

# 1.0 Cloud Architecture Diagram

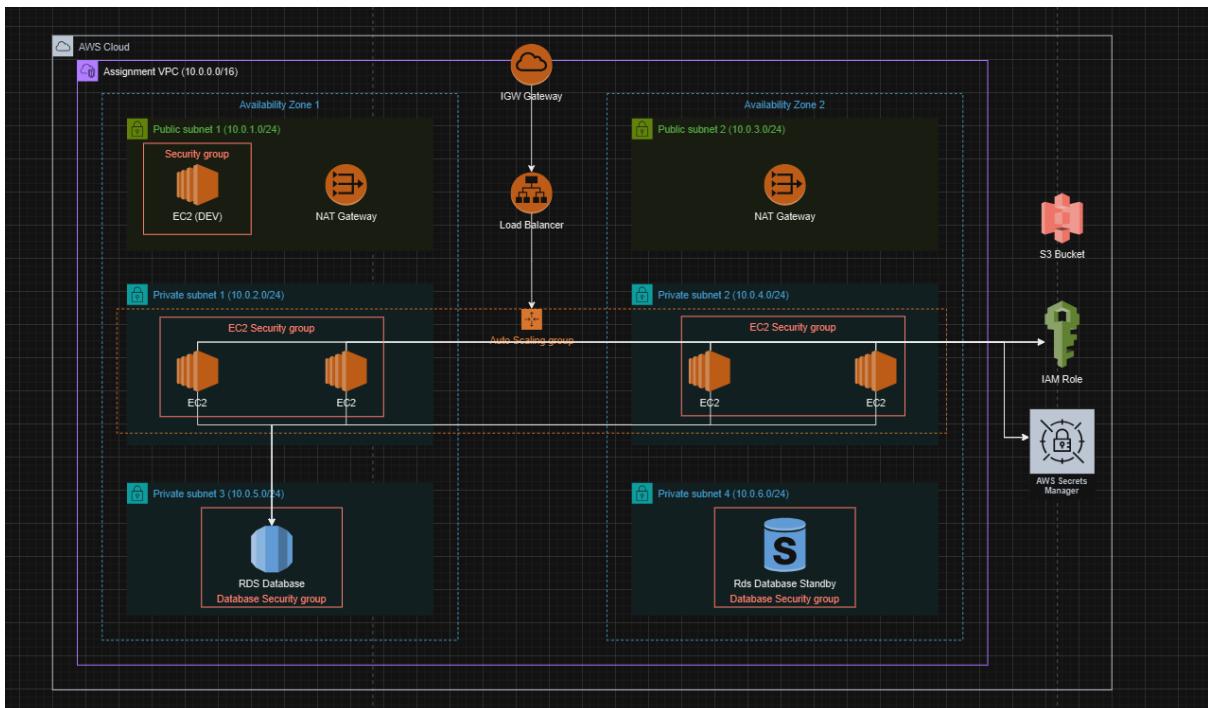


Figure 1: Cloud architecture diagram

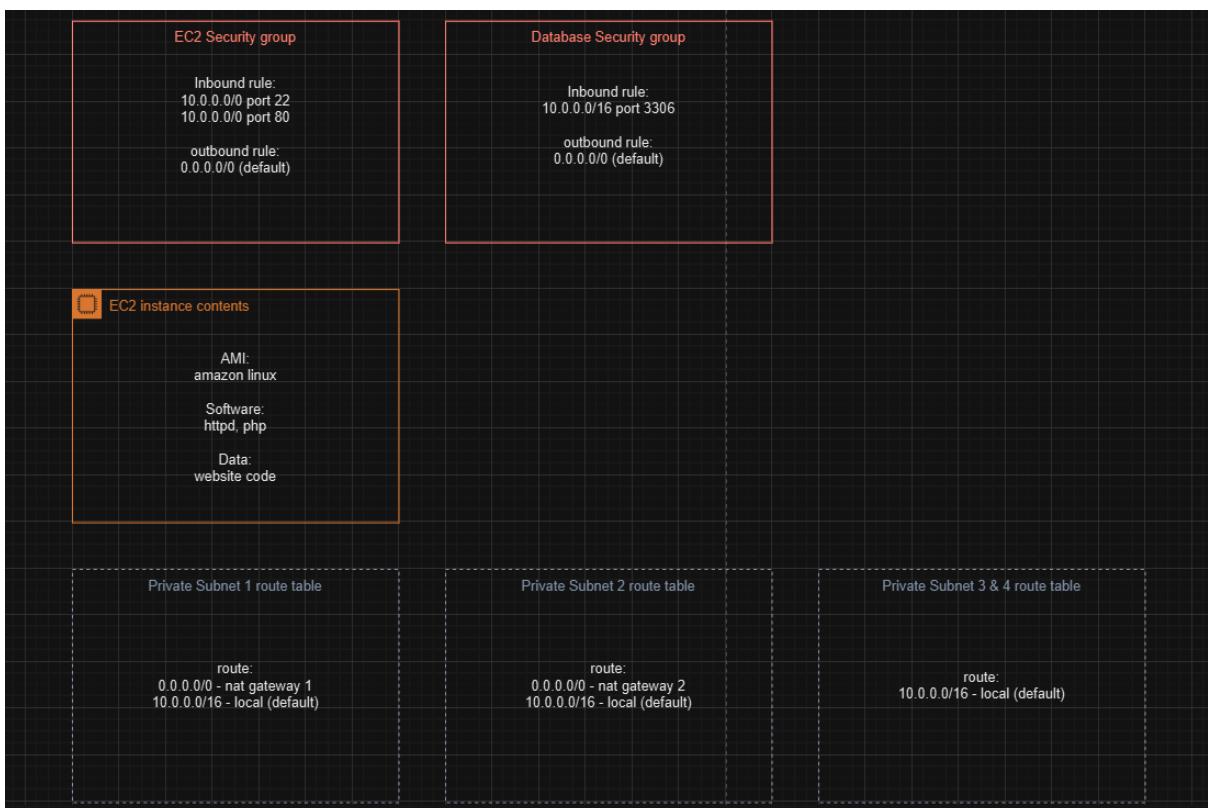


Figure 2: Security group and route table

## 2.0 Instructions to Construct the Cloud Architecture

### **Phrase 1: Creating a basic functional web application**

#### **Task 1: Create a virtual network (VPC, Internet gateway, NAT gateways, subnets, routes)**

- 1) At the top of the AWS Management Console, in the search bar, search for and choose VPC
- 2) Choose Create VPC, and configure the following:
  - **Resources to create:** Choose **VPC only**
  - **Name tag:** Enter “CinemaWebVPC”
  - **IPv4 CIDR:** Enter “10.0.0.0/16”
  - Choose **Create VPC**

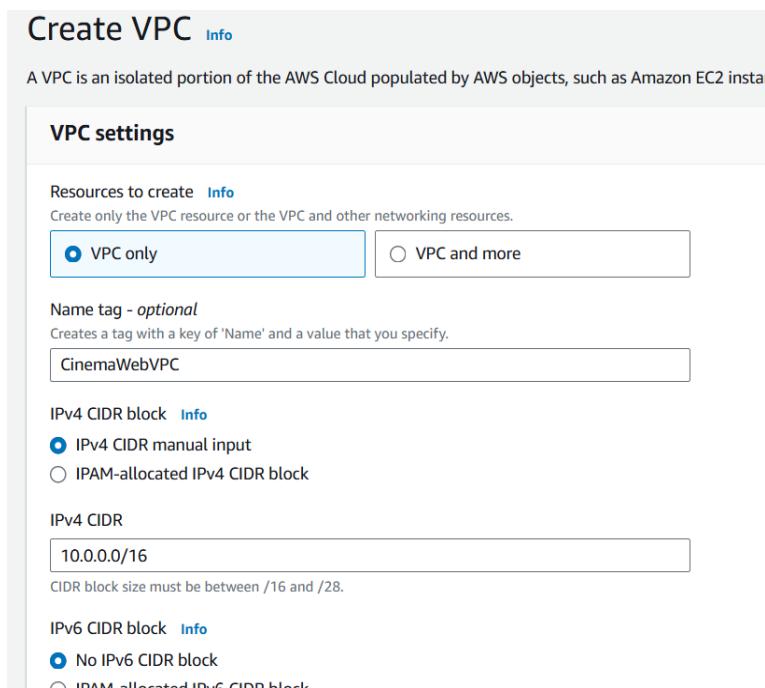


Figure 3: VPC setup

- 3) Update the settings for the VPC:
  - Choose **Actions > Edit VPC settings.**
  - In the **DNS settings** section, select **Enable DNS hostnames.**
  - Choose **Save.**

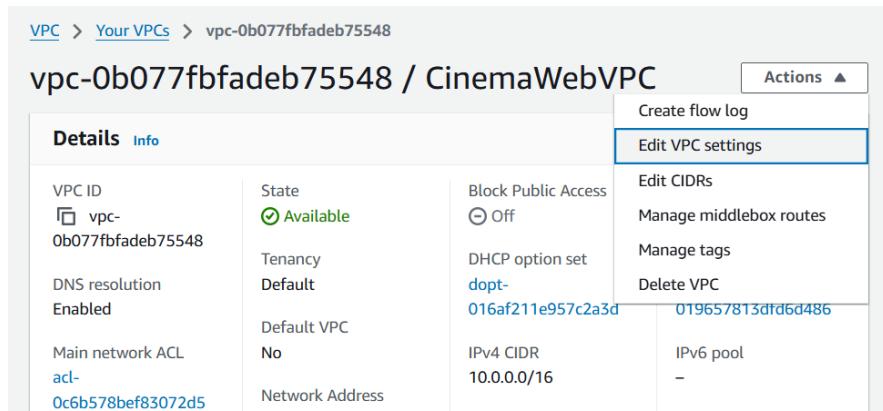


Figure 4: Editing VPC

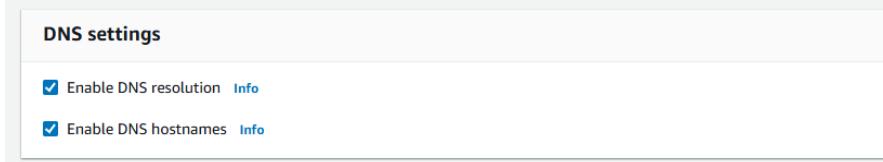


Figure 5: VPC: Enable dns setting

- 4) In the navigation pane, choose **Internet gateways**, and configure the following:
  - Choose **Create internet gateway**
  - **Name tag:** Enter “CinemaWeb-IGW”
  - Choose **Create internet gateway**

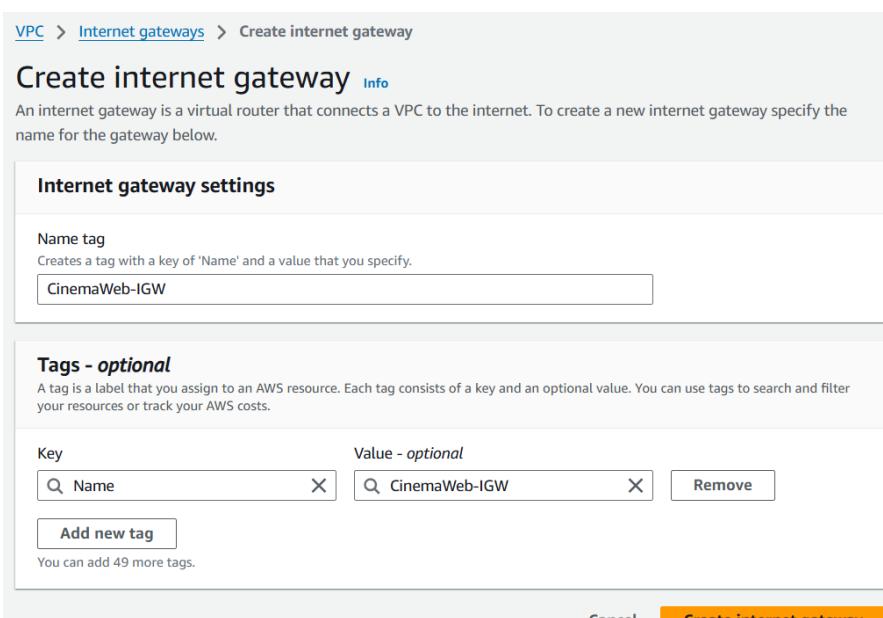


Figure 6:Creating NAT gateway

- 5) Attach the internet gateway to the VPC:
  - Choose **Actions > Attach to VPC**

