



A PROJECT REPORT ON

Designing and Implementing a Scalable Cloud Infrastructure on AWS for Ubuntu Server Administration in a centralized large network



SUBMITTED BY

MR. KARAN VERMA
MR. TUSHAR MORE
MS. NIKITA BACHA
MS. SHRUTI JAGTAP
MS. PRATIKSHA DESHMUKH
MR. NANDKUMAR CHAVAN
MR. VASKAR PAUL

UNDER THE GUIDANCE

MR. ZAKIR HUSSAIN

TABLE OF CONTENTS

SR. NO.	CONTENTS
1	Project Overview
2	Prerequisites
3	Architecture Design
4	Steps and Configuration
5	Conclusion
6	Result

1. PROJECT OVERVIEW

The purpose of this project is to design and implement a scalable cloud infrastructure on Amazon Web Services (AWS) using Ubuntu servers for administration in a centralized large network.

This infrastructure will support the seamless functioning of critical applications

- **Web hosting** (e.g., running an NGINX or Apache server).
- **API services** (e.g., hosting RESTful or GraphQL APIs for internal and external use).
- **Data processing** (e.g., handling analytics, machine learning workflows, or real-time data processing pipelines).
- **Microservices architecture** (breaking down a large application into smaller, independent services).

Goals : The main goals of the project include,

1. **Scalability:** The system should be able to scale horizontally or vertically based on demand, ensuring there are no performance bottlenecks as the user base grows.
2. **Centralized Management:** Implement tools and techniques to centrally monitor, update, and configure all Ubuntu servers, minimizing administrative overhead.
3. **High Availability:** The infrastructure should ensure **99.99% uptime**, leveraging features like multi-AZ deployments and automated failovers.
4. **Cost Efficiency:** Utilize AWS pricing models (e.g., on-demand, reserved instances) and resource optimization techniques to maintain a balance between performance and cost.
5. **Security:** Implement industry best practices for securing infrastructure, including least privilege access, encryption, and regular security audits.



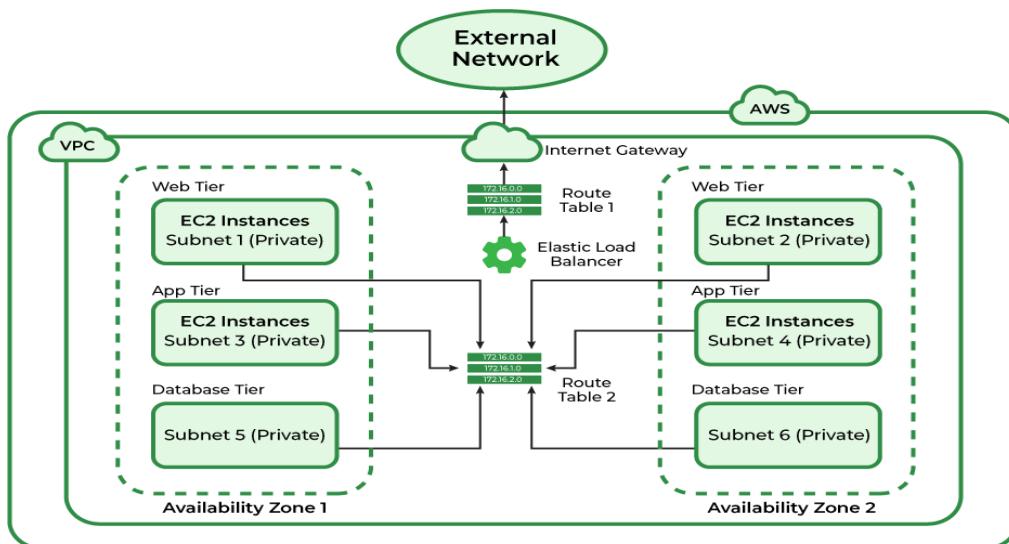
2. PREREQUISITES

Basics knowledge on –

- 1) AWS (Amazon Web Services) - AWS is a comprehensive cloud computing platform provided by Amazon. It offers a range of cloud services that include computing power, storage, databases, networking, machine learning, analytics, security, and application development tools. AWS allows organizations and individuals to build, deploy, and manage applications and services without the need for on-premises infrastructure.



- 2) VPC (Virtual Private Cloud) - Amazon Virtual Private Cloud (Amazon VPC) provides a logically isolated area of the AWS cloud where you can launch AWS resources in a virtual network that you define. You have complete control over your virtual networking environment, including a selection of your IP address range, the creation of subnets, and configuration of route tables and network gateways.



- 3) Amazon EC2 (Elastic Compute Cloud) is a web service that provides scalable, resizable virtual servers (instances) in the cloud to run applications.



- 4) AWS RDS (Relational Database Service) is a managed service that makes it easy to set up, operate, and scale relational databases in the cloud, supporting databases like MySQL, PostgreSQL, Oracle, SQL Server, and Amazon Aurora.



- 5) Elastic Load Balancing (ELB) is an AWS service that automatically distributes incoming traffic across multiple targets, such as EC2 instances, containers, and IP addresses, to ensure high availability and fault tolerance for applications.

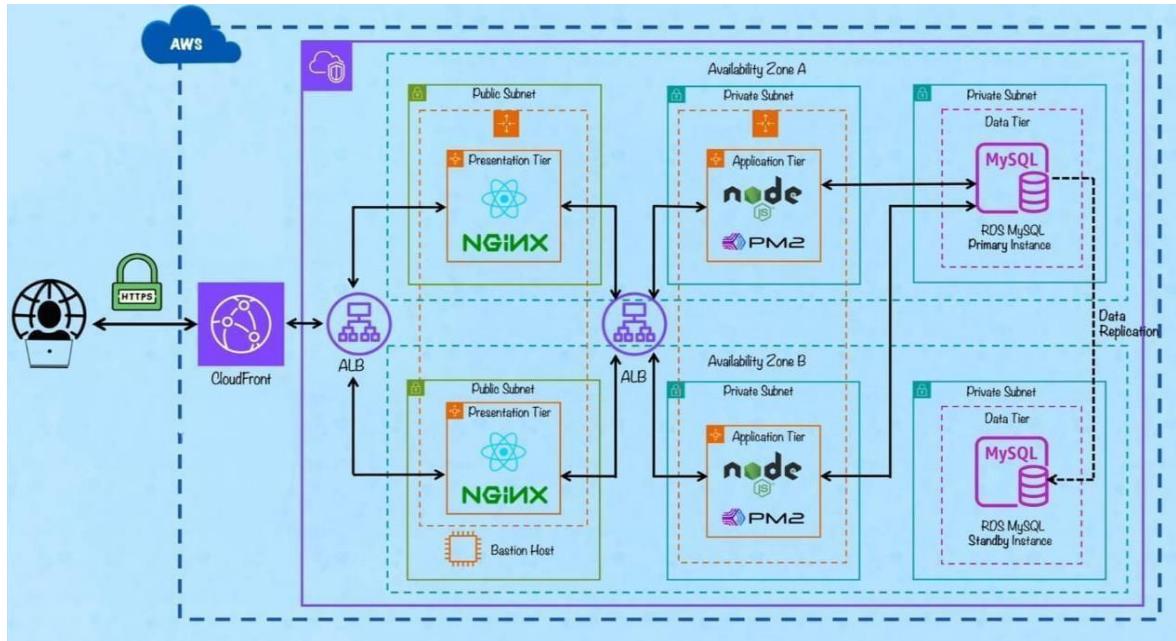


- 6) Amazon CloudFront speeds up distribution of your web content by delivering it through a worldwide network of data centers, which lowers latency and improves performance.



3. ARCHITECTURE DESIGN

This Architecture illustrates How a web application can be hosted on AWS with various services interconnected. It outlines the process from the web client reaching into the AWS cloud, going through security and management layers before reaching server instances and databases. Different AWS services like VPC, CloudFront ,Bastion Host, EC2 Instances, Application Load Balancer and RDS are represented with icons and labels. The architecture is divided into different zones including availability zones, subnets etc., each containing specific AWS resources/services.



This Architecture illustrates a highly available and scalable web application architecture on Amazon Web Services (AWS). Here's a breakdown of the components and data flow:

Components:

- a) CloudFront: A content delivery network (CDN) that caches static content (e.g., images, CSS, JavaScript) closer to end users, improving performance and reducing latency.
- b) Application Load Balancer (ALB): Distributes incoming traffic across multiple instances of the application layer (Presentation Tier and Application Tier) in both Availability Zones, ensuring high availability and load balancing.
- c) Availability Zones (AZs): Separate physical locations within a region that provide fault tolerance and redundancy.
- d) Public Subnet: A subnet within a VPC that allows internet traffic.
- e) Private Subnet: A subnet within a VPC that restricts internet traffic, ensuring security.

- f) Presentation Tier: The layer responsible for serving static content (e.g., HTML, CSS, JavaScript) to users. It uses NGINX as a web server.
- g) Application Tier: The layer that handles dynamic requests and business logic. It uses Node.js and PM2 for process management.
- h) Data Tier: The layer responsible for storing and managing data. It uses MySQL as the database and RDS MySQL for managed database services.
- i) Bastion Host: A server that provides secure access to the private network from the public internet.
- j) Data Replication: Ensures data consistency and redundancy by replicating data across Availability Zones

Data Flow:

This architecture represents a scalable, highly available AWS setup for deploying a web application. The clients (users) access the application through CloudFront (CDN) for fast content delivery. Traffic is routed via a application load balancer, which distributes it across multiple EC2 instances (servers running the application) deployed in separate availability zones within public subnets and private subnet.

Two instances (presentation layer instances) in public subnet in two availability zones used for hosting front end of the web application .

Two instances (application layer instances) in private subnet in two availability zones used for hosting backend end of the web application

The RDS (Relational Database Service) instances (DB-layer instances), located in private subnets , store the web application's data, like book info, author info, and configured in a multi-AZ setup for high availability. Both EC2 and RDS instances are secured with security groups that control traffic access.

Bastion Host: In this architecture, a bastion host is used to provide secure access to the private network from the public internet. It acts as a controlled gateway, allowing authorized users to connect to the private network while preventing unauthorized access.

While NAT gateways can be a useful tool in certain AWS architectures, they are not the best choice for this specific scenario due to security, performance, complexity, and cost considerations. The use of a bastion host provides a more secure, efficient, and cost- effective solution for accessing the private network in this case.

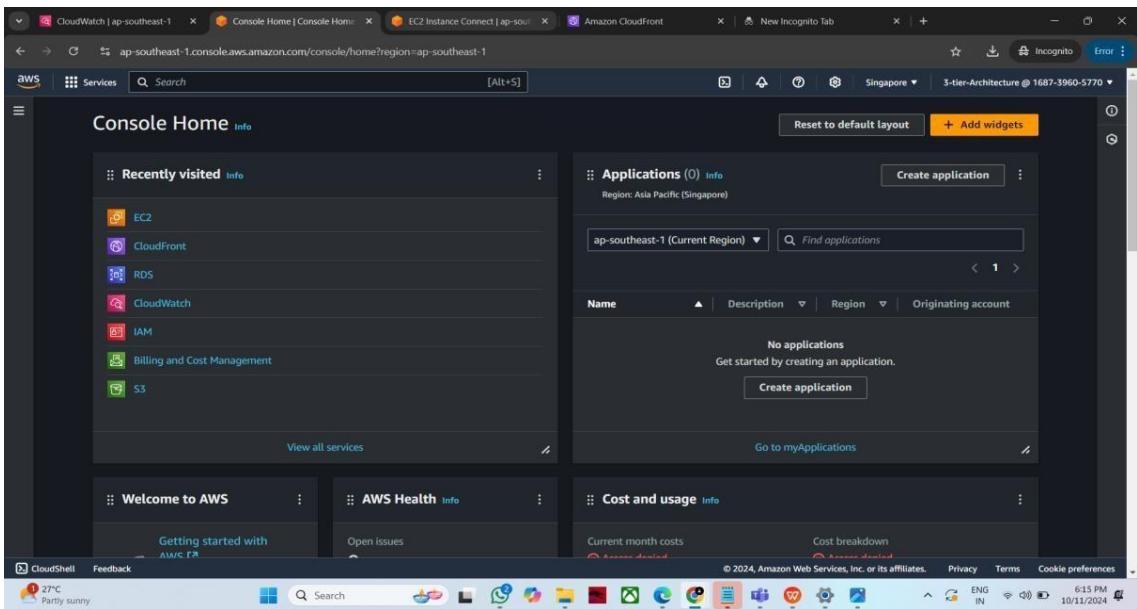
4. STEPS AND CONFIGURATION

Step 1: Login into AWS Console

1.1 Create an account and login into it by providing the credentials

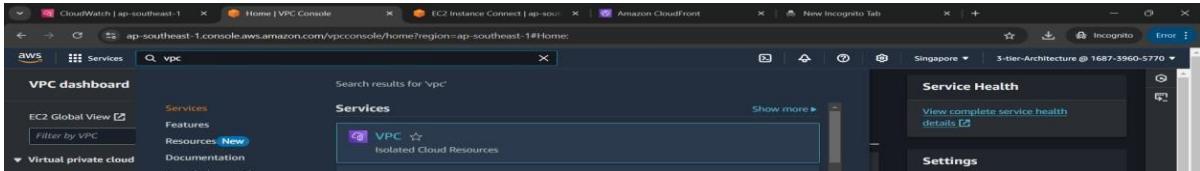
The image shows two screenshots of the AWS console. On the left, the sign-in page for an IAM user is displayed. It includes fields for Account ID, IAM user name (P2_User), and Password, along with a 'Sign In' button and links for root user email and forgot password. On the right, a password change form is shown, requiring the current AWS account (168739605770), IAM user name (P2_User), old password, new password, and retype new password. A 'Confirm password change' button and a link for sign in using root user email are also present.

