).
-
- We also set up alarms with notifications sent to an email address or a Slack channel via SNS.

We configured the alarm notifications to be sent to an email address subscribed to this topic channel:

The screenshot shows the AWS SNS Topic configuration page for 'Default_CloudWatch_Alarms_Topic'. The top navigation bar includes 'Edit', 'Delete', and 'Publish message' buttons. The main section is titled 'Details' and contains the following information:

Name	Default_CloudWatch_Alarms_Topic	Display name	-
ARN	arn:aws:sns:us-east-1:172615386837:Default_CloudWatch_Alarms_Topic	Topic owner	172615386837
Type	Standard		

Below the details, there are tabs for 'Subscriptions', 'Access policy', 'Data protection policy', 'Delivery policy (HTTP/S)', 'Delivery status logging', 'Encryption', 'Tags', and 'Integrations'. The 'Subscriptions' tab is selected, showing one subscription entry:

ID	Endpoint	Status	Protocol
65f9e03b-071f-4068-a693-f6e35a1c3af8	yasmine.benslimane@etudiant-fst.utm.tn	Confirmed	EMAIL

Buttons for 'Edit', 'Delete', 'Request confirmation', 'Confirm subscription', and 'Create subscription' are located at the top of the subscriptions table.

On a reçu un mail de demande de confirmation d'inscription :

AWS Notification - Subscription Confirmation External Spam x

 AWS Notifications <no-reply@sns.amazonaws.com>
to me ▾

Why is this message in spam? It is similar to messages that were identified as spam in the past.

[Report not spam](#)

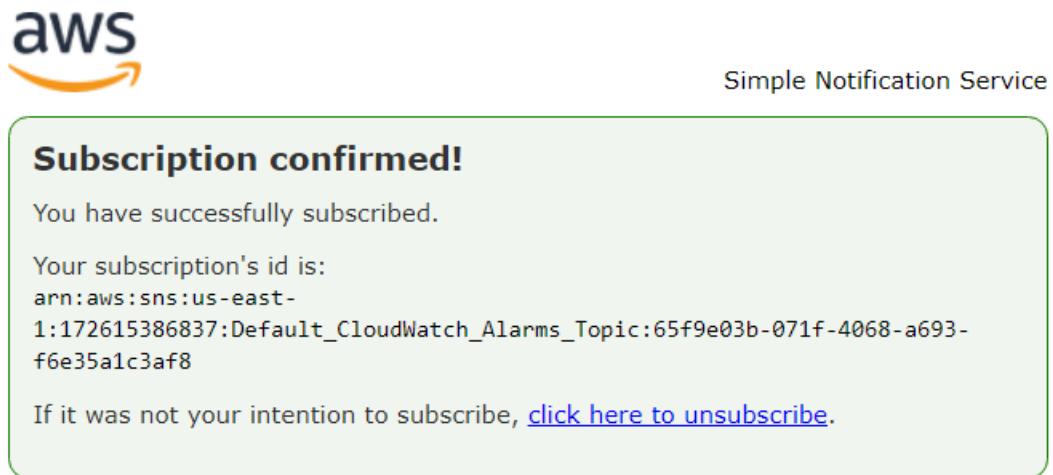
You have chosen to subscribe to the topic:
`arn:aws:sns:us-east-1:172615386837:Default_CloudWatch_Alarms_Topic`

To confirm this subscription, click or visit the link below (If this was in error no action is necessary):
[Confirm subscription](#)

Please do not reply directly to this email. If you wish to remove yourself from receiving all future SNS subscription confirmation requests please send an email to [sns-opt-out](#)

[Reply](#) [Forward](#)

Here is the email confirmation of the registration:



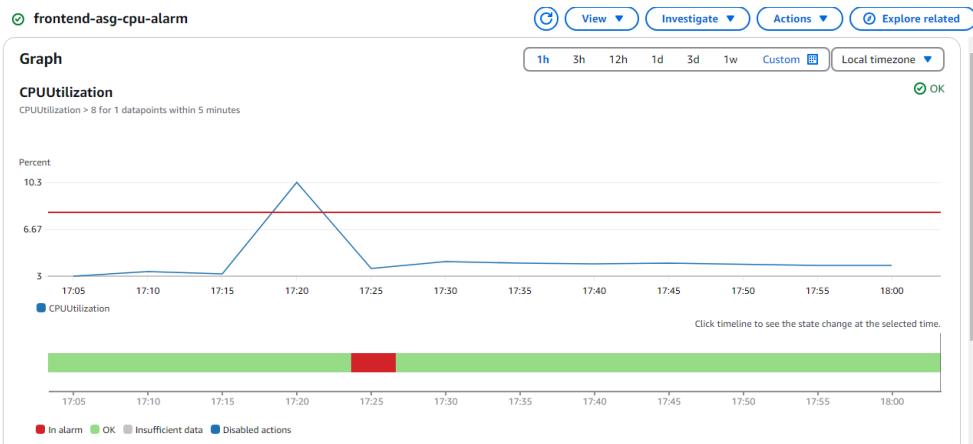
Here is the alarm for the CPU usage of the Frontend Auto Scaling Group:

Details

Name frontend-asg-cpu-alarm	State OK	Namespace AWS/EC2	Datapoints to alarm 1 out of 1
Type Metric alarm	Threshold CPUUtilization > 8 for 1 datapoints within 5 minutes	Metric name CPUUtilization	Missing data treatment Treat missing data as missing
Description No description	Last state update 2024-12-31 17:26:40 (Local)	AutoScalingGroupName frontend-autoscaling-group	Percentiles with low samples evaluate
	Actions Actions enabled	Statistic Average	ARN arn:aws:cloudwatch:us-east-1:172615386837:alarm:frontend-asg-cpu-alarm
		Period 5 minutes	

[▶ View EventBridge rule](#)

We can visualize the variation in CPU usage:



Note:

The red line indicates the thresholds we specified during the alarm creation.

When the value exceeds this threshold, we will receive a notification email like this:

ALARM: "frontend-asg-cpu-alarm" in US East (N. Virginia) [External](#) [Inbox](#)

 AWS Notifications <no-reply@sns.amazonaws.com>
to me ▾

5:23 PM (40 minutes ago) [Star](#) [Reply](#) [More](#)

You are receiving this email because your Amazon CloudWatch Alarm "frontend-asg-cpu-alarm" in the US East (N. Virginia) region has entered the ALARM state, because "Threshold Crossed: 1 out of the last 1 datapoints [15.829178044965692 (31/12/24 16:18:00)] was greater than the threshold (8.0) (minimum 1 datapoint for OK -> ALARM transition)." at "Tuesday 31 December, 2024 16:23:40 UTC".

View this alarm in the AWS Management Console:
<https://us-east-1.console.aws.amazon.com/cloudwatch/deeplink.js?region=us-east-1#alarmsV2:alarm/frontend-asg-cpu-alarm>

Alarm Details:

- Name: frontend-asg-cpu-alarm
- Description:
- State Change: OK -> ALARM
- Reason for State Change: Threshold Crossed: 1 out of the last 1 datapoints [15.829178044965692 (31/12/24 16:18:00)] was greater than the threshold (8.0) (minimum 1 datapoint for OK -> ALARM transition).
- Timestamp: Tuesday 31 December, 2024 16:23:40 UTC
- AWS Account: 172615386837
- Alarm Arn: arn:aws:cloudwatch:us-east-1:172615386837:alarm:frontend-asg-cpu-alarm

Threshold:

- The alarm is in the ALARM state when the metric is GreaterThanThreshold 8.0 for at least 1 of the last 1 period(s) of 300 seconds.

Monitored Metric:

- MetricNamespace: AWS/EC2
- MetricName: CPUUtilization
- Dimensions: [AutoScalingGroupName = frontend-autoscaling-group]
- Period: 300 seconds
- Statistic: Average
- Unit: not specified
- TreatMissingData: missing

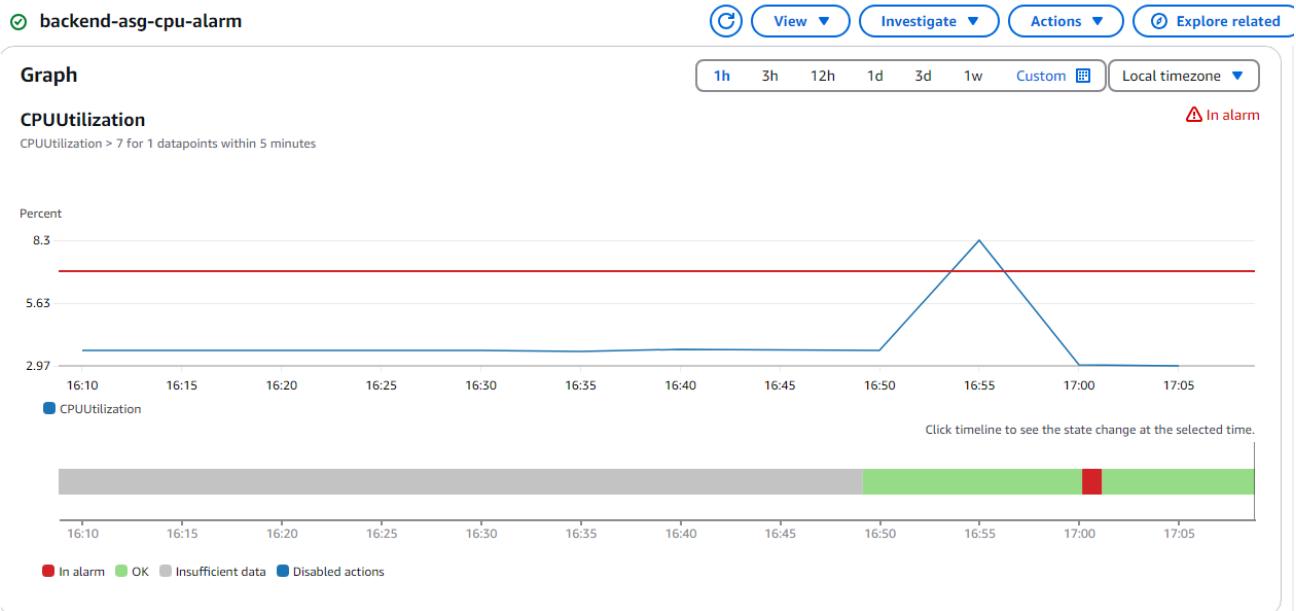
State Change Actions:

- OK:
- ALARM: [arn:aws:sns:us-east-1:172615386837:Default_CloudWatch_Alarms_Topic]
- INSUFFICIENT_DATA:

For maintenance, we repeated the same process for the backend:

[Details](#) [Tags](#) [Actions](#) [History](#) [Parent alarms](#)

Details			
Name backend-asg-cpu-alarm	State ⚠ In alarm	Namespace AWS/EC2	Datapoints to alarm 1 out of 1
Type Metric alarm	Threshold CPUUtilization > 7 for 1 datapoints within 5 minutes	Metric name CPUUtilization	Missing data treatment Treat missing data as missing
Description No description	Last state update 2024-12-31 17:00:10 (Local)	AutoScalingGroupName backend-autoscaling-group	Percentiles with low samples evaluate
	Actions ✔ Actions enabled	Statistic Average	ARN arn:aws:cloudwatch:us-east-1:172615386837:alarm:backend-asg-cpu-alarm
		Period 5 minutes	
▶ View EventBridge rule			



Notification received :

AWS Notifications <no-reply@sns.amazonaws.com>
to me ▾

5:00 PM (11 minutes ago) ⌂ ⌃ ⌚

You are receiving this email because your Amazon CloudWatch Alarm "backend-asg-cpu-alarm" in the US East (N. Virginia) region has entered the ALARM state, because "Threshold Crossed: 1 out of the last 1 datapoints [8.300920488586053 (31/12/24 15:55:00)] was greater than the threshold (7.0) (minimum 1 datapoint for OK -> ALARM transition)." at "Tuesday 31 December, 2024 16:00:10 UTC".

[View this alarm in the AWS Management Console.](#)
<https://us-east-1.console.aws.amazon.com/cloudwatch/deeplink.js?region=us-east-1#alarmsV2:alarm/backend-asg-cpu-alarm>

Alarm Details:

- Name: backend-asg-cpu-alarm
- Description:
- State Change: OK -> ALARM
- Reason for State Change: Threshold Crossed: 1 out of the last 1 datapoints [8.300920488586053 (31/12/24 15:55:00)] was greater than the threshold (7.0) (minimum 1 datapoint for OK -> ALARM transition).
- Timestamp: Tuesday 31 December, 2024 16:00:10 UTC
- AWS Account: 172615386837
- Alarm Arn: arn:aws:cloudwatch:us-east-1:172615386837:alarm:backend-asg-cpu-alarm

Threshold:
- The alarm is in the ALARM state when the metric is GreaterThanThreshold 7.0 for at least 1 of the last 1 period(s) of 300 seconds.

Monitored Metric:

- MetricNamespace: AWS/EC2
- MetricName: CPUUtilization
- Dimensions: [AutoScalingGroupName = backend-autoscaling-group]
- Period: 300 seconds
- Statistic: Average
- Unit: not specified
- TreatMissingData: missing

State Change Actions:

- OK:
- ALARM: [arn:aws:sns:us-east-1:172615386837:Default_CloudWatch_Alarms_Topic]
- INSUFFICIENT_DATA:

We also created another alarm for the front-end ALB, for which we specified the following metric:

[Details](#) | [Tags](#) | [Actions](#) | [History](#) | [Parent alarms](#)