- **Available VPCs:** Choose “CinemaWebVPC”
- Choose **Attach internet gateway**

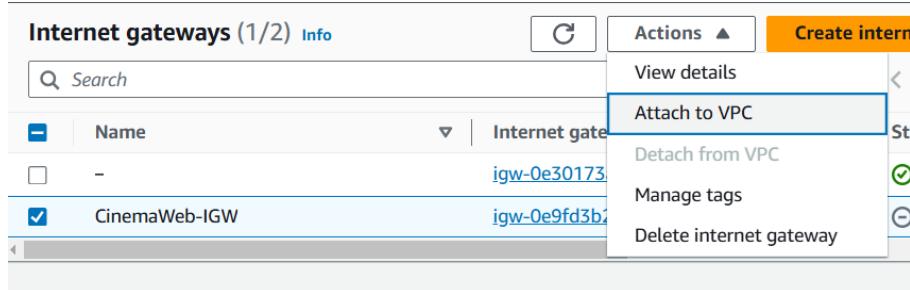


Figure 7:Attach internet gateway

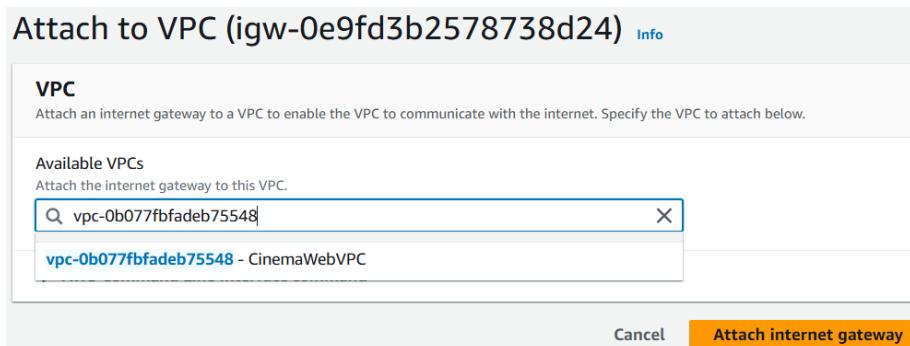


Figure 8:Attach to vpc

- 6) In the navigation pane, choose **Subnets**, and configure the following:
  - Click **Create subnet**.
  - **VPC ID:** Choose “CinemaWebVPC”
  - **Subnet name:** Enter “Public Subnet 1”
  - **Availability Zone:** Choose “ap-southeast-1a”
  - **IPv4 CIDR block:** Enter 10.0.1.0/24
  - Choose **Create subnet**.

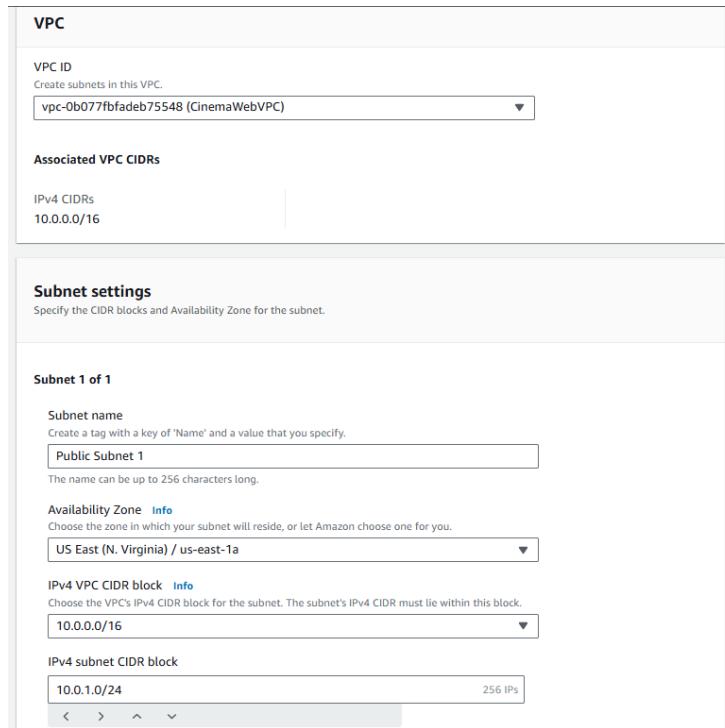


Figure 9:Creating of Public Subnet 1

Public Subnet 1 hosts EC2 instances for development (DEV) and a NAT Gateway.

7) Repeat the same steps for creating another 5 subnets following the table below:

| Subnet Name      | Availability Zone | IPv4 subnet CIDR block |
|------------------|-------------------|------------------------|
| Public subnet 2  | ap-southeast-1b   | 10.0.3.0/24            |
| Private subnet 1 | ap-southeast-1a   | 10.0.2.0/24            |
| Private subnet 2 | ap-southeast-1b   | 10.0.4.0/24            |
| Private subnet 3 | ap-southeast-1a   | 10.0.5.0/24            |
| Private subnet 4 | ap-southeast-1b   | 10.0.6.0/24            |

Public subnet 2 hosts a NAT Gateway

Private subnet 1 & Private subnet 2 host EC2 instances created by automated scaling.

Private subnet 3 hosts the primary RDS database.

Private subnet 4 hosts the RDS database standby instance.

## 8) Create route tables for public subnets:

In the navigation pane, choose **Route tables**, then click “Create route tables” and configure the following:

- Name it as “Public-RT”, and selects “CinemaWebVPC”, then click “Create route table”
- In the route tables list, choose the Public-RT
- On the **Routes** tab (in the lower pane), choose **Edit routes**.
- Choose **Add route**, and add the following route:
  - **Destination:** Enter “0.0.0.0/0”
  - **Target:** Enter “Internet Gateway” and then choose “CinemaWeb-IGW”.
- Choose **Save changes**.

| Route tables (1/7) <a href="#">Info</a>            |                                       | Last updated less than a minute ago     | <a href="#">Actions ▾</a> | <a href="#">Create route table</a> |
|--|---------------------------------------|---|---------------------------|------------------------------------|
| <a href="#">Find resources by attribute or tag</a> |                                       |   |                           |                                    |
| Name   | Route table ID                        |   |                           |                                    |
| <input checked="" type="checkbox"/> Public-RT      | <a href="#">rtb-0a80514d5</a>         |   |                           |                                    |
| <input type="checkbox"/> Private-RT-3&4            | <a href="#">rtb-007ecac46</a>         |   |                           |                                    |
| <input type="checkbox"/> -                         | <a href="#">rtb-019657813</a>         |   |                           |                                    |
| <input type="checkbox"/> -                         | <a href="#">rtb-0cb85cb59</a>         |   |                           |                                    |
| <input type="checkbox"/> -                         | <a href="#">rtb-0e164f344</a>         |   |                           |                                    |
| <input type="checkbox"/> Private-RT-1              | <a href="#">rtb-0132d79134fa02903</a> | <a href="#">subnet-0445ab99ff8745..</a> |                           |                                    |
| <input type="checkbox"/> Private-RT-2              | <a href="#">rtb-0e9d5fdc5ebd535d5</a> | <a href="#">subnet-0e20e79562651d</a>   |                           |                                    |

Figure 10:Edit routing

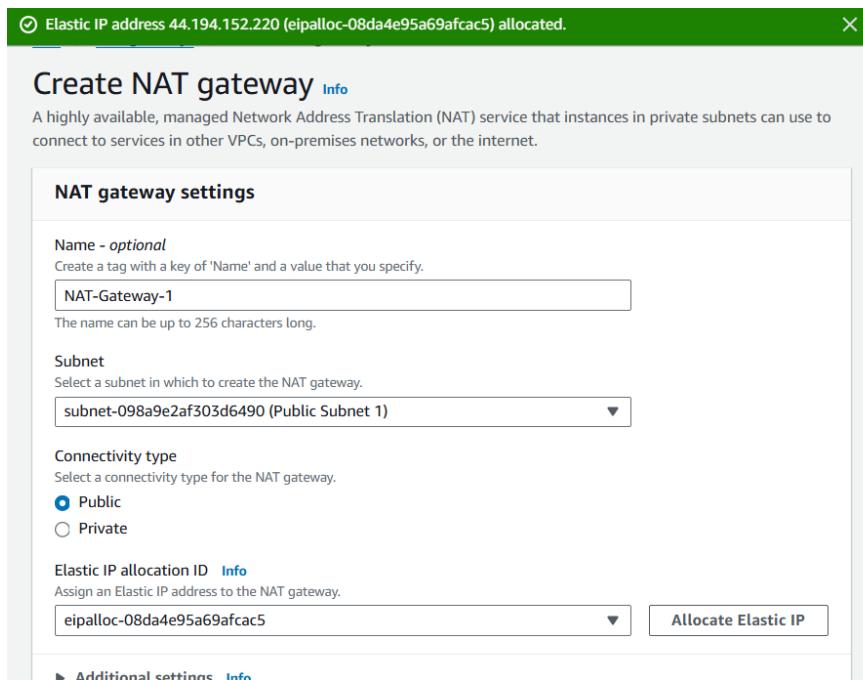
Edit routes

|  |   |  |   |
|--|---|--|---|
| <b>Route 1</b>                           | Destination<br>10.0.0.0/16                              | Target<br><input type="button" value="local"/>   | Status<br><input checked="" type="radio"/> Active |
| Propagated                               | No  |  |   |
| <b>Route 2</b>                           | Destination<br><input type="button" value="0.0.0.0/0"/> | Target<br><input type="button" value="Internet Gateway"/>  | Status<br>-                                       |
| Propagated                               | No  | <input type="button" value="igw-0e9fd3b2578738d24"/><br>Use: "igw-0e9fd3b2578738d24"<br><b>igw-0e9fd3b2578738d24</b> (CinemaWeb-IGW) |   |
| <input type="button" value="Remove"/>    |   |  |   |
| <input type="button" value="Add route"/> |   |  |   |

Figure 11: Routing configuration

## 9) Create NAT Gateway 1:

- Give name as “NAT-Gateway-1”
- Choose Public subnet 1 as subnet choice
- Connectivity type is Public
- Make sure to click the “allocate elastic IP”. Then, click “Create NAT gateway”.



## 10) Create NAT Gateway 2:

- Give name as “NAT-Gateway-2”
- Choose Public subnet 2 as subnet choice
- Connectivity type is Public
- Make sure to click the “allocate elastic IP”. Then, click “Create NAT gateway”.

## 11) Create route tables for Private subnet 1, Private subnet 2:

In the navigation pane, choose **Route tables**, then click “Create route tables” and configure the following:

- Name it as “Private-RT-1”, and selects “CinemaWebVPC”, then click “Create route table”
- In the route tables list, choose Private-RT-1
- On the **Routes** tab (in the lower pane), choose **Edit routes**.
- Choose **Add route**, and add the following route:

- **Destination:** Enter “0.0.0.0/0”
- **Target:** Enter “NAT Gateway 1” and then choose “CinemaWeb-IGW”.
- Choose **Save changes**.
- Repeat this for creating Private-RT-2 with the same configuration as Private Subnet 1, but targeting "NAT Gateway 2".