1.2 Give access to AWS Dashboard



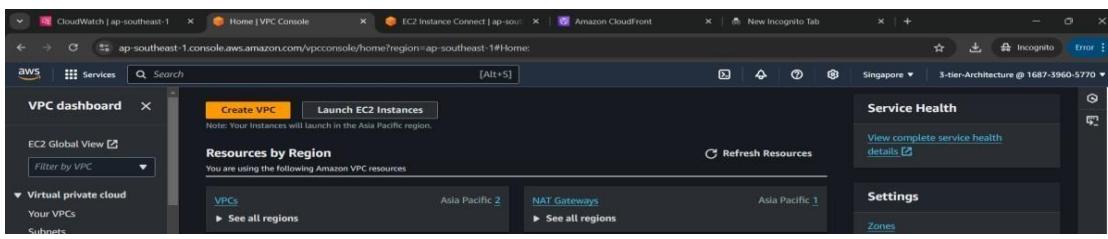
Step 2: Launch VPC in Console

2.1 In Dashboard, search for VPC service, click VPC to open it.



2.2 Click Create VPC to configure.

1. Name Tag: Give your VPC a name .
2. IPv4 CIDR block: Enter an IP range in CIDR notation (e.g., 10.0.0.0/16). This defines the range of private IPs for your VPC.
3. Click Create VPC.



VPC settings

Resources to create [Info](#)
Create only the VPC resource or the VPC and other networking resources.

VPC only **VPC and more**

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

IPv4 CIDR block [Info](#)
 IPv4 CIDR manual input **IPAM-allocated IPv4 CIDR block**

IPv4 CIDR

CIDR block size must be between /16 and /28.

IPv6 CIDR block [Info](#)
 No IPv6 CIDR block **IPAM-allocated IPv6 CIDR block** **Amazon-provided IPv6 CIDR block** **IPv6 CIDR owned by me**

Tenancy [Info](#)

Tags

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

No tags associated with the resource

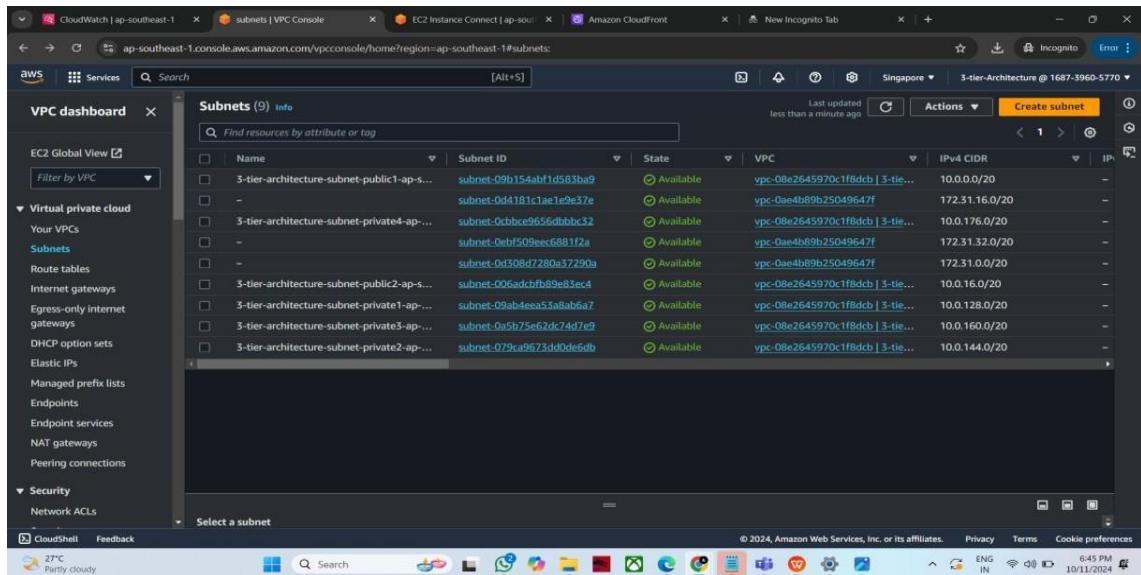
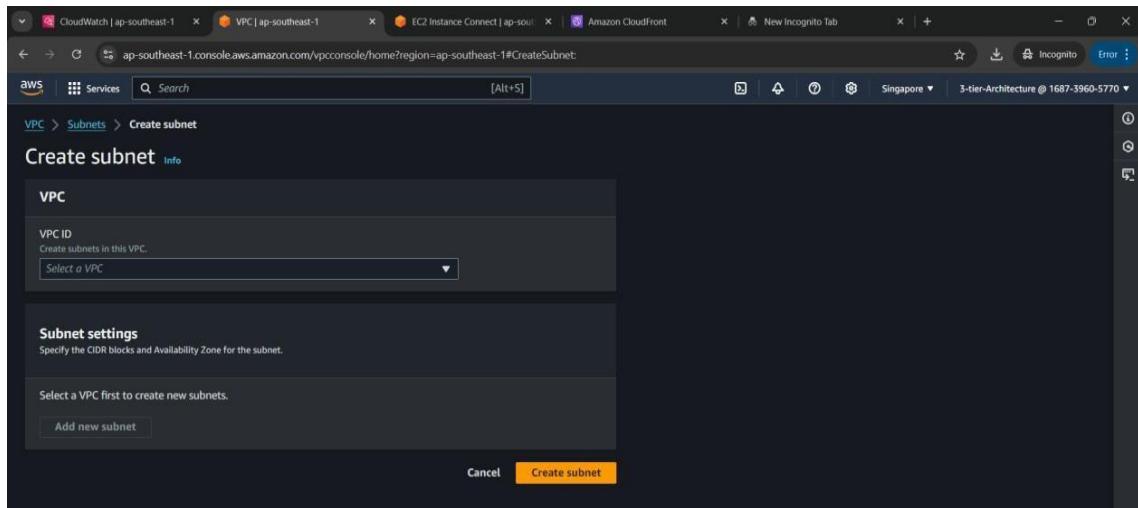
Add tag

You can add 50 more tags

Create VPC

2.3 Now, select the subnets in the left pane and create Subnets

- From the VPC dashboard, select Subnets from the left-hand menu.
- Click Create subnet.
- Select your VPC and specify the Subnet Name.
- Choose an Availability Zone (e.g., us-east-1a) for each subnet.
- Specify a CIDR block for the subnet (e.g., 10.0.1.0/24).
- Repeat the process for additional subnets (e.g., a public and a private subnet in different Availability Zones).



- | | | |
|---|------|---------------|
| 1 : 3-tier-architecture-subnet-public1-ap-southeast-1a , | ipv4 | 10.0.0.0/20 |
| 2 : 3-tier-architecture-subnet-public2-ap-southeast-1b , | ipv4 | 10.0.16.0/20 |
| 3 : 3-tier-architecture-subnet-private1-ap-southeast-1a , | ipv4 | 10.0.128.0/20 |
| 4 : 3-tier-architecture-subnet-private2-ap-southeast-1b , | ipv4 | 10.0.144.0/20 |
| 5 : 3-tier-architecture-subnet-private3-ap-southeast-1a , | ipv4 | 10.0.160.0/20 |
| 6 : 3-tier-architecture-subnet-private4-ap-southeast-1b , | ipv4 | 10.0.176.0/20 |

2.4 Create Route Tables

1. From the VPC dashboard, select Route Tables in the left-hand menu.
2. Create a new route table (or use the default one associated with your VPC).
3. Select Routes and click Edit Routes.
4. Add a route for internet traffic:
 1. Destination: 0.0.0.0/0
 2. Target: Select the Internet Gateway (IGW) you created.
5. Save the changes and associate this route table with the subnets.

The screenshot shows the AWS VPC dashboard with the 'Route tables' section selected. A table lists seven route tables, each with a name, route table ID, explicit subnet association, edge associations, main status, and VPC information. The table includes columns for Name, Route table ID, Explicit subnet assoc..., Edge associations, Main, and VPC. The VPC column shows IDs like vpc-0be2645970c1f8dcba, vpc-0be2645970c1f8dcbb, etc. The 'Actions' button at the top right of the table header has a dropdown menu with options including 'Create route table'.

2.5 Create an Internet Gateway (IGW)

1. From the VPC dashboard, click on Internet Gateways in the left-hand menu.
2. Click Create internet gateway.
3. Name the Internet Gateway (optional).
4. Click Create and then Attach to VPC to link it to the VPC you just created.

The screenshot shows the 'Create internet gateway' wizard. The first step, 'Internet gateway settings', is displayed. It includes fields for a 'Name tag' (containing 'my-internet-gateway') and a 'Tags - optional' section (which is currently empty). At the bottom are 'Cancel' and 'Create internet gateway' buttons.

2.6 Configure Security Groups and Network ACLs

1. Security Groups: Create or modify security groups to allow traffic (e.g., HTTP/HTTPS, SSH).
2. Network ACLs (optional): Configure access control lists for additional subnet-level security.

1. Bastion Host Security Group controls access to the bastion host in VPC, which is used to securely access private instances in a private subnet. It acts as a gateway for administrative access, typically through SSH (for Ubuntu), to resources that do not have direct internet access.
2. Presentation-Tier ALB (Application Load Balancer) Security Group controls the traffic allowed to reach the Application Load Balancer
3. Presentation-Tier EC2 Security Group is responsible for controlling access to the EC2 instances in the presentation tier, which typically host the web servers or frontend of an application . This security group ensures that only authorized traffic, such as HTTP and HTTPS requests, can access the web servers

Connection between Presentation-Tier EC2 and Bastion Host

4. Application-Tier ALB (Application Load Balancer) Security Group is responsible for controlling the traffic that flows into the application-tier of your architecture, typically between the Application Load Balancer and the backend resources (like EC2 instances running your application, web servers, or microservices).
5. **Application-Tier EC2 Security Group** controls traffic to and from the EC2 instances running in the application layer of your architecture, such as your backend servers or services. These instances usually handle the application logic, data processing, or API services. This security group needs to allow traffic from the **Application Load Balancer (ALB)** in the application tier and potentially restrict access from other external sources.

Connection between **Presentation-Tier EC2 and Bastion Host**

6. The **Data-Tier Security Group** is responsible for managing the traffic that flows to and from the database or data storage servers in the data-tier of your architecture. This tier typically includes resources like **Amazon RDS (Relational Database Service)**, **Amazon DynamoDB**, or **self-hosted databases** on EC2 instances. The security group must be configured to only allow necessary traffic, typically from the application tier or specific services that need to interact with the database.