Details	
Name frontend-alb-http-2xx-alarm	State ⚠ In alarm
Type Metric alarm	Threshold HTTPCode_Target_2XX_Count >= 2 for 1 datapoints within 5 minutes
Description No description	Last state update 2024-12-31 17:16:12 (Local)
	Actions ✔ Actions enabled
	Namespace AWS/ApplicationELB
	Datapoints to alarm 1 out of 1
	Metric name HTTPCode_Target_2XX_Count
	Missing data treatment Treat missing data as missing
	LoadBalancer app/frontend-alb/6b225e950f9c6a39
	Percentiles with low samples evaluate
	Statistic Sum
	Period 5 minutes
	ARN arn:aws:cloudwatch:us-east-1:172615386837:alarm:frontend-alb-http-2xx-alarm

▶ View EventBridge rule

Same way , we successfully created an alarm for http Frontend alb :

Details | Tags | Actions | History | Parent alarms

Details

Name front-alb-http-2xx-cpu-alarm	State OK	Namespace AWS/ApplicationELB	Datapoints to alarm 1 out of 1
Type Metric alarm	Threshold HTTPCode_Target_2XX_Count < 2 for 1 datapoints within 5 minutes	Metric name HTTPCode_Target_2XX_Count	Missing data treatment Treat missing data as missing
Description there was a very low user activity in the past 5 minutes	Last state update 2024-12-31 19:35:25 (Local)	LoadBalancer app/frontend-alb/b6225e950f9c6a39	Percentiles with low samples evaluate
	Actions Actions enabled	Statistic Sum	ARN arn:aws:cloudwatch:us-east-1:172615386837:alarm:front-alb-http-2xx-cpu-alarm
		Period 5 minutes	

OK front-alb-http-2xx-cpu-alarm

OK View ▾ OK Investigate ▾ OK Actions ▾ OK Explore related

Graph

HTTPCode_Target_2XX_Count
HTTPCode_Target_2XX_Count < 2 for 1 datapoints within 5 minutes

Count

6

4

2

18:45 18:50 18:55 19:00 19:05 19:10 19:15 19:20 19:25 19:30 19:35 19:40

HTTPCode_Target_2XX_Count

Click timeline to see the state change at the selected time.

18:45 18:50 18:55 19:00 19:05 19:10 19:15 19:20 19:25 19:30 19:35 19:40

■ In alarm ■ OK ■ Insufficient data ■ Disabled actions

Notification received :

ALARM: "front-alb-http-2xx-cpu-alarm" in US East (N. Virginia) [External](#) [Inbox](#)

AWS Notifications <no-reply@sns.amazonaws.com> to me ▾ 7:40PM (3 minutes ago)

You are receiving this email because your Amazon CloudWatch Alarm "front-alb-http-2xx-cpu-alarm" in the US East (N. Virginia) region has entered the ALARM state, because "Threshold Crossed: 1 out of the last 1 datapoints [1.0 (31/12/24 18:34:00)] was less than the threshold (2.0) (minimum 1 datapoint for OK -> ALARM transition)." at "Tuesday 31 December, 2024 18:40:25 UTC".

View this alarm in the AWS Management Console:
<https://us-east-1.console.aws.amazon.com/cloudwatch/deeplink.js?region=us-east-1#alarmsV2.alarm/front-alb-http-2xx-cpu-alarm>

Alarm Details:

- Name: front-alb-http-2xx-cpu-alarm
- Description: there was a very low user activity in the past 5 minutes
- State Change: INSUFFICIENT_DATA -> ALARM
- Reason for State Change: Threshold Crossed: 1 out of the last 1 datapoints [1.0 (31/12/24 18:34:00)] was less than the threshold (2.0) (minimum 1 datapoint for OK -> ALARM transition).
- Timestamp: Tuesday 31 December, 2024 18:40:25 UTC
- AWS Account: 172615386837
- Alarm Arn: arn:aws:cloudwatch:us-east-1:172615386837:alarm:front-alb-http-2xx-cpu-alarm

Threshold:

- The alarm is in the ALARM state when the metric is LessThanThreshold 2.0 for at least 1 of the last 1 period(s) of 300 seconds.

Monitored Metric:

- MetricNamespace: AWS/ApplicationELB
- MetricName: HTTPCode_Target_2XX_Count
- Dimensions: [LoadBalancer = app/frontend-alb/6b225e950f9c6a39]
- Period: 300 seconds
- Statistic: Sum
- Unit: not specified
- TreatMissingData: missing

State Change Actions:

- OK:
- ALARM: [arn:aws:sns:us-east-1:172615386837:Default_CloudWatch_Alarms_Topic]
- INSUFFICIENT_DATA:

Conclusion

In conclusion, we designed and deployed a robust and scalable cloud architecture on AWS, integrating modern solutions such as containers, advanced monitoring, and secrets management. By following best practices for security, high availability, and scalability, we implemented a comprehensive and optimized platform while rigorously documenting each step. This project demonstrates our ability to master AWS technologies and adopt a methodical approach to meet the needs of modern applications.