**12) Create route tables for Private subnet 3 and Private subnet 4:**

In the navigation pane, choose **Route tables**, then click “Create route tables” and configure the following:

- Name it as “Private-RT-3&4”, and selects “CinemaWebVPC”, then click “Create route table”
- In the route tables list, choose Private-RT-3&4.
- On the Routes tab, confirm that only one route is listed and that it has the following settings:
  - 1) Destination is 10.0.0.0/16.
  - 2) Target is local.

**13) In the navigation pane, choose **Subnets**, and configure the following:**

- Select **Public Subnet 1**.
- Choose **Actions > Edit subnet settings**.
- In the **Auto-assign IP settings** section, select **Enable auto-assign public IPv4 address**.
- Choose **Save**.
- Repeat this for Public Subnet 2.

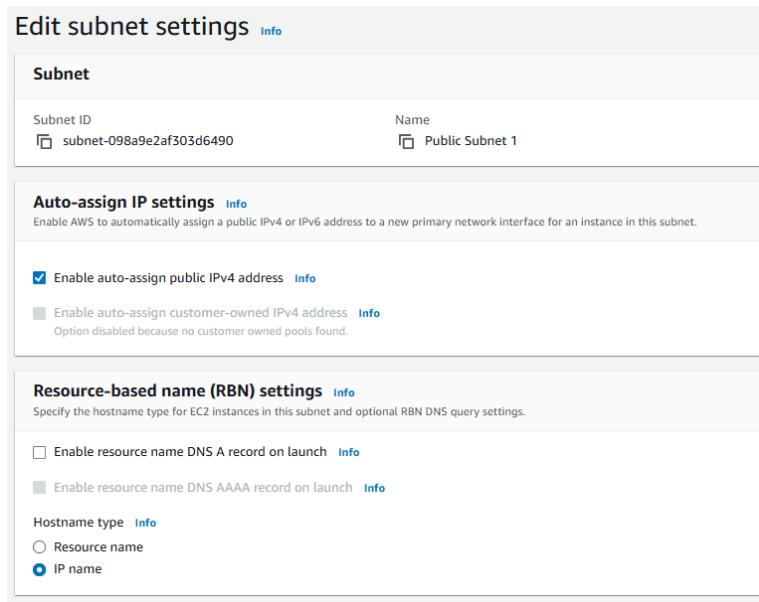


Figure 12:Enable auto assign IP setting

## Task 2: Creating a virtual machine (EC 2)

- 1) At the top of the AWS Management Console, in the search bar, search for and choose EC2
- 2) Choose **Launch instance > Launch instance**, and then configure the following:
  - 3) In the **Name and tags** section, for **Name**, enter “CinemaDevServer”
  - 4) In the **Application and OS Images** section, under **Quick Start**, choose **Amazon Linux**.
  - 5) In the **Instance Type** section, choose “t3.2xlarge”.
  - 6) In the **Key pair** section, create new key pair and enter key pair name “DevServerKey”. (remember to save the key pair file)
  - 7) In the **Network settings** section, configure the following:
    - Choose **Edit**.
    - **VPC:** Choose **CinemaWebVPC**.
    - **Auto-assign public IP:** Choose **Enable**.
    - **Firewall (security groups):** Choose **Create security group**.
    - **Security group name:** Enter “CinemaDevServer-SG”
    - Choose **Add security group rule**.
    - Keep the existing SSH rule, and add two new rules with the following settings:
    - **New rule 1:** For **Type**, choose **HTTP**. For **Source type**, choose **Anywhere**.

- **New rule 2:** For **Type**, choose **MYSQL/Aurora**. For **Source**, enter 10.0.0.0/16
- 8) Go to the **Advanced details** section. For the IAM role, choose the “LabInstanceProfile”.
  - 9) Keep the default values for all other settings and choose **Launch instance**.

**Launch an instance** Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

**Name and tags** Info

Name  
CinemaDevServer Add additional tags

---

**▼ Application and OS Images (Amazon Machine Image)** Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

**Recent** | **My AMIs** | **Quick Start**

       Browse more AMIs  
Including AMIs from AWS, Marketplace and the Community

**Amazon Machine Image (AMI)**

|  |                    |
|--|--------------------|
| <b>Amazon Linux 2023 AMI</b><br>ami-0453ec754f44f9a4a (64-bit (x86), uefi-preferred) / ami-0ed83e7a78a23014e (64-bit (Arm), uefi)<br>Virtualization: hvm ENA enabled: true Root device type: ebs | Free tier eligible |
|--|--------------------|

Figure 13:Launching Cinema Dev Server

**▼ Instance type** Info | [Get advice](#)

**Instance type**

**t3.2xlarge**  
Family: t3 8 vCPU 32 GiB Memory Current generation: true  
On-Demand RHEL base pricing: 0.448 USD per Hour  
On-Demand Linux base pricing: 0.3328 USD per Hour  
On-Demand Windows base pricing: 0.48 USD per Hour  
On-Demand SUSE base pricing: 0.4578 USD per Hour  
On-Demand Ubuntu Pro base pricing: 0.3468 USD per Hour

All generations [Compare instance types](#)

**Additional costs apply for AMIs with pre-installed software**

---

**▼ Key pair (login)** Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

**Key pair name - required**  
DevServerKey [Create new key pair](#)

Figure 14: Selection of instance type

**▼ Network settings** [Info](#)

**VPC - required** [Info](#)

vpc-0b2fb2886d4ca7043 (assignment-VPC)  
10.0.0.0/16

**Subnet** [Info](#)

subnet-09beb67a9e8a74437 public-subnet-1  
VPC: vpc-0b2fb2886d4ca7043 Owner: 890638553423  
Availability Zone: us-east-1a Zone type: Availability Zone  
IP addresses available: 248 CIDR: 10.0.1.0/24

**Create new subnet** [Info](#)

**Auto-assign public IP** [Info](#)

Enable

Additional charges apply when outside of [free tier allowance](#)

**Firewall (security groups)** [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group  Select existing security group

**Security group name - required**

CinemaDevServer-SG

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and \_-:/()#,@[]+=&;\$\*

**Description - required** [Info](#)

launch-wizard-1 created 2024-11-24T07:09:34.024Z

Figure 15: Firewall security set up rule

**Inbound Security Group Rules**

**▼ Security group rule 1 (TCP, 22, 14.192.213.159/32)** [Remove](#)

| Type | Protocol | Port range |
|------|----------|------------|
| ssh  | TCP      | 22         |

| Source type | Name              | Description - optional     |
|-------------|-------------------|----------------------------|
| My IP       | 14.192.213.159/32 | e.g. SSH for admin desktop |

**▼ Security group rule 2 (TCP, 80, 0.0.0.0/0)** [Remove](#)

| Type | Protocol | Port range |
|------|----------|------------|
| HTTP | TCP      | 80         |

| Source type | Source    | Description - optional     |
|-------------|-----------|----------------------------|
| Anywhere    | 0.0.0.0/0 | e.g. SSH for admin desktop |

**▼ Security group rule 3 (TCP, 3306, 10.0.0.0/16)** [Remove](#)

| Type         | Protocol | Port range |
|--------------|----------|------------|
| MYSQL/Aurora | TCP      | 3306       |

| Source type | Source      | Description - optional     |
|-------------|-------------|----------------------------|
| Custom      | 10.0.0.0/16 | e.g. SSH for admin desktop |

[Add security group rule](#)

Figure 16: Security group setup rule

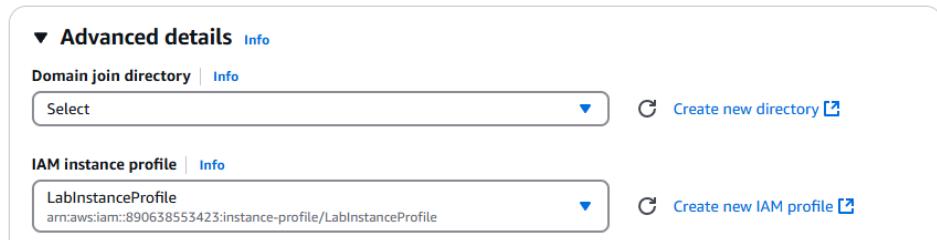


Figure 17: Assigning IAM profile

10) After the instance is created and its status is in running status,

- Go to navigation pane, choose **Elastic IP addresses**
- Click **Allocate Elastic IP address**
- Click **Allocate** (One ip address will be generated)

11) Then, go back to the Elastic Ip addresses page, and click “Allocate Elastic IP address” then choose the instance we just created “CinemaWebServer”. (Now, the CinemaWebServer instance’s IP address will be always same as the elastic address you allocated to it. This serve as a static ip address which is good for easier management because even after the instance being stopped and restarted, the instance will remain the same public IP address.)

[EC2](#) > [Elastic IP addresses](#) > Associate Elastic IP address

[Associate Elastic IP address](#) [Info](#)

Choose the instance or network interface to associate to this Elastic IP address (52.202.78.109)

**Elastic IP address: 52.202.78.109**

**Resource type**  
Choose the type of resource with which to associate the Elastic IP address.

Instance  
 Network interface

If you associate an Elastic IP address with an instance that already has an Elastic IP address associated, the previously associated Elastic IP address will be disassociated, but the address will still be allocated to your account. [Learn more](#)

If no private IP address is specified, the Elastic IP address will be associated with the primary private IP address.

**Instance**

Choose an instance  
i-097059cb76fce369e (CinemaWebServer) - running

The private IP address with which to associate the Elastic IP address.

Choose a private IP address

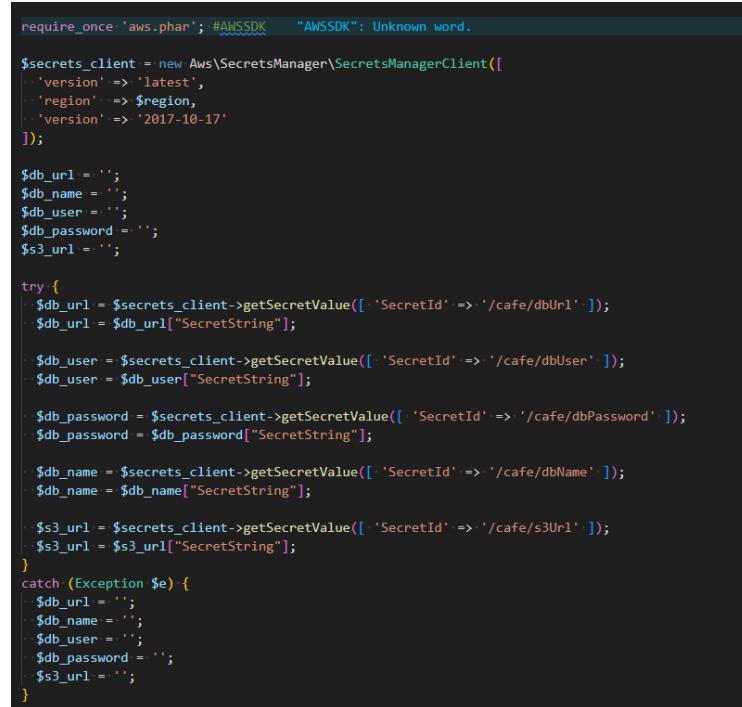
**Reassociation**  
Specify whether the Elastic IP address can be reassigned with a different resource if it already associated with a resource.