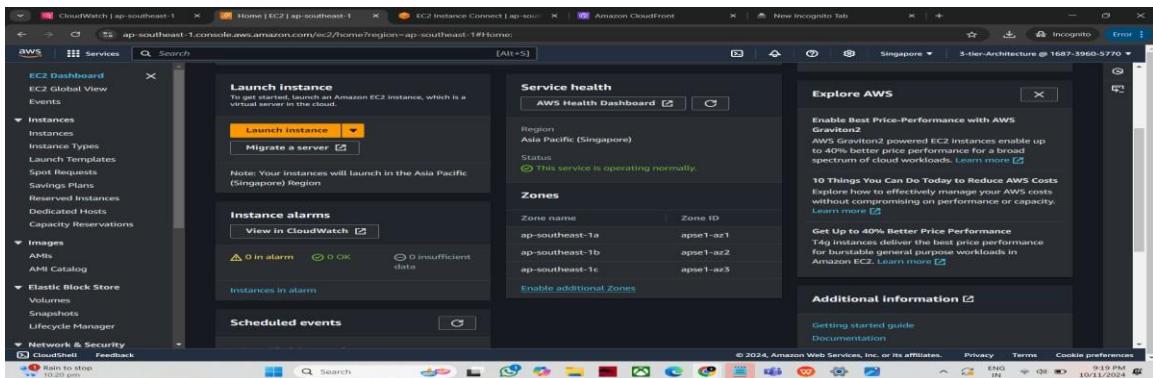
Connection between **Application-Tier EC2 and Data-Tier (MY SQL)** .

Connection between **Bastion Host and Data-Tier (MY SQL)** .

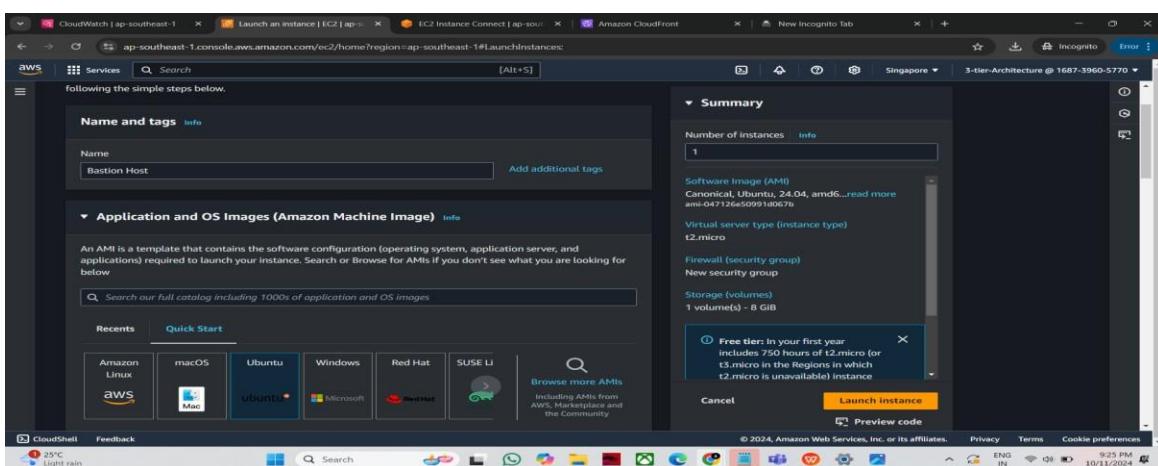
Step 3: Launching The Bastion Host

A Bastion Host is an EC2 instance that acts as a gateway for secure SSH or RDP access to instances in private subnets within your Virtual Private Cloud (VPC). Bastion hosts are typically deployed in public subnets with strict security group rules to restrict access. This allows for secure administrative access without exposing your private servers directly to the internet.

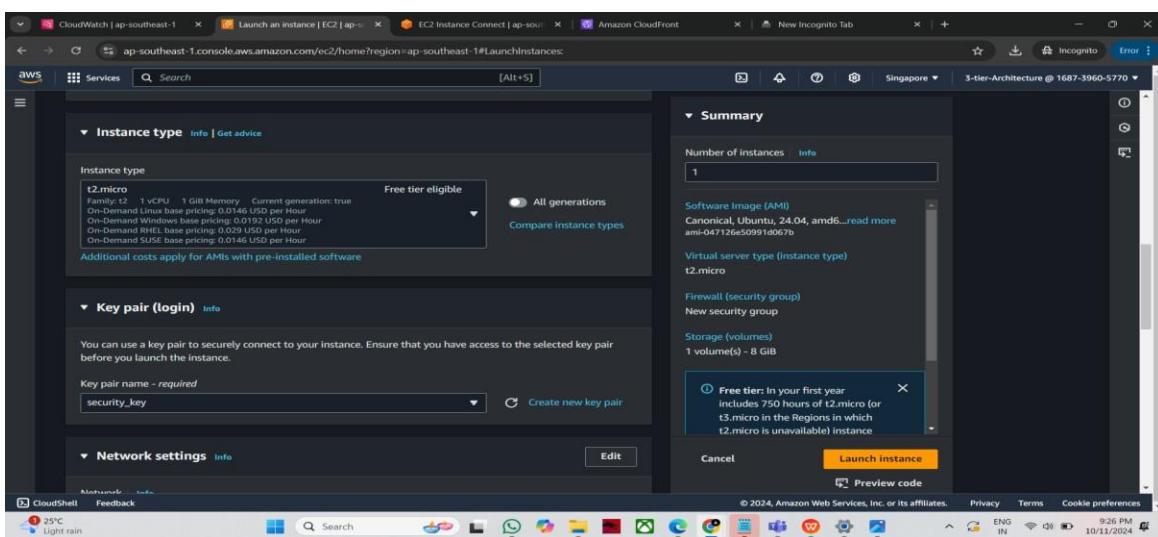
3.1 In Dashboard, search for EC2 service, click EC2 to open it.



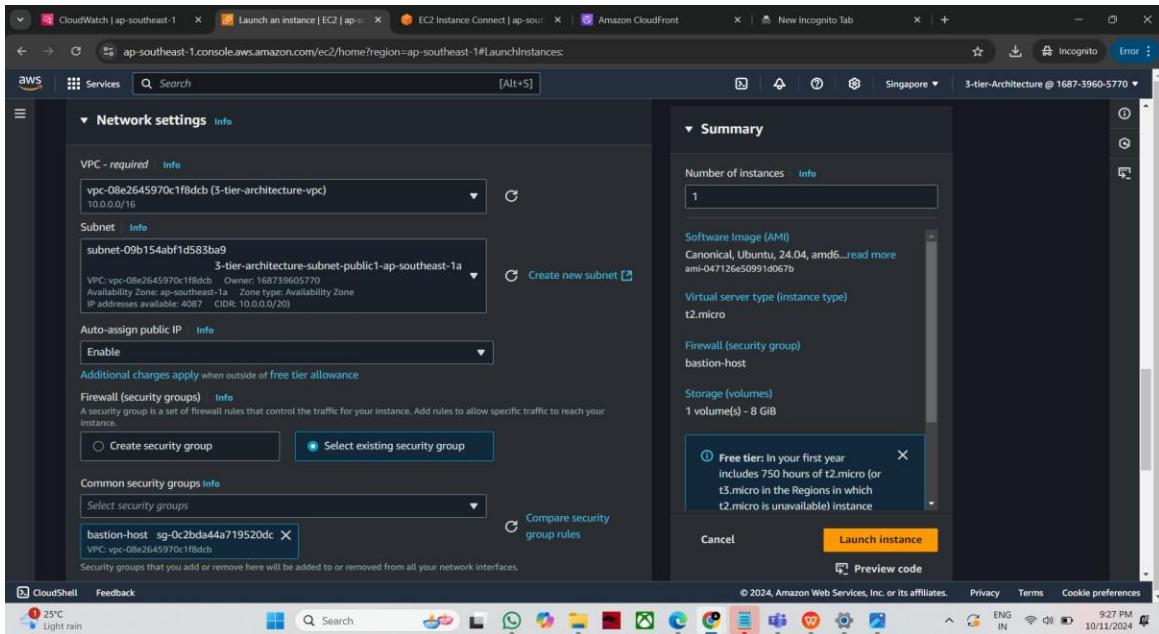
3.2 Name the Instance and Select the application and OS image.



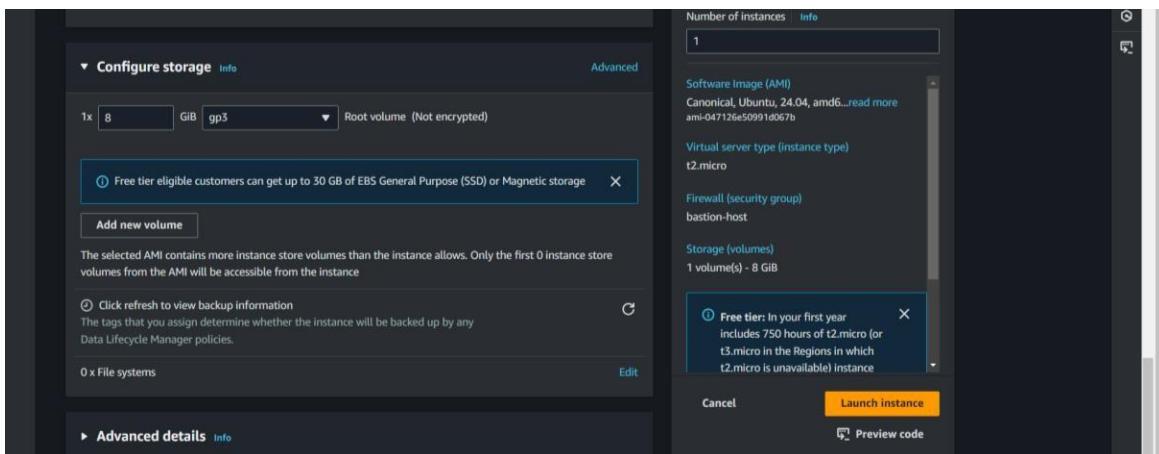
3.3 select Instance Type and Key pair (Security-Key) .



3.4 Network Settings and Select Security Group (Bastion-Host) .



3.5 Launch Instance (Bastion Host) .

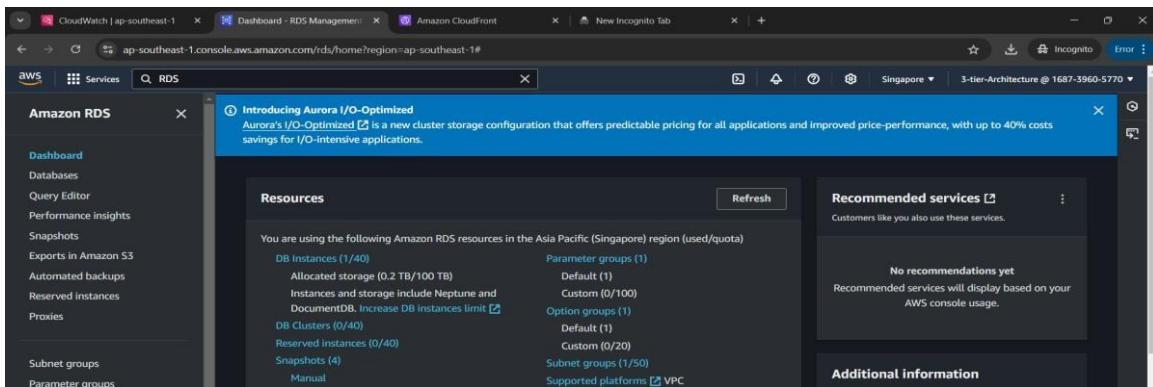


Step 4: Setting up Data Tier with RDS MySQL Instance

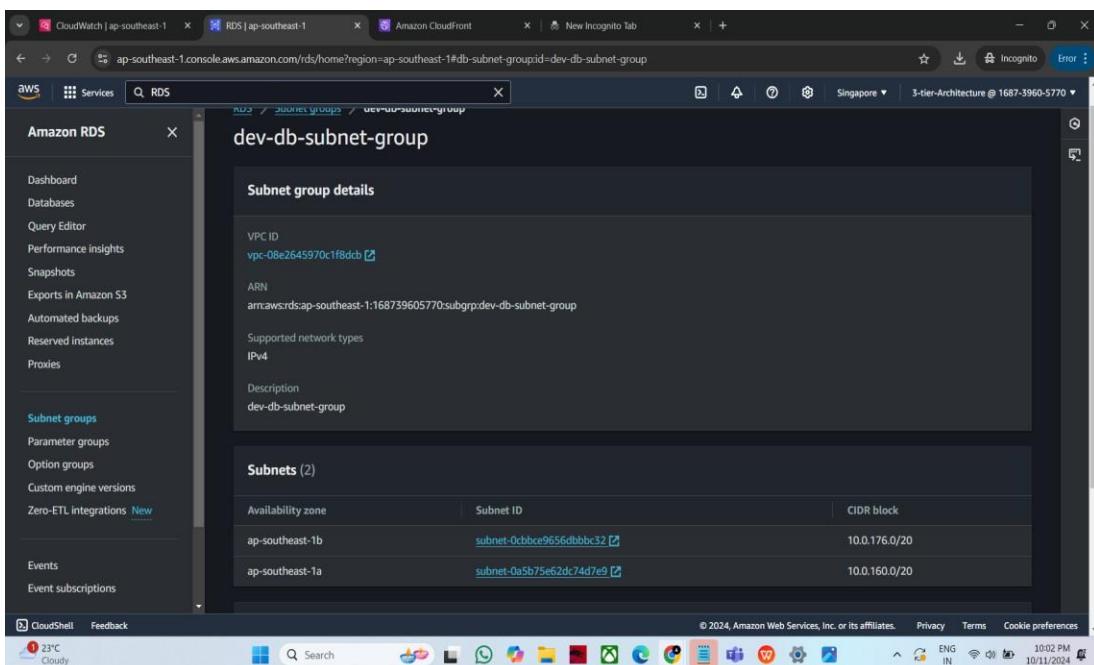
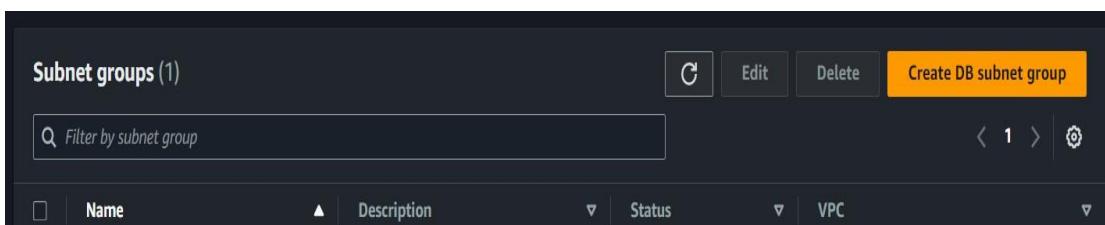
Amazon RDS MySQL is a managed relational database service that allows you to easily set up, operate, and scale MySQL databases in the cloud. It automates time-consuming tasks like backups, patching, and scaling, enabling high availability and durability with features like automated backups, Multi-AZ deployment, and read replicas.

Steps to Set Up the Data Tier with RDS MySQL :

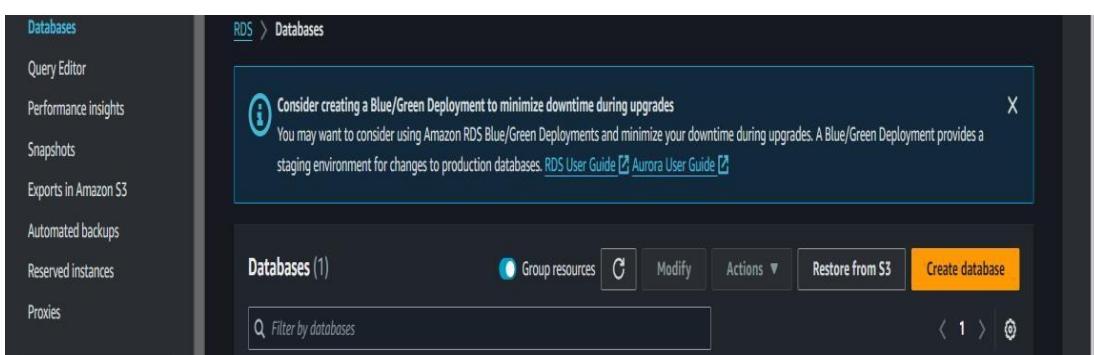
4.1 In Dashboard, search for RDS service, click RDS to open it.



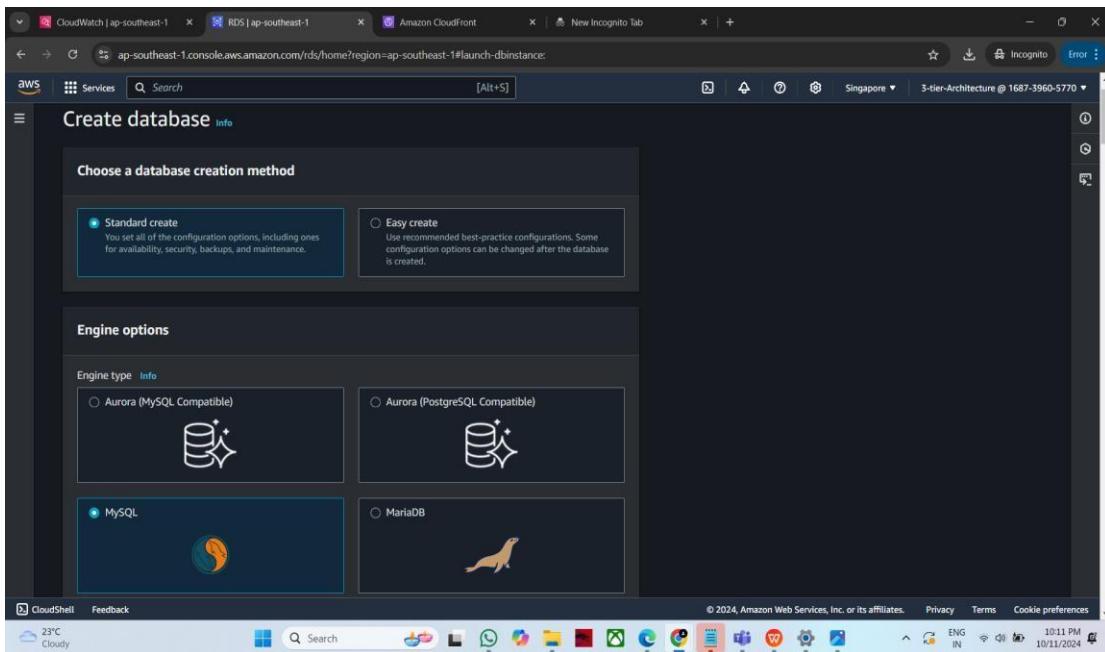
4.2 Click on the Create Subnet Groups button on the RDS dashboard



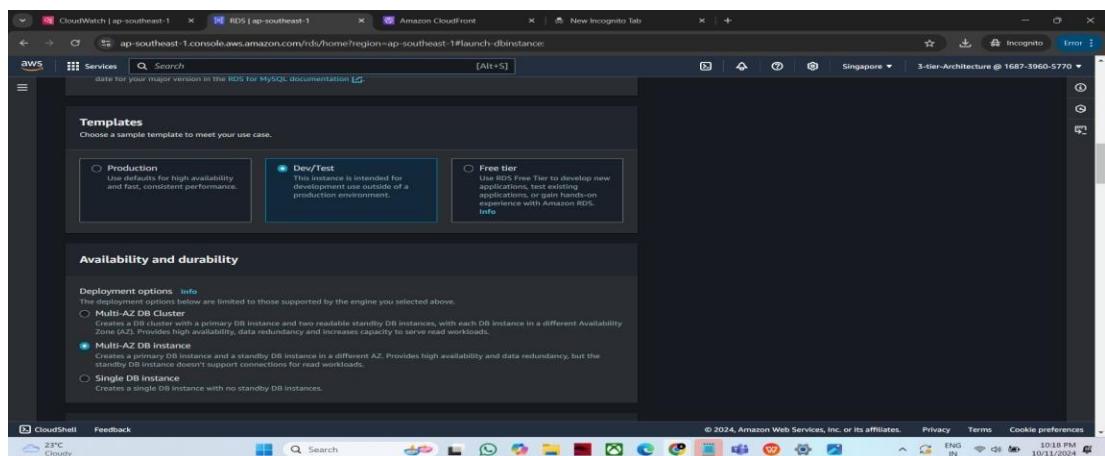
4.3 Click on the Create database button on the RDS dashboard.



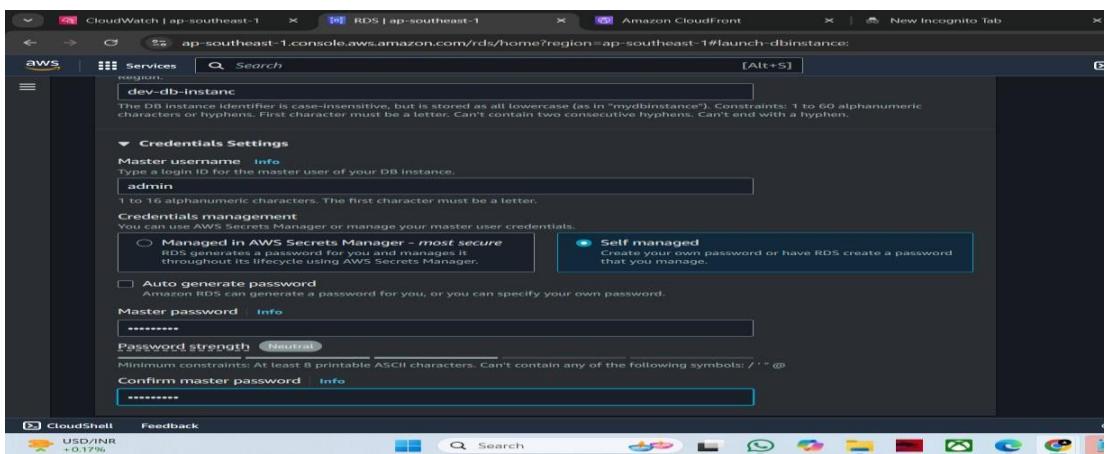
4.3.1 Choose a database creation method and Engine options