Allow this Elastic IP address to be reassigned

[Cancel](#) [Associate](#)

### Task 3: Setup ec2 environment and testing.

- 1) We have to edit our website source code to ensure it can work well with the aws environment. So, we will be using AWS Secret Manager to store our environment variables. We will be using AWS PHP SDK to retrieve those variables in our php code.



```
require_once 'aws.phar'; #AWSSDK    "AWS SDK": Unknown word.

$secrets_client = new Aws\SecretsManager\SecretsManagerClient([
    'version' => 'latest',
    'region'  => $region,
    'version' => '2017-10-17'
]);

$db_url = '';
$db_name = '';
$db_user = '';
$db_password = '';
$s3_url = '';

try {
    $db_url = $secrets_client->getSecretValue(['SecretId' => '/cafe/dbUrl']);
    $db_url = $db_url["SecretString"];

    $db_user = $secrets_client->getSecretValue(['SecretId' => '/cafe/dbUser']);
    $db_user = $db_user["SecretString"];

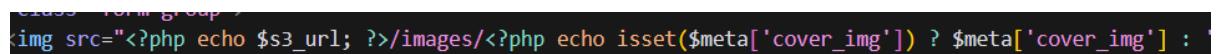
    $db_password = $secrets_client->getSecretValue(['SecretId' => '/cafe/dbPassword']);
    $db_password = $db_password["SecretString"];

    $db_name = $secrets_client->getSecretValue(['SecretId' => '/cafe/dbName']);
    $db_name = $db_name["SecretString"];

    $s3_url = $secrets_client->getSecretValue(['SecretId' => '/cafe/s3Url']);
    $s3_url = $s3_url["SecretString"];
}
catch (Exception $e) {
    $db_url = '';
    $db_name = '';
    $db_user = '';
    $db_password = '';
    $s3_url = '';
}
```

Figure 18:EC2 environment testing

- 2) Since later we will store all the images in S3 bucket, so some changes needed to be done to make sure the website can access, upload, remove images from S3 bucket. We need to modify image src path to accept dynamic URL.



```
<?php echo $s3_url; ?>/images/<?php echo isset($meta['cover_img']) ? $meta['cover_img'] : '';
```

Figure 19:Modify source in php

- 3) Now, we will set up ec2 environment to hosting our website. We will be using SSH to connect the ec2 instance. (This instruction will be using OpenSSH command, windows 10 build 1809 & above and windows 11 have built-in OpenSSH)
- 4) Open Terminal, then connect to the EC2 Instance by typing “**ssh -i “<file path of downloaded SSH key>” ec2-user@<your-ec2-ip-address>**”

```

PS C:\Users\HP> ssh -i "C:\Users\HP\Downloads\test0.pem" ec2-user@52.202.78.109
'_
#_
~~ \_#####_      Amazon Linux 2023
~~ \###|
~~ \#/ ___ https://aws.amazon.com/linux/amazon-linux-2023
~~ \V~' '-->
~~ /_/
~~ ./_/_
/_/ /_/
/m/_

```

- 5) After successfully connecting to the ec2, exit the ec2 by typing “exit”. Now we need to upload source code of the website to ec2. Direct to the directory where you stored the ssh key file and zip file of source code, type “scp -i "<file path of downloaded SSH key> " "<file path of the zip file>" ec2-user@<your-ec2-ip-address>:~”

```

PS C:\Users\HP> cd C:\Users\HP\Downloads
PS C:\Users\HP\Downloads> scp -i "C:\Users\HP\Downloads\test0.pem" "book_movie_web.zip" ec2-user@52.202.78.109:/var/www/html/
book_movie_web.zip                                         100%   13MB  2.7MB/s  00:04
PS C:\Users\HP\Downloads> ssh -i "C:\Users\HP\Downloads\test0.pem" ec2-user@52.202.78.109
#_

```

- 6) After transferring the zip file to EC2, connect to the EC2 Instance again. Then execute the following command.

```

# install required package
sudo yum install httpd
sudo yum install php
sudo yum install php-mysqli
sudo yum install mariadb105-server

# start service
sudo systemctl enable httpd
sudo systemctl start httpd
sudo systemctl enable mariadb
sudo systemctl start mariadb

# create variable at secrets manager
aws secretsmanager create-secret --name "/cafe/dbUrl" --secret-string "localhost"
aws secretsmanager create-secret --name "/cafe/dbName" --secret-string "theater_db"
aws secretsmanager create-secret --name "/cafe/dbUser" --secret-string "root"
aws secretsmanager create-secret --name "/cafe/dbPassword" --secret-string "root"
aws secretsmanager create-secret --name "/cafe/s3Url" --secret-string "<your ec2 ip address>"

# unzip file
sudo unzip ~/<zip file name> -d /var/www/html

# local database setup
cd /var/www/html
echo "SET PASSWORD FOR 'root'@'%' = PASSWORD('root');" > setPassword.sql
sudo mariadb -p < setPassword.sql # should prompt to enter password, just press enter key
sudo mariadb -p < theater_db.sql # should prompt to enter password, password is 'root'

```

Figure 20:Configuration in Open SSH by using the .pem key file

- 7) After executing the command above, the ec2 environment should be complete setup. Now you can go to web browser and navigate to “http://<your ec2 ip address>”. You should be able to see the website.

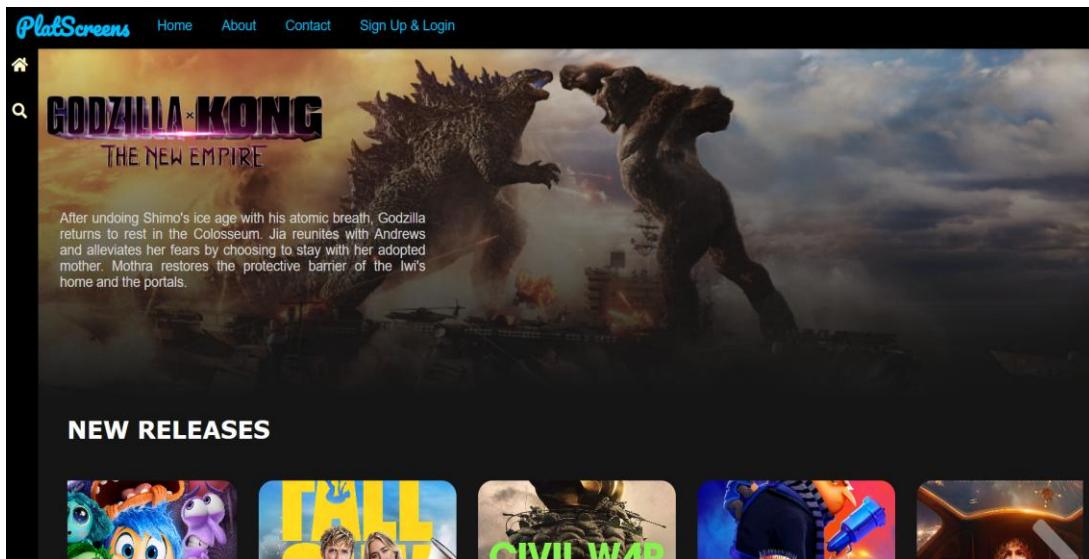


Figure 21: Movie booking platform website

- In the user page, ensure that you can register account, login account, see the most up-to-date movie information, book movie from the website, view your booking history, manage your user profile and change password.