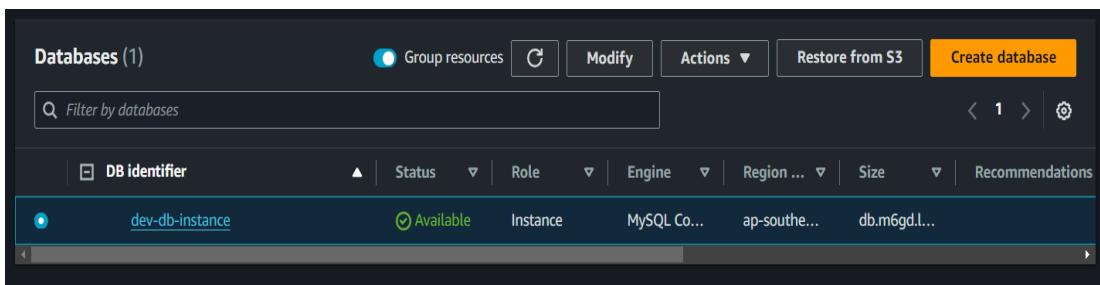
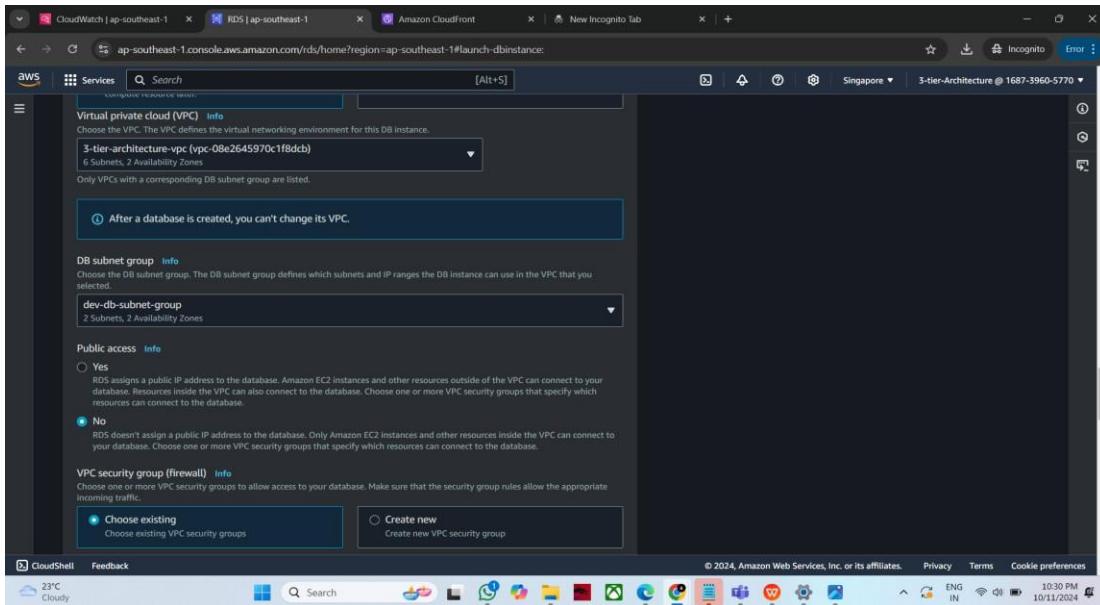
4.3.2 Select Template



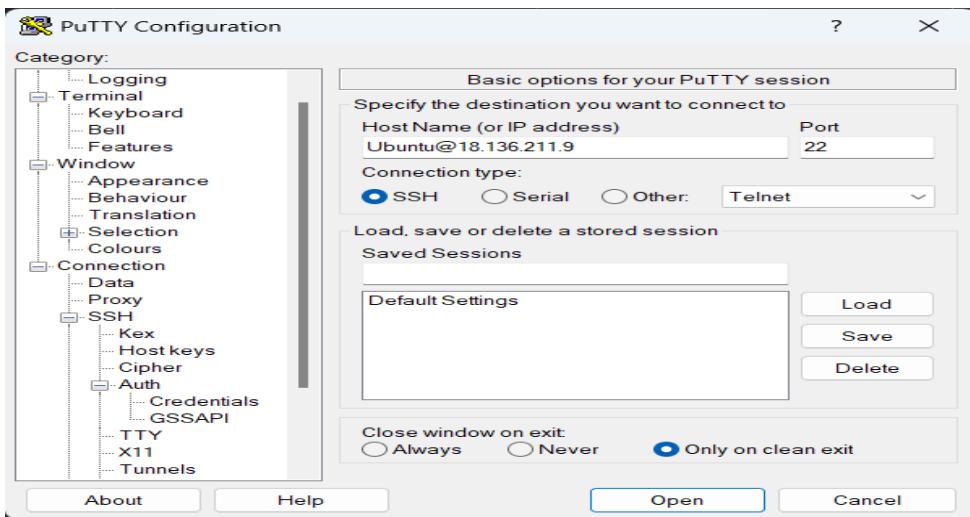
4.3.3 Setting up Db name, Master Username and Master Password.

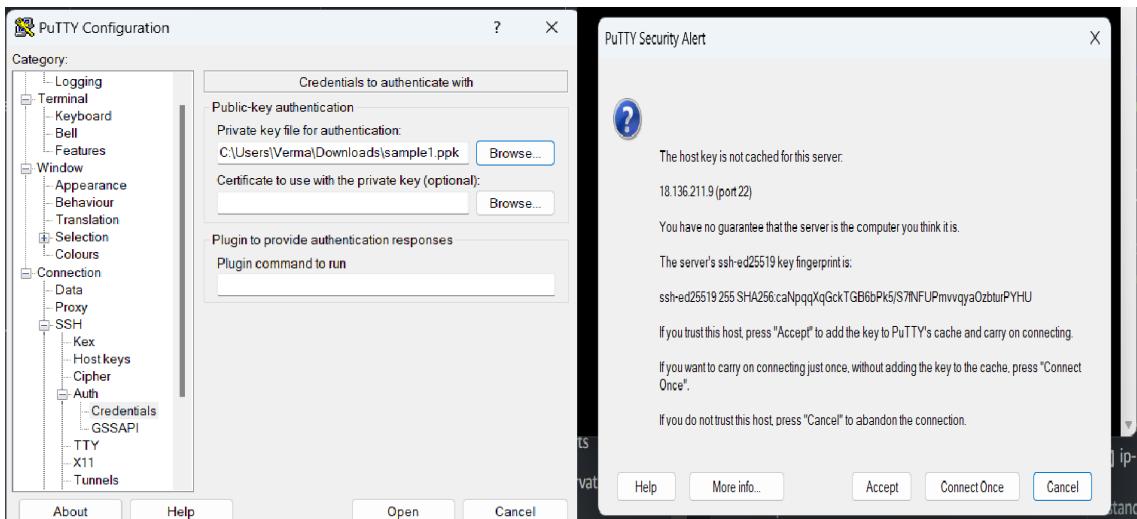


4.3.4 Setting up connectivity .



4.4 Connecting to RDS Instance (using PUTTY)





```
Last login: Fri Oct 11 03:23:10 2024 from 3.0.5.36
ubuntu@ip-10-0-13-84:~$ sudo apt update
Hit:1 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Hit:3 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Get:4 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [542 kB]
Get:5 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [133 kB]
Get:6 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [9048 B]
Get:7 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:8 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [384 kB]
Get:9 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [84.6 kB]
Fetched 1405 kB in 5s (291 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
12 packages can be upgraded. Run 'apt list --upgradable' to see them.
ubuntu@ip-10-0-13-84:~$
```

```
ubuntu@ip-10-0-13-84:~$ sudo apt install mysql-client -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
mysql-client is already the newest version (8.0.39-0ubuntu0.24.04.2).
```

```
ubuntu@ip-10-0-13-84:~$ mysql -h dev-db-instance.c9wqug8isood.ap-southeast-1.rds.amazonaws.com -u admin -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 515
Server version: 8.0.39 Source distribution
```

SQL COMMANDS

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

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

Database changed
mysql>
```

```
mysql> show tables;
+-----+
| Tables_in_react_node_app |
+-----+
| author |
| book |
+-----+
2 rows in set (0.00 sec)
```

```
mysql> desc author ;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id | int | NO | PRI | NULL | auto_increment |
| name | varchar(255) | NO | | NULL |
| birthday | date | NO | | NULL |
| bio | text | NO | | NULL |
| createdAt | date | NO | | NULL |
| updatedAt | date | NO | | NULL |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> desc book ;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id | int | NO | PRI | NULL | auto_increment |
| title | varchar(255) | NO | | NULL |
| releaseDate | date | NO | | NULL |
| description | text | NO | | NULL |
| pages | int | NO | | NULL |
| createdAt | date | NO | | NULL |
| updatedAt | date | NO | | NULL |
| authorId | int | YES | MUL | NULL |
+-----+-----+-----+-----+-----+-----+
8 rows in set (0.01 sec)
```

```

mysql> select * from author ;
+----+-----+-----+-----+
| id | name           | birthday | bio
+----+-----+-----+-----+
| 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        | 1960-07-11 | 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)

```

```

6 rows in set (0.00 sec)

mysql> select * from book ;
+----+-----+-----+-----+
| id | title           | releaseDate | description
+----+-----+-----+-----+
| 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 sophomores investigate a malevolent threat to their Hogwarts classmates, a menacing beast that hides within the castle. | 251 | 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. Austen satirizes the social classes of the English gentry through a witty and ironic narrative voice. | 224 | 2024-05-29 | 2024-05-29 | 3 |
| 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. | 317 | 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, enter the water and face his greatest fears. | 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, enter the water and face his greatest fears. | 317 | 2024-05-29 | 2024-05-29 | 1 |
+----+-----+-----+-----+
6 rows in set (0.00 sec)

```

COMMANDS

-- Create Database and User

1. The command To creates a new database in MySQL called react_node_app:

CREATE DATABASE react_node_app;

2. The command to creates a new user in MySQL named appuser who can connect from any host (%), with the password learnIT02#.

CREATE USER 'appuser'@'%' IDENTIFIED BY 'learnIT02#';

3. To grant privileges to the user, you'll likely need to run a GRANT statement. For example, to give appuser full privileges on a specific database:

GRANT ALL PRIVILEGES ON react_node_app.* TO 'appuser'@'%';

4. After granting privileges, make sure to run the following command to apply the changes:

FLUSH PRIVILEGES;

```
CREATE DATABASE react_node_app;
CREATE USER 'appuser'@'%' IDENTIFIED BY 'learnIT02#';
GRANT ALL PRIVILEGES ON react_node_app.* TO 'appuser'@'%';
FLUSH PRIVILEGES;
```

This SQL command creates a table named author in a MySQL database, which stores information about authors such as their name, birthday, biography, and timestamps for creation and updates.

```
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;
```

ENGINE=InnoDB: Specifies the storage engine for the table, which supports transactions and foreign keys.

AUTO_INCREMENT=8: The next value for the auto-increment field id will start at 8.

CHARSET=utf8mb4: Sets the character encoding to utf8mb4, supporting a wide range of characters, including emojis.

COLLATE=utf8mb4_0900_ai_ci: Sets the collation (sorting) rules for the table, ensuring case-insensitive comparison in Unicode.

This SQL command creates a table named book in a MySQL database, storing information about books such as their title, release date, description, number of pages, timestamps for creation and updates, and a foreign key linking each book to an author in the author table.

```
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`
) ENGINE=InnoDB AUTO_INCREMENT=10 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

TABLE AUTHOR

```
INSERT INTO `author`(`id`, `name`, `birthday`, `bio`, `createdAt`, `updatedAt`) VALUES
```

(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');

TABLE BOOK

```
INSERT INTO `book` (`id`, `title`, `releaseDate`, `description`, `pages`, `createdAt`, `updatedAt`, `authorId`) VALUES
```

(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) ;

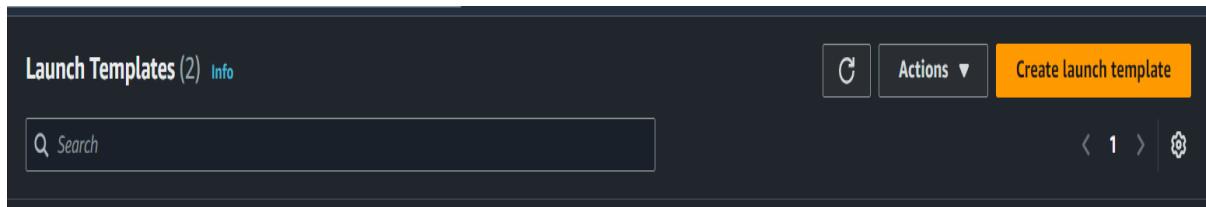
```
mysql> desc author ;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id   | int  | NO   | PRI | NULL    | auto_increment |
| name | varchar(255) | NO   |     | NULL    |                |
| birthday | date | NO   |     | NULL    |                |
| bio  | text  | NO   |     | NULL    |                |
| createdat | date | NO   |     | NULL    |                |
| updatedat | date | NO   |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> desc book ;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id   | int  | NO   | PRI | NULL    | auto_increment |
| title | varchar(255) | NO   |     | NULL    |                |
| releaseDate | date | NO   |     | NULL    |                |
| description | text  | NO   |     | NULL    |                |
| pages | int   | NO   |     | NULL    |                |
| createdat | date | NO   |     | NULL    |                |
| updatedat | date | NO   |     | NULL    |                |
| authorId | int   | YES  | MUL | NULL    |                |
+-----+-----+-----+-----+-----+-----+
8 rows in set (0.01 sec)
```

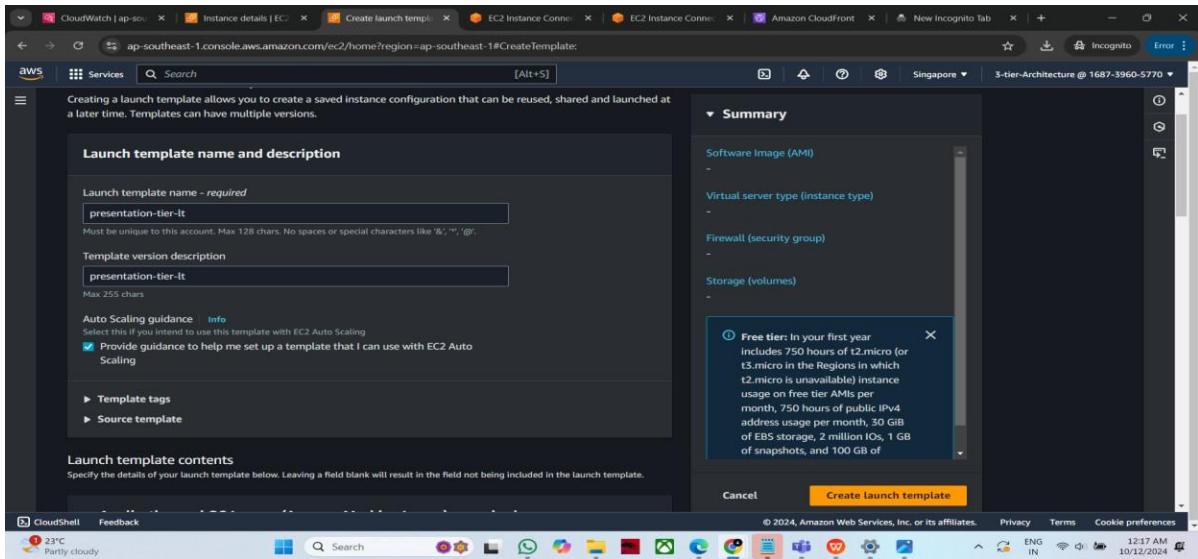
Step 5: Setting up Presentation Tier

5.1 Creating Presentation Tier Launch Template

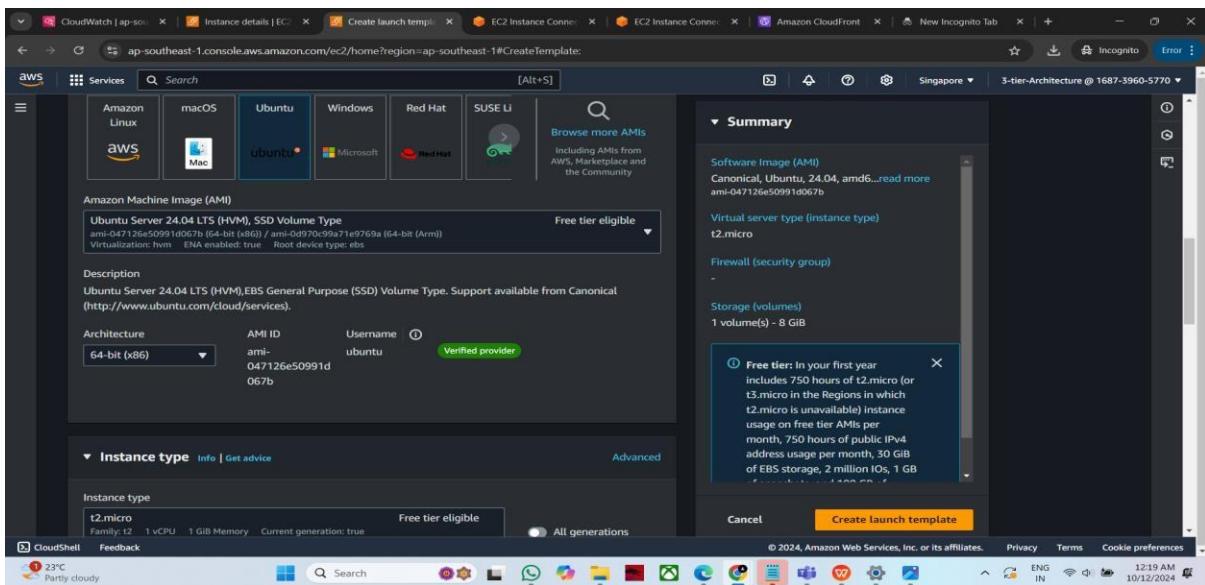
5.1.1 In Dashboard, search for ec2 service, click on Launch Template to open it.



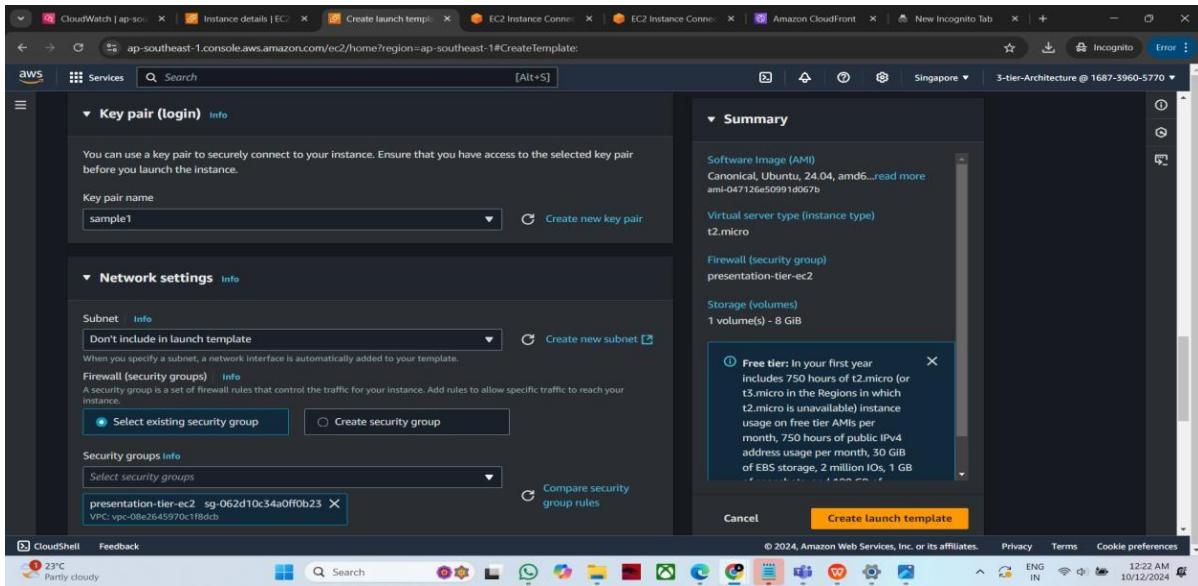
5.1.2 Launch Template Name and Description



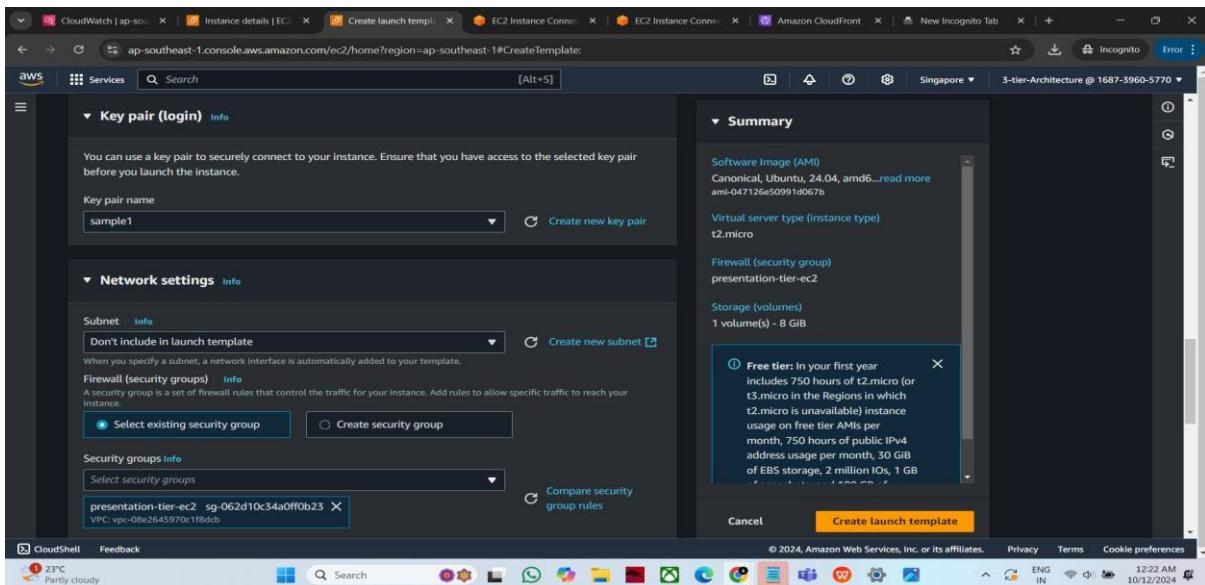
5.1.3 Launch Template Content and Instance Type



5.1.4 Key Pair and Network Setting



5.1.5 Advanced Details setting



```
#!/bin/bash

# Update the package list and install NGINX

sudo apt update -y
sudo apt install nginx -y

# Start and enable NGINX

sudo systemctl start nginx
sudo systemctl enable nginx

# Fetch metadata token
```

```
TOKEN=$(curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600")
```

```
# Fetch instance details using IMDSv2
```

```
INSTANCE_ID=$(curl -H "X-aws-ec2-metadata-token: $TOKEN" "http://169.254.169.254/latest/meta-data/instance-id")
```

```
AVAILABILITY_ZONE=$(curl -H "X-aws-ec2-metadata-token: $TOKEN" "http://169.254.169.254/latest/meta-data/placement/availability-zone")
```

```
PUBLIC_IP=$(curl -H "X-aws-ec2-metadata-token: $TOKEN" "http://169.254.169.254/latest/meta-data/public-ipv4")
```

```
# Create a simple HTML page displaying instance details
```

```
sudo bash -c "cat > /usr/share/nginx/html/index.html <<EOF
<h1>Instance Details</h1>
<p><b>Instance ID:</b> $INSTANCE_ID</p>
<p><b>Availability Zone:</b> $AVAILABILITY_ZONE</p>
<p><b>Public IP:</b> $PUBLIC_IP</p>
EOF"
```

```
# Restart NGINX to ensure changes are applied
```

```
sudo systemctl restart nginx
```

1. **#!/bin/bash**

```
#!/bin/bash
```

This line indicates that the script should be executed using the Bash shell. It's a standard way to specify the interpreter for the script.

2. **Update Package List and Install NGINX**

```
# Update the package list and install NGINX
sudo apt update -y
sudo apt install nginx -y
```

3. **Start and Enable NGINX**

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

4. Fetch Metadata Token

```
TOKEN=$(curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600")
```

TOKEN=\$(...): This line stores the output of the command within the parentheses in a variable named TOKEN.

curl -X PUT: Sends a PUT request to the instance metadata service (IMDSv2) to obtain a token required for accessing instance metadata. This token improves security by requiring that only requests with a valid token can retrieve metadata.

-H "X-aws-ec2-metadata-token-ttl-seconds: 21600": Specifies the TTL (Time To Live) for the token, indicating that it should remain valid for 21600 seconds (6 hours).

5. Fetch Instance Details using IMDSv2

```
# Fetch instance details using IMDSv2
```

```
INSTANCE_ID=$(curl -H "X-aws-ec2-metadata-token: $TOKEN"  
"http://169.254.169.254/latest/meta-data/instance-id")
```

```
AVAILABILITY_ZONE=$(curl -H "X-aws-ec2-metadata-token: $TOKEN"  
"http://169.254.169.254/latest/meta-data/placement/availability-zone")
```

```
PUBLIC_IP=$(curl -H "X-aws-ec2-metadata-token: $TOKEN"  
"http://169.254.169.254/latest/meta-data/public-ipv4")
```

INSTANCE_ID=\$(...): Retrieves the Instance ID of the EC2 instance and stores it in the INSTANCE_ID variable.