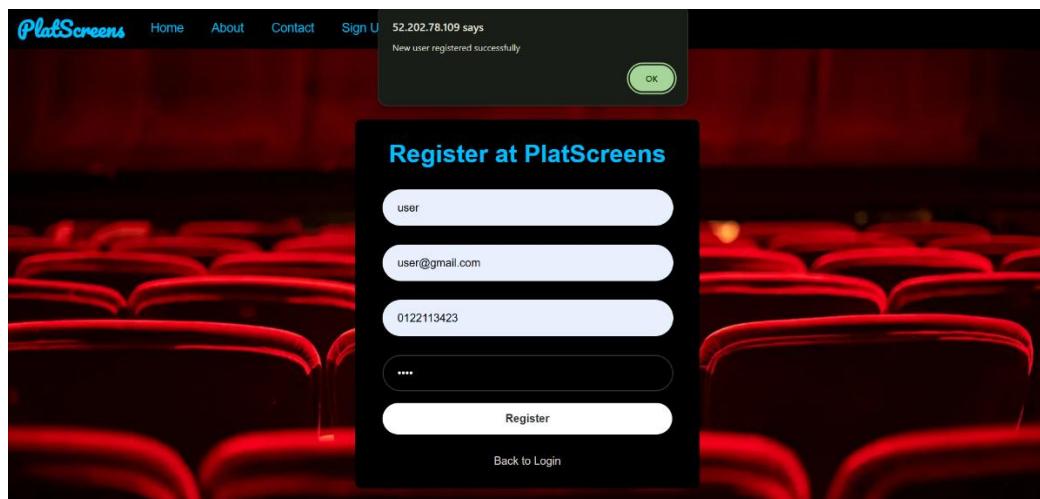


Figure 22: User registration in our platform

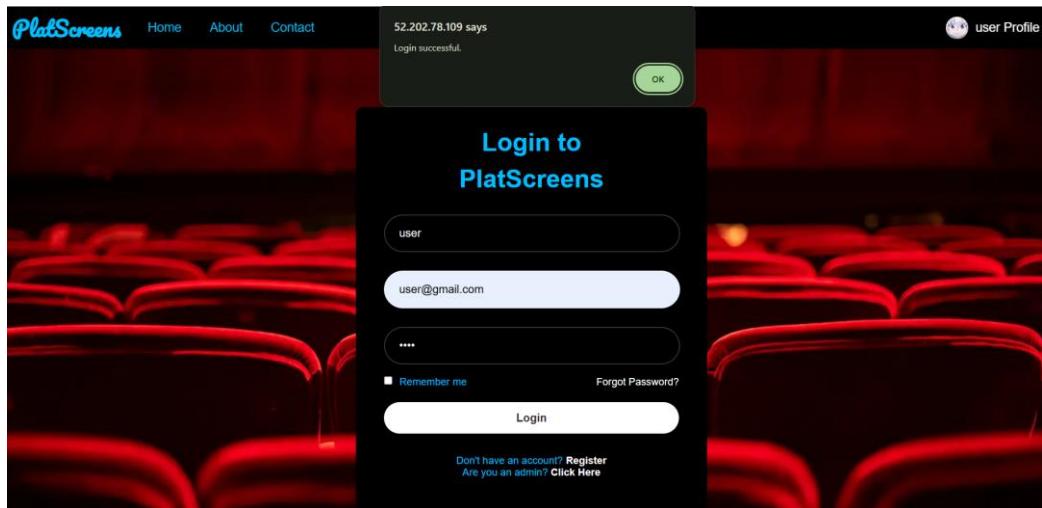


Figure 23: Signing in with existing user details in our platform

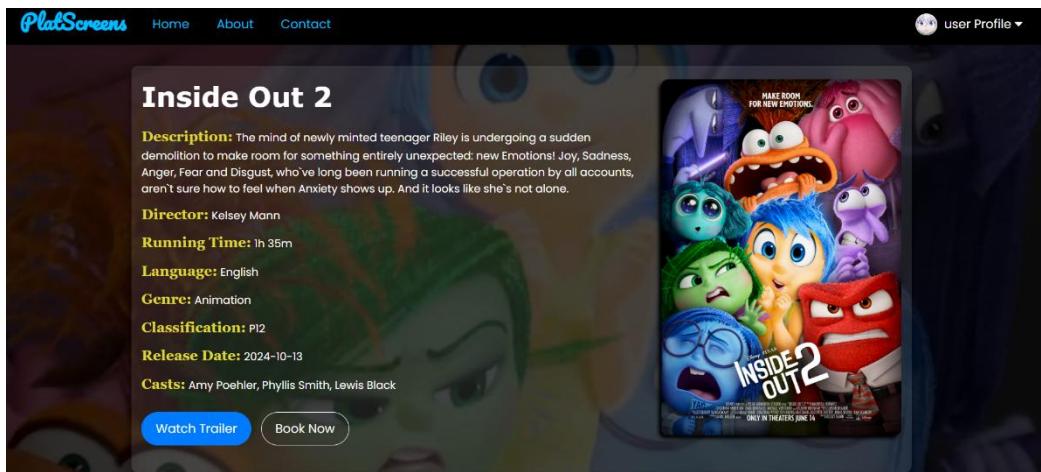


Figure 24: browsing movie details

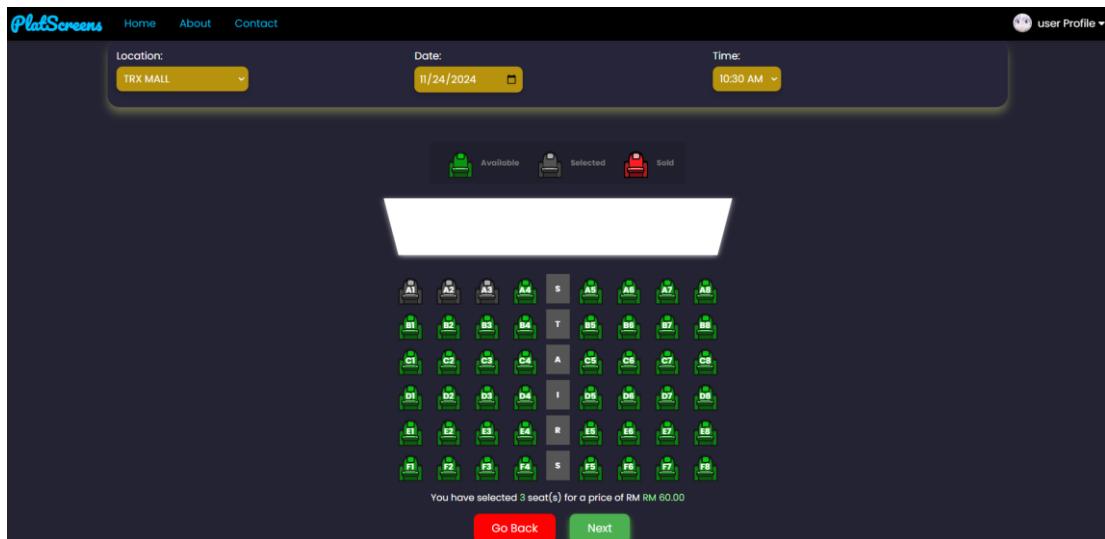


Figure 25: Seat selection in movie booking platform

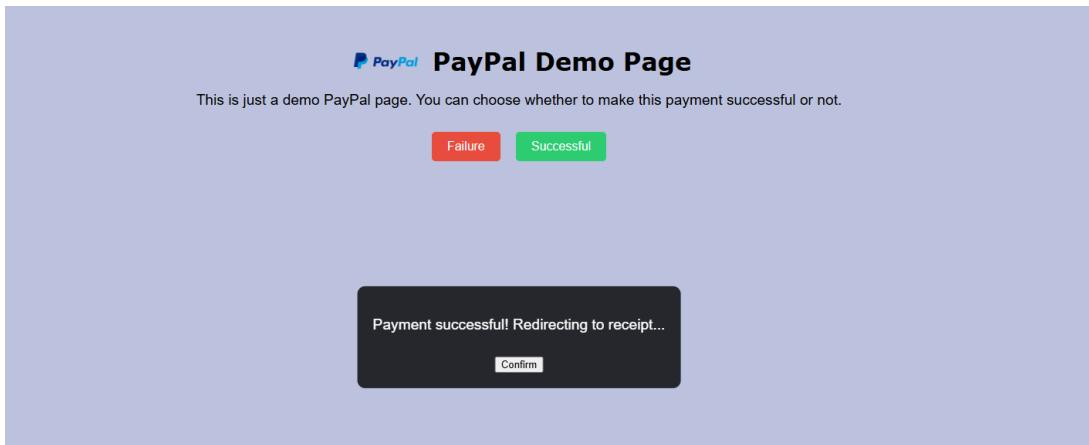


Figure 26:payment page

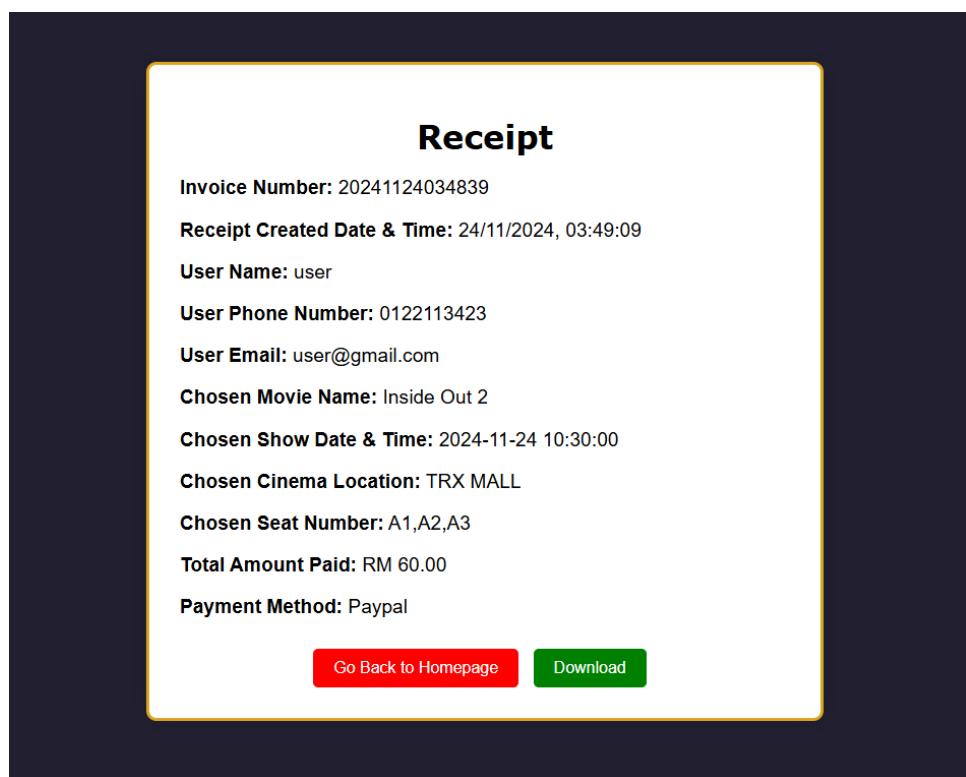


Figure 27:Confirmation of the payment with receipt

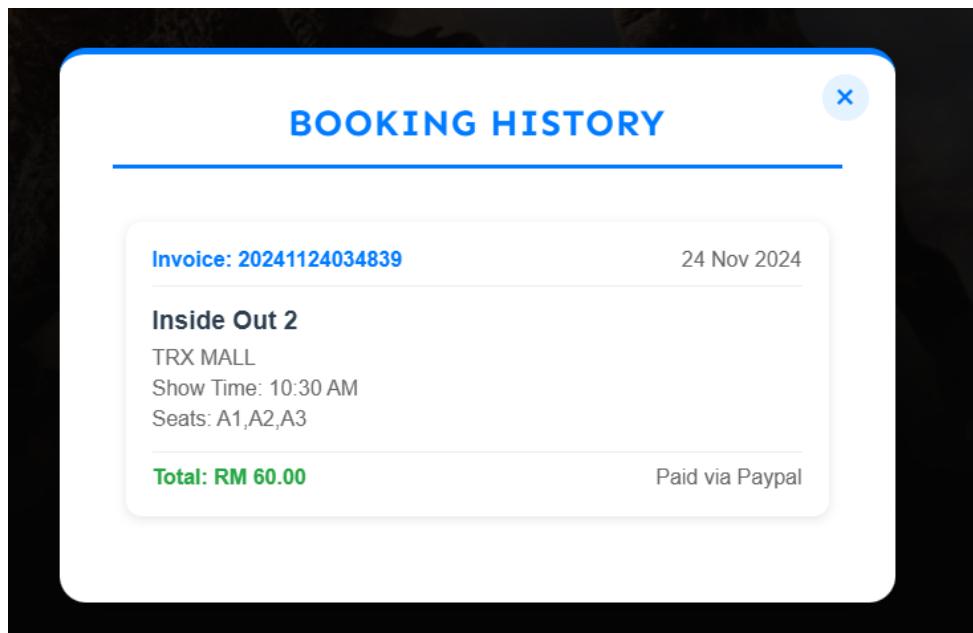


Figure 28:details of booking history

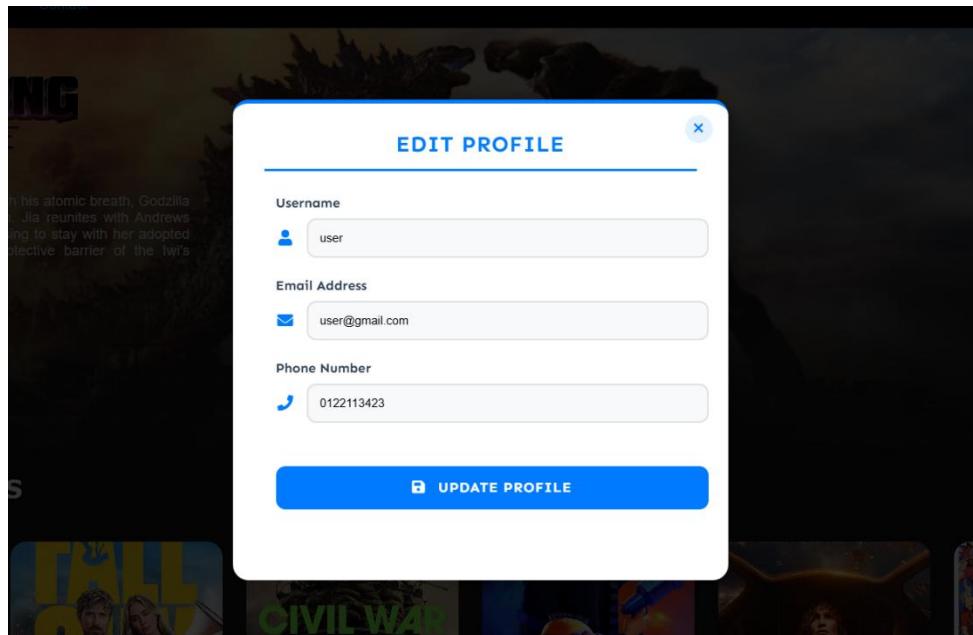


Figure 29:Editing user profile

- In the admin dashboard, ensure that you are allowed to see all the booking information made by customers, add new movie, update movie posters and information, update movie show dates/end dates, and remove movie from database.

| Administrator |    |                |                    |                |                       |              |                       |              |              |   |                             |
|---------------|----|----------------|--------------------|----------------|-----------------------|--------------|-----------------------|--------------|--------------|---|-----------------------------|
|               | #  | Date           | Customer Name      | Contact        | Email                 | Show Date    | Show Time             | Seat         | Total Amount | Payment Method                                  | Action                      |
|               | 3  | 20240712113222 | THOMASTAN ZHENG YE | 01139555279    | thomas89@gmail.com    | Inside Out 2 | ONE UTAMA MALL        | Jul 12, 2024 | 03:30 PM     | A3  | RM 20.00 Touch Go           |
|               | 4  | 20240712121947 | Wilson             | 018223343633   | wilson8@gmail.com     | Inside Out 2 | ONE UTAMA MALL        | Jul 12, 2024 | 03:30 PM     | A5  | RM 20.00 Online Bank Trans  |
|               | 5  | 20240712124712 | Emma Watson        | 01092123768    | emma56@gmail.com      | Fall Guy     | PAVILLION BUKIT JALIL | Jul 13, 2024 | 10:30 AM     | C3  | RM 20.00 Online Bank Trans  |
|               | 6  | 20240714201322 | Helen              | 99999999999999 | helen@GMAIL.COM       | Fall Guy     | TRX MALL              | Jul 14, 2024 | 10:30 AM     | A2  | RM 20.00 Paypa              |
|               | 7  | 20240714201435 | kevin ng           | 013394355279   | kevin@GMAIL.COM       | Fall Guy     | TRX MALL              | Jul 14, 2024 | 10:30 AM     | C4,C5,C6,D5,D6                                  | RM 100.00 Online Bank Trans |
|               | 8  | 20240714204942 | Taylor Swift       | 0184672999     | taylor56@GMAIL.COM    | Fall Guy     | SURIA KLCC            | Jul 14, 2024 | 10:30 AM     | A1,B1,B2  | RM 60.00 Online Bank Trans  |
|               | 9  | 20240714205157 | Johannes Vermeer   | 011111134234   | johannes509@gmail.com | Fall Guy     | PAVILLION BUKIT JALIL | Jul 15, 2024 | 10:30 AM     | C5,C6,C7,C8,D5,D6,D7,D8                         | RM 160.00 Online Bank Trans |
|               | 10 | 20240714205530 | Daniel Craig       | 01932343443    | Daniel125@yahoo.com   | Inside Out 2 | TRX MALL              | Jul 16, 2024 | 10:30 AM     | A3,A4,A5,A6,B3,B4,B5,B6,C3,C4,C5,C6,D3,D4,D5,D6 | RM 320.00 Touch Go          |
|               | 11 | 20241124034839 | user               | 0122113423     | user@gmail.com        | Inside Out 2 | TRX MALL              | Nov 24, 2024 | 10:30 AM     | A1,A2,A3  | RM 60.00 Paypa              |

Figure 30:overview of admin dashboard

| Administrator   |       |                    |         |        |  |
|-----------------|-------|--------------------|---------|--------|--|
| Booking History |       |                    |         |        |  |
| Movie List      |       |                    |         |        |  |
| New Movie       |       |                    |         |        |  |
| #               | Cover | Title              | Status  | Action |  |
| 1               |       | The Garfield Movie | Pending | Action |  |
| 2               |       | Inside Out 2       | Showing | Action |  |
| 3               |       | Fall Guy           | Showing | Action |  |
| 4               |       | IF                 | Ended   | Action |  |
| 5               |       | Civil War          | Showing | Action |  |
| 6               |       | Despicable Me 4    | Showing | Action |  |
| 7               |       | Atlas              | Showing | Action |  |

Figure 31:booking history in admin dashboard

Figure 32: Admin uploading new movies

| # | Cover | Title              | Status  | Action                  |
|---|-------|--------------------|---------|-------------------------|
| 1 |       | The Garfield Movie | Showing | <button>Action</button> |
| 2 |       | Inside Out 2       | Showing | <button>Action</button> |

Figure 33: Movie being updated in the list

## **Phrase 2: Decoupling the application components**

### **Task 1: Creating and configuring the Amazon RDS database**

- 1) At the top of the AWS Management Console, in the search bar, search for and choose EC2
- 2) In the navigation pane, choose **Security Groups**.
- 3) Choose **Create security group**, and configure the following:
  - **Security group name:** Enter “CinemaWebDBSG”
  - **VPC:** Choose **CinemaWebVPC**
  - In the **Inbound rules** section, choose **Add rule** and configure the following:
    - **Type:** Choose **MYSQL/Aurora**.
    - **Source:** Enter 10.0.0.0/16 in the field to the right of **Custom**.
  - Choose **Create security group**.

**Create security group** Info

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

**Basic details**

**Security group name** Info  
CinemaWebDBSG  
Name cannot be edited after creation.

**Description** Info  
Allows SSH access to developers

**VPC** Info  
vpc-0b2fb2886d4ca7043 (CinemaWebVPC)

**Inbound rules** Info

**Inbound rule 1** Delete

| Type         | Protocol | Port range |
|--------------|----------|------------|
| MySQL/Aurora | TCP      | 3306       |

**Source type** Info  
Custom

**Source** Info  
10.0.0.0/16 X

**Description - optional** Info

**Add rule**

Figure 34: create security group for database

- 4) At the top of the AWS Management Console, in the search bar, search for and choose RDS, then choose “Subnet groups”
- 5) In the subnet groups page, click “create DB subnet group”

- 6) Give “CinemaRDS\_SubnetGroup” as the name. Then, choose CinemaWebVPC as the VPC choice
- 7) Add 2 availability zones: ap-southeast-1a” & “ap-southeast-1b
- 8) Select subnets: Private subnet 3 & Private subnet 4.
- 9) Then click “Create”
- 10) At the left navigation pane on the RDS dashboard, choose Databases, then click “Create database”.
- 11) In the **Engine options** section, configure the following:
  - **Engine type:** Choose **MariaDB**.
  - In the **Templates** section, choose **Production**.
  - In the **Settings** section, configure the following:
    - **DB instance identifier:** Enter “CinemaWebDB”
    - **Master username:** Enter “admin”
    - **Master password:** click **auto generate password** (remember to save)
  - In the **Instance configuration** section, for **DB instance class**, choose “db.t4g.large”
  - In the **Storage** section, for **Allocated storage**, enter 20 GiB.
  - In the **Connectivity** section, configure the following:
    - **Virtual private cloud (VPC):** Choose **CinemaWebVPC**.
    - **Public access:** Choose **No**.
    - **VPC security group:** Keep **Choose existing** selected.
    - **Existing VPC security groups:** Remove the **default** group. Add the **CinemaWebDBSG** group.
    - **Availability Zone:** Choose “ap-southeast-1a” & “ap-southeast-1b”
    - **Subnet group:** choose CinemaRDS\_SubnetGroup
  - **Additional configuration** section, and for **Initial database name**, enter “theater\_db”
  - Choose **Create database**.
- 12) When the database is available, find and record the RDS database endpoint:
  - From the list of databases, choose the **DB identifier** link for **CinemaWebDB**.
  - On the **Connectivity & security** tab, under **Endpoint & port**, record the **Endpoint** value to use later.

## Task 2: Creating and configuring the S3 Bucket

- 1) At the top of the AWS Management Console, in the search bar, search for and choose IAM.
- 2) Modify role setting:
  - Choose **Role** on left navigation panel
  - Choose “LabRole”
  - Click “add permissions” > "attach policy"
  - Search for **AmazonS3FullAccess** and check it
  - click “attach policy”.

The screenshot shows the IAM Permissions policies page with 8 items listed. The columns are Policy name, Type, and Attached entities. Policies include AmazonEC2ContainerRegistryReadOnly, AmazonEKSClusterPolicy, AmazonEKSWorkerNodePolicy, AmazonS3FullAccess, AmazonSSMManagedInstanceCore, and three customer-managed policies starting with c131286a3330732l7411243t1w281288598678... .

| Policy name   | Type             | Attached entities |
|---|------------------|-------------------|
| AmazonEC2ContainerRegistryReadOnly                        | AWS managed      | 1                 |
| AmazonEKSClusterPolicy                                    | AWS managed      | 1                 |
| AmazonEKSWorkerNodePolicy                                 | AWS managed      | 1                 |
| AmazonS3FullAccess  | AWS managed      | 1                 |
| AmazonSSMManagedInstanceCore                              | AWS managed      | 1                 |
| <a href="#">c131286a3330732l7411243t1w281288598678...</a> | Customer managed | 1                 |
| <a href="#">c131286a3330732l7411243t1w281288598678...</a> | Customer managed | 1                 |
| <a href="#">c131286a3330732l7411243t1w281288598678...</a> | Customer managed | 1                 |

Figure 35: IAM Permission policies

- 3) At the top of the AWS Management Console, in the search bar, search for and choose S3.
- 4) Create S3 Bucket:
  - Choose **Create bucket**
  - **Bucket name:** “Cinema-Bucket”
  - **Access Setting:** uncheck the “block all public access”
  - Choose **Create bucket**
- 5) Choose bucket name we just created “Cinema-Bucket” and upload “images” folder from the zip file which include all images of the website to the bucket

The screenshot shows the S3 Cinema-Bucket objects page with one item: images/. The object is a folder.

| Name    | Type   | Last modified | Size | Storage class |
|---------|--------|---------------|------|---------------|
| images/ | Folder | -             | -    | -             |

Figure 36: uploading images into the S3 bucket

- 6) Copy S3 bucket URL for later use.

### Task 3: Update Secrets Manager and Migrating the database

- 1) Connect to EC2 instance, type following command:

```
# update variable at secretsmanager
aws secretsmanager update-secret --name "/cafe/dbUrl" --secret-string "<rds database endpoint url>"
aws secretsmanager update-secret --name "/cafe/dbName" --secret-string "theater_db"
aws secretsmanager update-secret --name "/cafe/dbUser" --secret-string "admin"
aws secretsmanager update-secret --name "/cafe/dbPassword" --secret-string "<password that set in aws panel>"
aws secretsmanager create-secret --name "/cafe/s3Url" --secret-string "<your s3 bucket url>

# rds database setup
cd /var/www/html
sudo mariadb -h <rds database endpoint url> -u admin -p < theater_db.sql # use the password that you set in aws panel
```

Figure 37:Connecting to Secret managers

- 2) After you have run the command, you can test the movie booking website.
- 3) Now, we can stop and remove the local database on original EC2 instance.

```
# stop the service
sudo systemctl disable mariadb
sudo systemctl stop mariadb

# removing the package
sudo yum remove mariadb105-server
```

Figure 38: Stop and removal of local database from original ec2

### Task 4: Testing the application

#### User Web Page:

Registration & Login: Test creating a new user account and logging in successfully.

Movie Information: Verify that you can see the most up-to-date movie information, including titles, descriptions, posters, and showtimes.

Booking: Try booking a movie ticket. Ensure the process is smooth and that you receive confirmation.

Booking History: Check your booking history to confirm that past bookings are displayed correctly.

User Profile Management: Access your profile, update your details, and change your password.

## Admin Dashboard:

Booking Management: View all booking information made by customers.

Movie Management: Add new movies to the database.

Updating Movie Posters & Information: Modify movie posters and descriptions.

Updating Show Dates: Adjust the start and end dates for movie screenings.

Removing Movies: Delete movies from the database.

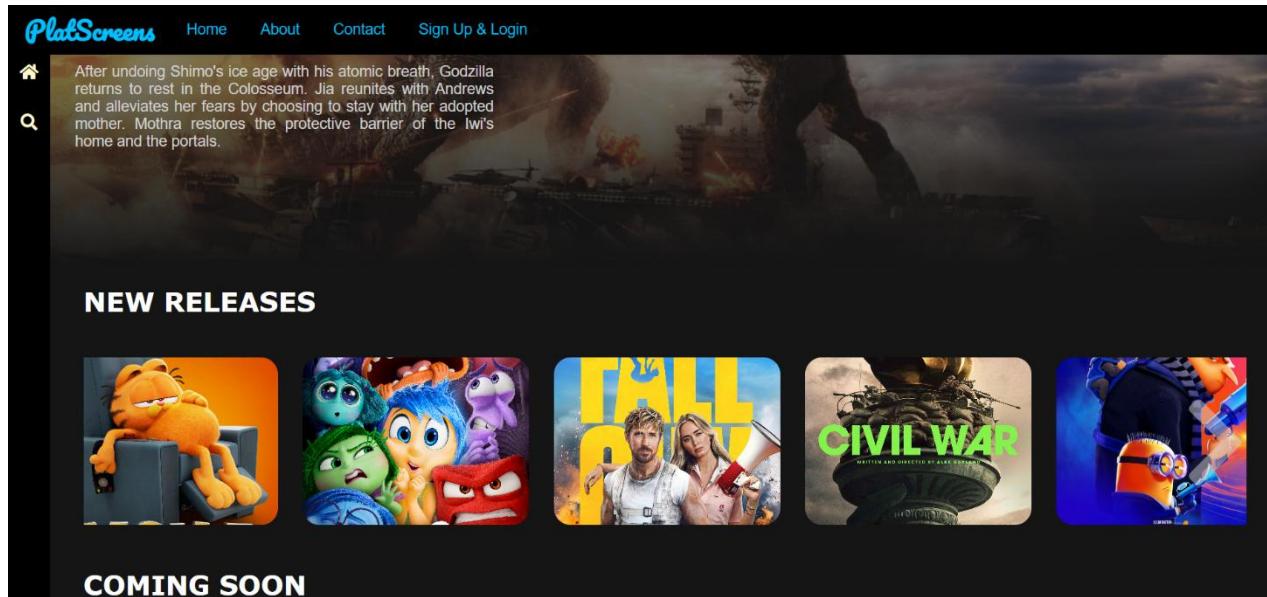
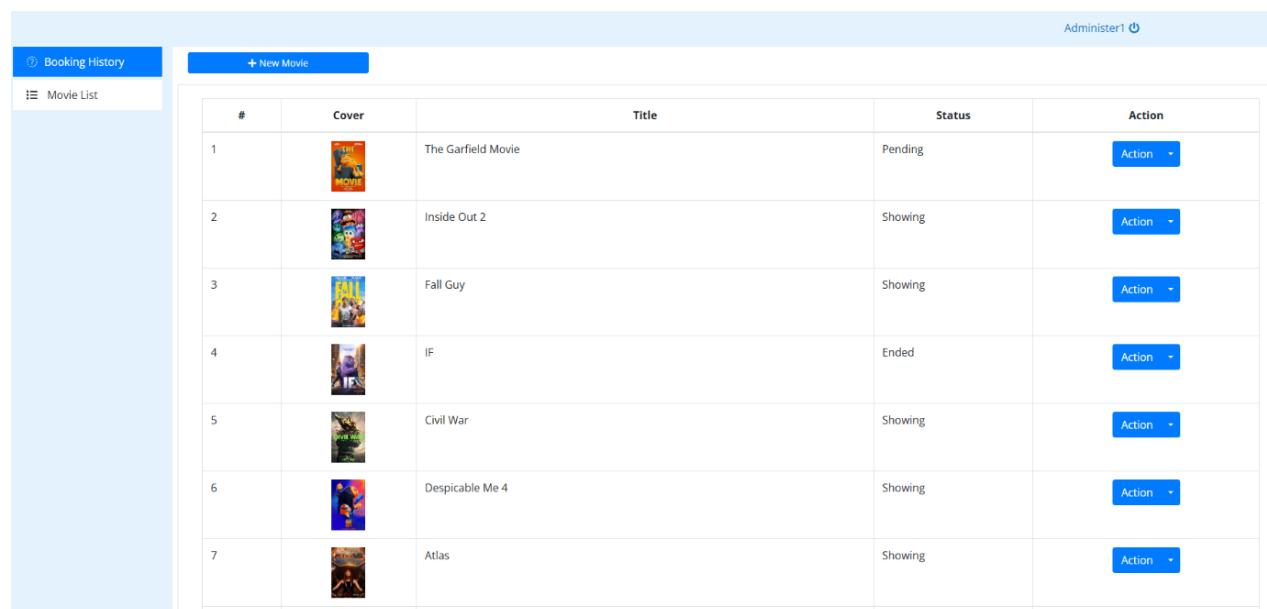


Figure 40: Homepage of movie booking platform

The image shows a screenshot of a movie booking application interface. On the left, there is a sidebar with options for 'Booking History' (selected), 'Movie List' (disabled), and '+ New Movie'. The main area is a table titled 'Movie List' with columns for '#', 'Cover', 'Title', 'Status', and 'Action'. The table contains the following data:

| # | Cover | Title              | Status  | Action |
|---|-------|--------------------|---------|--------|
| 1 |       | The Garfield Movie | Pending | Action |
| 2 |       | Inside Out 2       | Showing | Action |
| 3 |       | Fall Guy           | Showing | Action |
| 4 |       | IF                 | Ended   | Action |
| 5 |       | Civil War          | Showing | Action |
| 6 |       | Despicable Me 4    | Showing | Action |
| 7 |       | Atlas              | Showing | Action |

Figure 39: Browsing of movie history

## **Phase 3: Implementing high availability and scalability**

### **Task 1: Creating an Application Load Balancer**

- 1) At the top of the AWS Management Console, in the search bar, search for and choose EC2
- 2) In the navigation pane, choose **Load Balancers**.
- 3) Choose **Create load balancer**, and configure the following:
  - In the **Load balancer types** section, for **Application Load Balancer**, choose **Create**.
  - **Load balancer name:** enter “CinemaWebALB”
  - In the **Network mapping** section, configure the following:
    - **VPC:** Choose **CinemaWebVPC**.
    - **Mappings:** Select both Availability Zones. The **Subnet** dropdown menu appears when an Availability Zone is selected.
    - **Subnet:** For each Availability Zone, select the public subnet 1 and public subnet 2.
  - In the **Security groups** section, configure the following:
    - Choose **Create security group**.
    - **Security group name:** Enter “CinemaWeb-SG”
    - Choose **Add security group rule**.
    - **rule 1:** For **Type**, choose **HTTP**. For **Source type**, choose **Anywhere**.
    - **rule 2:** For **Type**, choose **SSH**. For **Source type**, choose **10.0.0.0/16**.
    - **rule 3:** For **Type**, choose **MYSQL/Aurora**. For **Source type**, choose **10.0.0.0/16**.
  - In the **Listeners and routing** section, choose **Create target group**, which is located below the **Default action** field. A new tab or window opens.
    - In the new tab or window, configure the following:
      - **Choose a target type:** Choose **Instances**.
      - **Target group name:** Enter CinemaWebTG
      - **VPC:** Choose **CinemaWebVPC**.
      - **Health check path:** “/homepage/index.php”
      - Keep all other default settings and choose **Next**.
      - At the bottom of the page, choose **Create target group**.

- Return to the tab or window where you started to create the load balancer.
- In the **Listeners and routing** section, choose the refresh icon, and then choose **CinemaWebTG** for **Default action**.
- Choose **Create load balancer**.
- Choose **View load balancer**.
- Record the **DNS name** for the load balancer. This name will be used as an endpoint to access the website later.

### Basic configuration

Settings in this section can't be changed after the target group is created.

#### Choose a target type

Instances

- Supports load balancing to instances within a specific VPC.
- Facilitates the use of [Amazon EC2 Auto Scaling](#) to manage and scale your EC2 capacity.

IP addresses

- Supports load balancing to VPC and on-premises resources.
- Facilitates routing to multiple IP addresses and network interfaces on the same instance.
- Offers flexibility with microservice based architectures, simplifying inter-application communication.
- Supports IPv6 targets, enabling end-to-end IPv6 communication, and IPv4-to-IPv6 NAT.

Lambda function

- Facilitates routing to a single Lambda function.
- Accessible to Application Load Balancers only.

Application Load Balancer

- Offers the flexibility for a Network Load Balancer to accept and route TCP requests within a specific VPC.
- Facilitates using static IP addresses and PrivateLink with an Application Load Balancer.

#### Target group name

CinemaWebTG

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

#### Protocol : Port

Choose a protocol for your target group that corresponds to the Load Balancer type that will route traffic to it. Some protocols now include anomaly detection for the targets and you can set mitigation options once your target group is created. This choice cannot be changed after creation

HTTP



80

1-65535

Figure 41: create target group for load balancer

**Health checks**

The associated load balancer periodically sends requests, per the settings below, to the registered targets to test their status.

**Health check protocol**

HTTP ▾

**Health check path**

Use the default path of "/" to perform health checks on the root, or specify a custom path if preferred.

/homepage/index.php

Up to 1024 characters allowed.

► Advanced health check settings

Figure 42: setting target group health check

**Create Application Load Balancer** Info

The Application Load Balancer distributes incoming HTTP and HTTPS traffic across multiple targets such as Amazon EC2 instances, microservices, and containers, based on request attributes. When the load balancer receives a connection request, it evaluates the listener rules in priority order to determine which rule to apply, and if applicable, it selects a target from the target group for the rule action.

► How Application Load Balancers work

**Basic configuration**

**Load balancer name**

Name must be unique within your AWS account and can't be changed after the load balancer is created.

CinemaWebALB

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

**Scheme** | Info  
Scheme can't be changed after the load balancer is created.

**Internet-facing**

- Serves internet-facing traffic.
- Has public IP addresses.
- DNS name is publicly resolvable.
- Requires a public subnet.

**Internal**

- Serves internal traffic.
- Has private IP addresses.
- DNS name is not publicly resolvable.
- Compatible with the IPv4 and Dualstack IP address types.

**Load balancer IP address type** | Info

Select the front-end IP address type to assign to the load balancer. The VPC and subnets mapped to this load balancer must include the selected IP address types. Public IPv4 addresses have an additional cost.

**IPv4**

Includes only IPv4 addresses.

**Dualstack**

Includes IPv4 and IPv6 addresses.

**Dualstack without public IPv4**

Includes a public IPv6 address, and private IPv4 and IPv6 addresses. Compatible with **internet-facing** load balancers only.

Figure 43: Load balancer setup 1

**Network mapping** Info

The load balancer routes traffic to targets in the selected subnets, and in accordance with your IP address settings.

**VPC** | Info  
The load balancer will exist and scale within the selected VPC. The selected VPC is also where the load balancer targets must be hosted unless routing to Lambda or on-premises targets, or if using VPC peering. To confirm the VPC for your targets, view [target groups](#). For a new VPC, [create a VPC](#).

CinemaWebVPC  
vpc-0b2fb2886d4c7043  
IPv4 VPC CIDR: 10.0.0.0/16

**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-09beb67a9e8a74437  
IPv4 subnet CIDR: 10.0.1.0/24

**IPv4 address**  
Assigned by AWS

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

Subnet

subnet-0aa413a73c7135aa3  
IPv4 subnet CIDR: 10.0.4.0/24

**IPv4 address**  
Assigned by AWS

Figure 44: Load balancer setup 2

The screenshot shows the AWS Load Balancer configuration page. At the top, under 'Security groups', a dropdown menu is open with the option 'Select up to 5 security groups'. A single item, 'CinemaWebSG' (sg-0d11b6651a4923168), is listed with a VPC: vpc-0b2fb2886d4ca7043. Below this, the 'Listeners and routing' section is shown. It contains a table for a listener named 'Listener HTTP:80'. The 'Protocol' is set to 'HTTP' and the 'Port' is '80'. The 'Default action' is 'Forward to' a target group named 'CinemaWebTG' (Target type: Instance, IPv4). The 'Create target group' button is visible. There is also a section for 'Listener tags - optional' with a 'Add listener tag' button and a note about adding up to 50 more tags. At the bottom of the screen, there is a blue 'Add listener' button.

Figure 45: Load balancer setup 3

- After the load balancer was created, select the “Target group” from the left navigation pane.
- Now, choose “Cinema” and select Health checks at the bottom on the page.
- Select “Edit”.
- Expand the “advanced health check setting” and check the “turn on stickiness” for 1 days. (This allows load balancer to use cookies to ensure that all requests from particular user during a session are sent to the same target instance.)

The screenshot shows the 'Edit target group attributes' page for the 'CinemaTG' target group. At the top, there is a 'Slow start duration' section with a note that it specifies a period where newly registered targets receive a lower number of requests. A dropdown menu shows '0 seconds' with a note that 30-900 seconds or 0 is allowed. Below this is the 'Target selection configuration' section. Under 'Stickiness', the 'Turn on stickiness' checkbox is checked. The 'Stickiness type' is set to 'Load balancer generated cookie'. The 'Stickiness duration' is set to '1 days'. In the 'Cross-zone load balancing' section, it says 'Inherit settings from load balancer attributes' and notes that cross-zone settings are used by default.

## Task 2: Implementing Amazon EC2 Auto Scaling

- 1) Create an Amazon Machine Image (AMI) of the EC2 instance that hosts the web application, and use it to create the launch configuration for automatic scaling:
  - In the navigation pane, choose **Instances**.
  - From the list of instances, select the **CinemaDevServer** instance.
  - Choose **Actions > Image and templates > Create image**.
  - For **Image name**, enter **CinemaWebAMI**
  - Choose **Create image**.
  - In the banner at the top of the page, choose the link for the AMI name. A new tab or window opens and displays a list of AMIs.

### Create image Info

An image (also referred to as an AMI) defines the programs and settings that are applied when you launch an EC2 instance. You can create an image from the configuration of an existing instance.

The screenshot shows a dialog box titled 'Create image'. It has two fields: 'Instance ID' which contains 'i-04174b280511d5003 (CinemaDevServer)' and 'Image name' which contains 'CinemaWebAMI'. Below the 'Image name' field is a note: 'Maximum 127 characters. Can't be modified after creation.'

Figure 46: create AMI

- 2) Create a launch template:
  - In the left navigation pane, under **Instances**, and choose **Launch Templates**.
  - Choose **Create launch template**.
  - In the **Launch template name and description** section, configure the following:
    - **Launch template name:** Enter “CinemaWebTemplate”
    - **Template version description:** “Web Server for Cinema”
    - **Auto Scaling guidance:** Select **Provide guidance to help me set up a template that I can use with EC2 Auto Scaling**.
  - In the **Application and OS Images** section, choose **My AMIs**. Notice that **CinemaWebAMI** is already chosen.
  - In the **Instance type** section, for **Instance type**, choose “t3.2xlarge”.
  - In the **Key pair** section, create new key pair and enter key pair name “WebServerKey”. (remember to save the key pair file)
  - In the **Network settings** section, for **Security groups**, select **CinemaWeb-SG**.
  - Expand the **Advanced details** section.

- For **IAM instance profile**, choose **LabInstanceProfile**.
- Choose **Create launch template**.
- Choose **View launch templates**

### Create launch template

Creating a launch template allows you to create a saved instance configuration that can be reused, shared and launched at a later time. Templates can have multiple versions.

**Launch template name and description**

Launch template name - required

Must be unique to this account. Max 128 chars. No spaces or special characters like '&', '\*', '@'.

Template version description

Max 255 chars

**Auto Scaling guidance** | [Info](#)

Select this if you intend to use this template with EC2 Auto Scaling

Provide guidance to help me set up a template that I can use with EC2 Auto Scaling

Figure 47: Create Launch Template 1

▼ Application and OS Images (Amazon Machine Image) - required [Info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Recents | **My AMIs** | Quick Start

Owned by me |  Shared with me

[Browse more AMIs](#)  
Including AMIs from AWS, Marketplace and the Community

**Amazon Machine Image (AMI)**

|   |
|---|
| CinemaWebAMI<br>ami-04ecf7592d7a5114f<br>2024-11-18T06:44:19.000Z Virtualization: hvm ENA enabled: true Root device type: ebs |
|---|

Figure 48: Create Launch Template 2

▼ Key pair (login) [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name

[Create new key pair](#)

▼ Network settings [Info](#)

Subnet | [Info](#)

[Create new subnet](#)

When you specify a subnet, a network interface is automatically added to your template.

Firewall (security groups) | [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Select existing security group |  Create security group

Security groups | [Info](#)

[Compare security group rules](#)

VPC: vpc-0b2fb2886d4ca7045

► Advanced network configuration

Figure 49: Create Launch Template 3



Figure 50: Create Launch Template 4

3) Create an Auto Scaling group:

- Choose **CinemaWebTemplate**, and then choose **Actions > Create Auto Scaling group**.
- Configure the following settings:
  - **Auto Scaling group name:** Enter “CinemaWebAutoScalingGroup”
  - Choose **Next**.
  - **VPC:** Choose **CinemaWebVPC**.
  - **Availability Zones and subnets:** Select ”Private Subnet 1” & “Private Subnet 2”.
  - Choose **Next**.
  - In the **Load balancing** section, choose **Attach to an existing load balancer**.
  - **Existing load balancer target groups:** Select **CinemaWebTG**.
  - In the **Health checks** section, for **Health check type**, select **ELB**.
  - Choose **Next**.
  - In the **Group size** section, configure the following:
    - **Desired capacity:** Enter 2
    - **Minimum capacity:** Enter 2
    - **Maximum capacity:** Enter 6
  - In the **Scaling policies** section, configure the following:
    - Choose **Target tracking scaling policy**.
    - **Metric type:** Choose **Average CPU utilization**.
    - **Target value:** Enter 50
    - Choose **Next**.
  - On the **Add notifications** page, choose **Next**.
  - Choose **Add tag**, and then configure the following:
    - **Key:** Enter “Name”
    - **Value:** Enter “Auto Scaling Instance”

- Choose **Next**.
- On the **Review** page, choose **Create Auto Scaling group**.

## Choose launch template or configuration Info

Specify a launch template that contains settings common to all EC2 instances that are launched by this Auto Scaling group. If you currently use launch configurations, you might consider migrating to launch templates.

**Name**

**Auto Scaling group name**  
Enter a name to identify the group.

CinemaWebAutoScalingGroup

Must be unique to this account in the current Region and no more than 255 characters.

Figure 51: Create Auto Scaling Group 1

**Network Info**

For most applications, you can use multiple Availability Zones and let EC2 Auto Scaling balance your instances across the zones. The default VPC and default subnets are suitable for getting started quickly.

**VPC**  
Choose the VPC that defines the virtual network for your Auto Scaling group.

vpc-0b2fb2886d4ca7043 (CinemaWebVPC) ▾
 C

10.0.0.0/16

[Create a VPC](#)

**Availability Zones and subnets**  
Define which Availability Zones and subnets your Auto Scaling group can use in the chosen VPC.

Select Availability Zones and subnets ▾
 C

us-east-1a | subnet-07cb6b599fa5eaa55 X
  
 (private-subnet-1)
   
 10.0.2.0/24

us-east-1b | subnet-07019294bee202f07 X
  
 (private-subnet-3)
   
 10.0.5.0/24

[Create a subnet](#)

**Availability Zone distribution - new**  
Auto Scaling automatically balances instances across Availability Zones. If launch failures occur in a zone, select a strategy.

**Balanced best effort**  
If launches fail in one Availability Zone, Auto Scaling will attempt to launch in another healthy Availability Zone.

**Balanced only**  
If launches fail in one Availability Zone, Auto Scaling will continue to attempt to launch in the unhealthy Availability Zone to preserve balanced distribution.

Figure 52: Create Auto Scaling Group 2

**Load balancing** Info

Use the options below to attach your Auto Scaling group to an existing load balancer, or to a new load balancer that you define.

No load balancer  
 Traffic to your Auto Scaling group will not be fronted by a load balancer.

Attach to an existing load balancer  
 Choose from your existing load balancers.

Attach to a new load balancer  
 Quickly create a basic load balancer to attach to your Auto Scaling group.

**Attach to an existing load balancer**

Select the load balancers that you want to attach to your Auto Scaling group.

Choose from your load balancer target groups  
 This option allows you to attach Application, Network, or Gateway Load Balancers.

Choose from Classic Load Balancers

**Existing load balancer target groups**  
 Only instance target groups that belong to the same VPC as your Auto Scaling group are available for selection.

Select target groups ▾

CinemaWebTG | HTTP X C

Application Load Balancer: web-server-load-balancer

Figure 53: Create Auto Scaling Group 3

**Group size** Info

Set the initial size of the Auto Scaling group. After creating the group, you can change its size to meet demand, either manually or by using automatic scaling.

**Desired capacity type**  
 Choose the unit of measurement for the desired capacity value. vCPUs and Memory(GiB) are only supported for mixed instances groups configured with a set of instance attributes.

Units (number of instances) ▾

**Desired capacity**  
 Specify your group size.

2

Figure 54: Create Auto Scaling Group 4

**Scaling** [Info](#)

You can resize your Auto Scaling group manually or automatically to meet changes in demand.

**Scaling limits**  
Set limits on how much your desired capacity can be increased or decreased.

|                                     |  |
|-------------------------------------|--|
| <b>Min desired capacity</b>         | <b>Max desired capacity</b>            |
| 2                                   | 6                                      |
| Equal or less than desired capacity | Equal or greater than desired capacity |

**Automatic scaling - optional**

**Choose whether to use a target tracking policy** [Info](#)  
You can set up other metric-based scaling policies and scheduled scaling after creating your Auto Scaling group.

|  |  |
|--|--|
| <input type="radio"/> No scaling policies<br>Your Auto Scaling group will remain at its initial size and will not dynamically resize to meet demand. | <input checked="" type="radio"/> Target tracking scaling policy<br>Choose a CloudWatch metric and target value and let the scaling policy adjust the desired capacity in proportion to the metric's value. |
|--|--|

**Scaling policy name**

Target Tracking Policy

**Metric type** [Info](#)  
Monitored metric that determines if resource utilization is too low or high. If using EC2 metrics, consider enabling detailed monitoring for better scaling performance.

Average CPU utilization ▾

**Target value**

50

**Instance warmup** [Info](#)  
300 seconds

Disable scale in to create only a scale-out policy

Figure 55: Create Auto Scaling Group 5

### Task 3: Accessing and Load testing the application

- 1) Use the DNS name of the load balancer to access the web application.
- 2) Test the application functionality.
- 3) Run following command in ec2 instance (CinemaDevServer) to load test “ab -k -c 1000 -n 1000000 http://<LoadBalancerDNS>”
- 4) Navigate to the instances list in the Amazon EC2 console. Observe that the number of instances increases based on the load to the application.

### **3.0 Conclusion**

We are so proud to have successfully assisted PLATSCREEN Cinema in hosting their movie booking website on the AWS cloud environment. This implementation enables their customers to seamlessly browse movie information, book tickets and interact with the web servers and RDS database to add, update, remove, create data. The AWS environment that we created ensures a reliable and scalable infrastructure for enhancing user experience on using the website.

Through this project, we gained extensive knowledge of AWS cloud computing. We have learned how to design an architectural diagram that outlines the interactions between different AWS services based on the needs and conditions of the movie booking website. Moreover, We also get to explore different selection of regions and availability zones, different regions and availability zones, and choose the best options based on factors like latency, reliability, performance, and cost. Configuring essential components such as VPCs, subnets, route tables, NAT gateways, and internet gateways further strengthened our understanding of building secure and efficient network architectures.

Furthermore, we also evaluated multiple storage options in S3 and selected the most suitable storage types for the website ‘needs. Similarly, we compared various AWS database solutions to choose an option that can balance functionality, scalability, reliability, and cost considerations. By determining the optimal EC2 instance types, we ensured the deployment met the website’s performance requirements.

Besides, we deployed the PHP app on AWS and separating the web server and database to improve the efficiency and security of the environment. We configured network security settings to establish secure communication channels between the different components of the application.

In our project, we implemented a combination of load balancers and auto-scaling on AWS to ensure the high availability and scalability of our application. We configured load balancers to distribute incoming traffic across multiple instances, preventing any single server from becoming overloaded and maintaining responsiveness even during peak traffic periods. We

also implemented auto-scaling to dynamically adjust the number of instances based on predefined metrics such as CPU utilization and request rate. This allowed our application to handle fluctuations in traffic effectively, scaling up when necessary and scaling down when traffic decreased.

This assignments enriched our experience on AWS cloud computing, and it allows us to navigate the complexities of cloud architecture and deployment while delivering a robust and efficient solution for PLANSCREEN cinema's movie booking system.