AVAILABILITY_ZONE=\$(...): Retrieves the Availability Zone in which the EC2 instance is located and stores it in the AVAILABILITY_ZONE variable.

PUBLIC_IP=\$(...): Retrieves the Public IP address of the EC2 instance and stores it in the PUBLIC_IP variable.

Each of these commands uses curl to send a GET request to the instance metadata service, passing the previously obtained token in the request header for authentication.

6. Create a Simple HTML Page Displaying Instance Details

```
# Create a simple HTML page displaying instance details  
sudo bash -c "cat > /var/www/html/index.html <<EOF  
<h1>Instance Details</h1>  
<p><b>Instance ID:</b> $INSTANCE_ID</p>  
<p><b>Availability Zone:</b> $AVAILABILITY_ZONE</p>  
<p><b>Public IP:</b> $PUBLIC_IP</p>  
EOF"
```

`sudo bash -c "...":` Executes the commands within the double quotes as a new Bash shell with elevated privileges (sudo).

`cat > /var/www/html/index.html <<EOF ... EOF:` Uses a here document to create and write content to the index.html file located in the NGINX default web root directory.

- `<h1>Instance Details</h1>:` Creates a header for the web page.
- `<p>Instance ID: $INSTANCE_ID</p>:` Displays the Instance ID in a paragraph.
- `<p>Availability Zone: $AVAILABILITY_ZONE</p>:` Displays the Availability Zone in a paragraph.
- `<p>Public IP: $PUBLIC_IP</p>:` Displays the Public IP in a paragraph.

This creates a simple HTML page that presents instance details to any client that accesses the web server.

7. Restart NGINX to Ensure Changes are Applied

```
# Restart NGINX to ensure changes are applied
sudo systemctl restart nginx
```

5.2 Creating Presentation Tier Target Groups

The screenshot shows the AWS Lambda console. At the top, there's a navigation bar with tabs like 'CloudWatch logs', 'Instance details', 'EC2', 'Stop 1 Create target...', 'EC2 Instance Connect', 'EC2 Instance Connect', 'Amazon CloudFront', 'New Incognito Tab', and 'Error'. Below the navigation bar, the main content area has a title 'Target groups (2) Info' and a search bar labeled 'Filter target groups'. There are two target group entries listed, each with an 'Actions' dropdown menu and a 'Create target group' button. The interface is dark-themed.

The screenshot shows the AWS Lambda console. At the top, there's a navigation bar with tabs like 'CloudWatch logs', 'Instance details', 'EC2', 'Stop 1 Create target...', 'EC2 Instance Connect', 'EC2 Instance Connect', 'Amazon CloudFront', 'New Incognito Tab', and 'Error'. Below the navigation bar, the main content area has a title 'Target groups (2) Info' and a search bar labeled 'Filter target groups'. There are two target group entries listed, each with an 'Actions' dropdown menu and a 'Create target group' button. The interface is dark-themed.

5.3 Creating Presentation Tier Load Balancer

5.3.1 Create Application Load Balancer

5.3.2 Basic Configuration

5.3.3 Network Mapping

The screenshot shows the 'Network mapping' step of the 'Create ALB Wizard'. The VPC dropdown is set to '3-tier-architecture-vpc' with the subnet 'subnet-09b154af1d583ba9' selected. Below it, two availability zones are listed: 'ap-southeast-1a (apse1-az1)' and 'ap-southeast-1b (apse1-az2)', each with its respective subnet and IPv4 CIDR range.

5.3.4 Security Group and Listeners and routing

The screenshot shows the 'Security groups' and 'Listeners and routing' steps of the 'Create ALB Wizard'. In the 'Security groups' section, the 'presentation-tier-alb' security group is selected. In the 'Listeners and routing' section, a new listener for port 80 is being configured.

5.3.5 Create Load Balancer

The screenshot shows the 'EC2 Dashboard' with the 'Load balancers' section open. A specific load balancer named 'presentation-tier-alb' is selected. The 'Details' tab shows the load balancer type as 'Application', status as 'Active', and VPC as 'vpc-08e2645970c1fbdc'. The 'Listeners and rules' tab shows one listener for port 80, and the 'Network mapping' tab shows the associated subnets and availability zones.

5.4 Creating Presentation Tier Auto Scaling Group

Step 1: Choose launch template

Group details		
Auto Scaling group name presentation-tier-asg		
Launch template	Version	Description
Launch template presentation-tier-it [x] lt-08001889f3ebfcfa96	Default	version 01

Step 2: Choose instance launch options

Network		
Network		
VPC	vpc-00a1b6ed443baf5b6 [x]	
Availability Zone	Subnet	
us-east-1a	subnet-0b6b70f3149f6a255 [x]	10.0.0.0/20
us-east-1b	subnet-0fce79fe22bc1573 [x]	10.0.16.0/20

Instance type requirements		
This Auto Scaling group will adhere to the launch template.		

Step 3: Configure advanced options

Load balancing		
Load balancer 1		
Name presentation-tier-alb [x]	Type Application/HTTP	Target group presentation-tier-tg [x]

VPC Lattice integration options		
VPC Lattice target groups -		

Health checks		
Health check type EC2, ELB	Health check grace period 300 seconds	

Additional settings		
Monitoring Enabled	Default instance warmup Disabled	

Step 4: Configure group size and scaling policies

Group size	
Desired capacity 3	Desired capacity type Units (number of instances)

Scaling		
Minimum desired capacity 2	Maximum desired capacity 4	
Target tracking policy Policy type Target tracking scaling	Scaling policy name Target Tracking Policy	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

Instance maintenance policy		
Replacement behavior No policy	Min healthy percentage -	Max healthy percentage -

Step 5: Add notifications

Notifications	
No notifications	

Step 6: Add tags

Tags (0)		
Key	Value	Tag new instances
No tags		

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

Step 6: Setting up Application Tier

Application Tier is handling the backend code which in nodejs. This tier is also consist of the several EC2 instances which are being created automatically using the auto scaling group based on the demand. Load Balancers are also configured to direct the traffic among the available EC2 instances. And following are the steps showing the implementation of the Application Tier.

6.1 Creating Application Tier Launch Template :

Go to the EC2 instance>Launch Template console and then click on ‘Launch Template’:

EC2 launch templates

Streamline, simplify and standardize instance launches

Use launch templates to automate instance launches, simplify permission policies, and enforce best practices across your organization. Save launch parameters in a template that can be used for on-demand launches and with managed services, including EC2 Auto Scaling and EC2 Fleet. Easily update your launch parameters by creating a new launch template version.

New launch template

Create launch template

Enter all the configuration details and click on ‘Create Launch Template’ :

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
application-tier-lt

Template version description
version 1

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

► Template tags
► Source template

Software Image (AMI)
-

Virtual server type (instance type)
-

Firewall (security group)
-

Storage (volumes)
-

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of

Cancel Create launch template

6.2 Creating Application Tier Target Group :

Go to EC2>Target Group console and click on ‘Create Target Group’:

EC2 > Target groups

Target groups (3) Info

Filter target groups

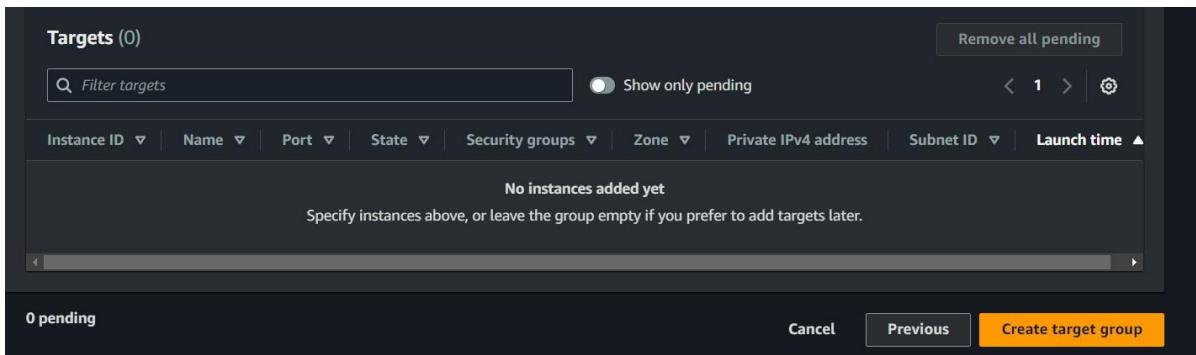
□	Name	▼	ARN	▼	Port	▼	Protocol	▼	Target type	▼	Load balancer
---	------	---	-----	---	------	---	----------	---	-------------	---	---------------

Create target group

Then enter the following configuration details :

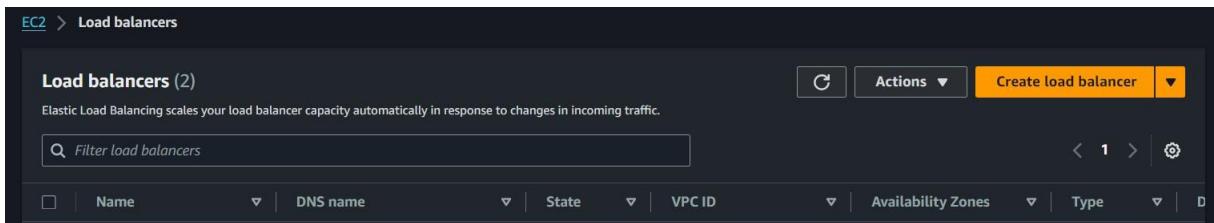
- Target Type
- Target Group Name
- Protocol Port
- Appropriate VPC where Application Tier is deployed
- Health Check Endpoint

Once everything is configured then click on ‘Create Target Group’ :



6.3 Creating Application Tier Load Balancer :

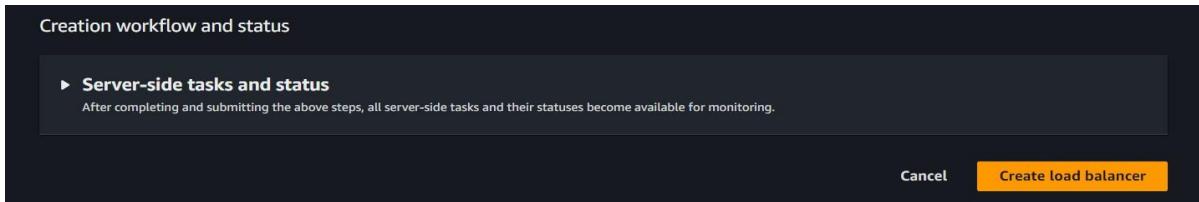
Go to EC2>Load Balancers console and click on ‘Create Load Balancer’ :



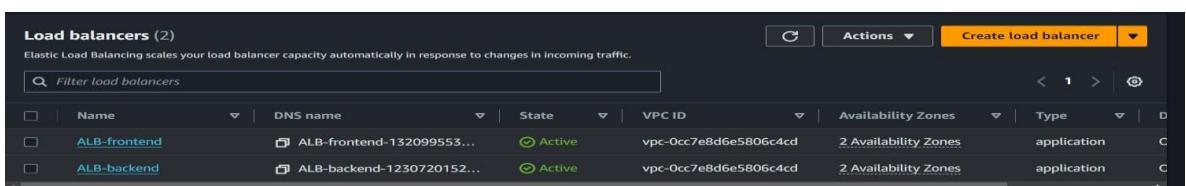
Then enter the configuration details like,

- Type of the Load Balancer
- Name of the Load Balancer
- Scheme (it should be internal in application tier)
- Appropriate VPC and Availability Zones
- Correct Security Groups
- Attach Target Group
-

Then click on ‘Create Load Balancer’:

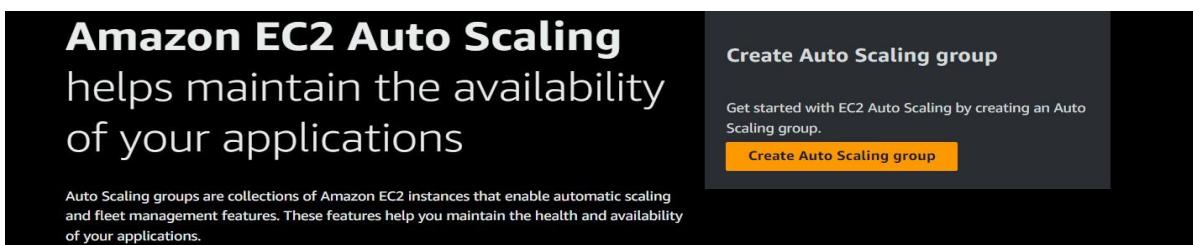


Then we can see the Load Balancer on the Dashboard :



6.4 Creating Application Tier Auto Scaling Group :

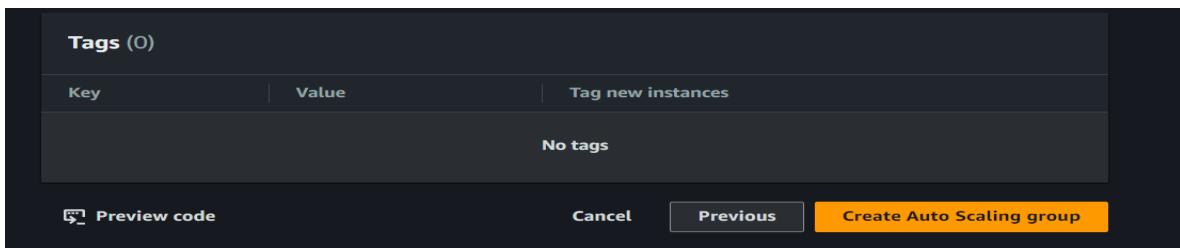
Go to EC2>Auto Scaling and click on ‘Create Auto Scaling Group’ :



Then enter the configuration details like,

- Name of the auto scaling group
- Then select a launch template
- Attach the load balancer
- Set health Check
- Scaling limits
- Integrate with cloudwatch
-

Then click on ‘Create Auto Scaling Group ’ :



After this it will automatically launch the new EC2 instances using the selected template and the number of the EC2 instances will depend on the desired capacity of the auto scaling group.

The bash script is designed to set up a Node.js application on an Ubuntu server. It includes steps for installing necessary packages, cloning a Git repository, configuring environment variables, and running the application using PM2 (a process manager for Node.js applications). Here’s a breakdown of what each section of the script does:

1. Update and Install Required Packages

```
# Update package list and install required packages
sudo apt update -y
sudo apt install -y git curl
```

```
sudo apt update -y: Updates the package list on the Ubuntu server.  
sudo apt install -y git curl: Installs Git (for version control) and curl (to download files).
```

2. Install Node.js

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

3. Install PM2

```
# Install PM2 globally  
sudo npm install -g pm2
```

sudo npm install -g pm2: Installs PM2 globally, allowing it to manage Node.js applications and keep them running in the background.

4. Define Variables

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

REPO_URL: The URL of the Git repository to clone.

BRANCH_NAME: The specific branch of the repository to check out.

REPO_DIR: The directory where the repository will be cloned.

ENV_FILE: The path to the environment variable file for the application.

5. Clone the Repository

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

cd /home/ubuntu: Navigates to the home directory of the Ubuntu user.
sudo -u ubuntu git clone \$REPO_URL: Clones the Git repository as the ubuntu user to ensure correct file permissions.
cd react-node-mysql-app: Changes into the cloned repository's directory.

6. Checkout to the Specific Branch

```
# Checkout to the specific branch  
sudo -u ubuntu git checkout $BRANCH_NAME  
cd backend
```

sudo -u ubuntu git checkout \$BRANCH_NAME: Checks out the specified branch within the repository.

cd backend: Changes into the backend directory of the cloned repository

6. Define Log Directory

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

LOG_DIR: Specifies the directory where logs will be stored.

mkdir -p \$LOG_DIR: Creates the log directory, including parent directories if they don't exist.
sudo chown -R ubuntu:ubuntu \$LOG_DIR: Changes the ownership of the log directory to the ubuntu user to ensure that the application can write logs.

7. Append Environment Variables

```
# Append environment variables to the .env file  
echo "LOG_DIR=$LOG_DIR" >> "$ENV_FILE"  
echo "DB_HOST=\"dev-db-instance.c9wqug8isood.ap-southeast-1.rds.amazonaws.com\"" >> "$ENV_F  
echo "DB_PORT=\"3306\"" >> "$ENV_FILE"  
echo "DB_USER=\"appuser\"" >> "$ENV_FILE"  
echo "DB_PASSWORD=\"learnIT02#\" >> "$ENV_FILE" # Replace with actual password  
echo "DB_NAME=\"react_node_app\"" >> "$ENV_FILE"
```

Appends various environment variables to the .env file in the backend directory, which typically include configurations for logging and database connection.

8. Install Node.js Dependencies

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

sudo -u ubuntu npm install: Installs the Node.js dependencies listed in the package.json file, run as the ubuntu user.

9. Start the Application Using PM2

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

sudo -u ubuntu npm run serve: Starts the Node.js application using PM2. This command should be defined in the package.json file.

10. Ensure PM2 Restarts on Reboot

```
# Ensure PM2 restarts on reboot as ubuntu user
sudo -u ubuntu pm2 startup systemd
sudo -u ubuntu pm2 save
```

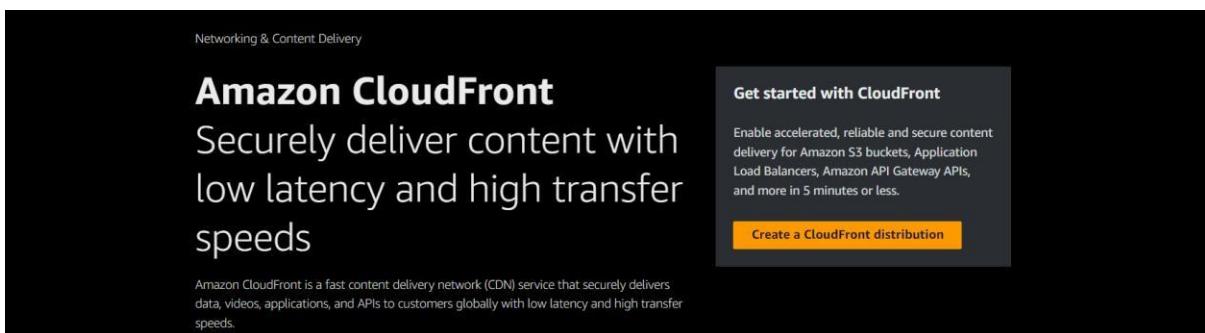
sudo -u ubuntu pm2 startup systemd: Configures PM2 to launch on system startup.

sudo -u ubuntu pm2 save: Saves the current list of processes managed by PM2 so they can be restored on reboot

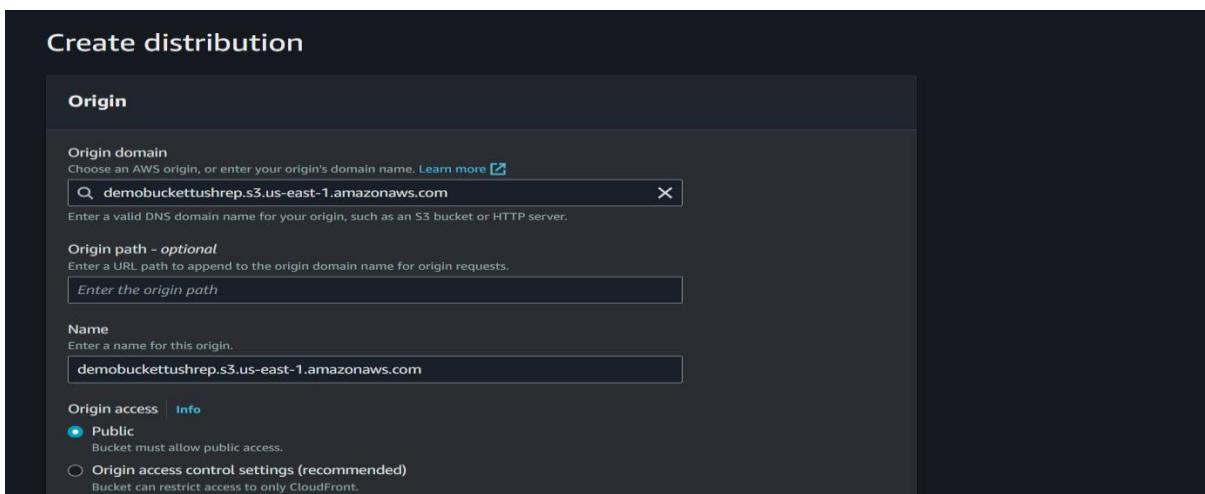
Step 7: Creating CloudFront Distribution

CloudFront is AWS's content delivery network (CDN) that helps distribute content globally with low latency. To create a CloudFront distribution, you configure an origin (such as an S3 bucket or an HTTP server) from which CloudFront will fetch content. The distribution routes requests through edge locations worldwide, caching content to improve access speed.

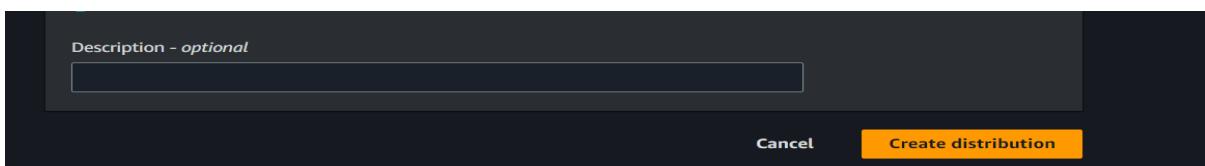
7.1 Go to CloudFront Console and Click on ‘Create a CloudFront distribution’:



7.2 Then configure the distribution by entering appropriate details:



7.3 After setting all the information click on ‘Create Distribution’ :



7.4 After Successful Creation we can see the distribution on the dashboard:

Distributions (1) Info									
<input type="text"/> Search all distributions									
	ID	Description	Type	Domain	Alternate	Origins	Status	Last modified	
<input type="checkbox"/>	E3975BRO96MXRK	-	Production	d3qc96f1...	-	presentation-tik	Enabled	October 11, 2024 at 4:22:45 AM UTC	Edit

Screenshot of the AWS CloudFront distribution configuration page.

The URL in the browser is `us-east-1.console.aws.amazon.com/cloudfront/v4/home?region=ap-southeast-1#/distributions/E397SBRO96MXRK`.

The left sidebar shows the CloudFront navigation menu:

- Distributions
- Policies
- Functions
- What's new
- Telemetry
- Monitoring
- Alarms
- Logs
- Reports & analytics
- Cache statistics
- Popular objects
- Top referrers
- Usage
- Viewers
- Security
- Origin access
- Field-level encryption
- Key management
- CloudShell
- Feedback

The main content area displays the distribution details for **E397SBRO96MXRK**:

General tab selected.

Details section:

Distribution domain name	ARN	Last modified
d3qc96f1o4gegp.cloudfront.net	arn:aws:cloudfront::168739605770:distribution/E397SBRO96MXRK	October 11, 2024 at 4:22:45 AM UTC

Settings section:

Description	Alternate domain names	Standard logging
Price class	-	Off
Use only North America and Europe	-	Cookie logging
Supported HTTP versions	-	Off
HTTP/2, HTTP/1.1, HTTP/1.0	-	Default root object

Bottom right corner shows system status: 23°C, Partly cloudy, ENG IN, 10/12/2024, 1:54 AM.

5. CONCLUSION

In this project, we successfully designed and implemented a scalable cloud infrastructure on AWS using Ubuntu servers to support centralized administration in a large network. The infrastructure was built to handle critical applications, including web hosting, API services, data processing, and a microservices architecture. By leveraging AWS services like EC2, VPC, CloudFront, and RDS, along with tools for centralized management, high availability, and cost optimization, we achieved the project's goals.

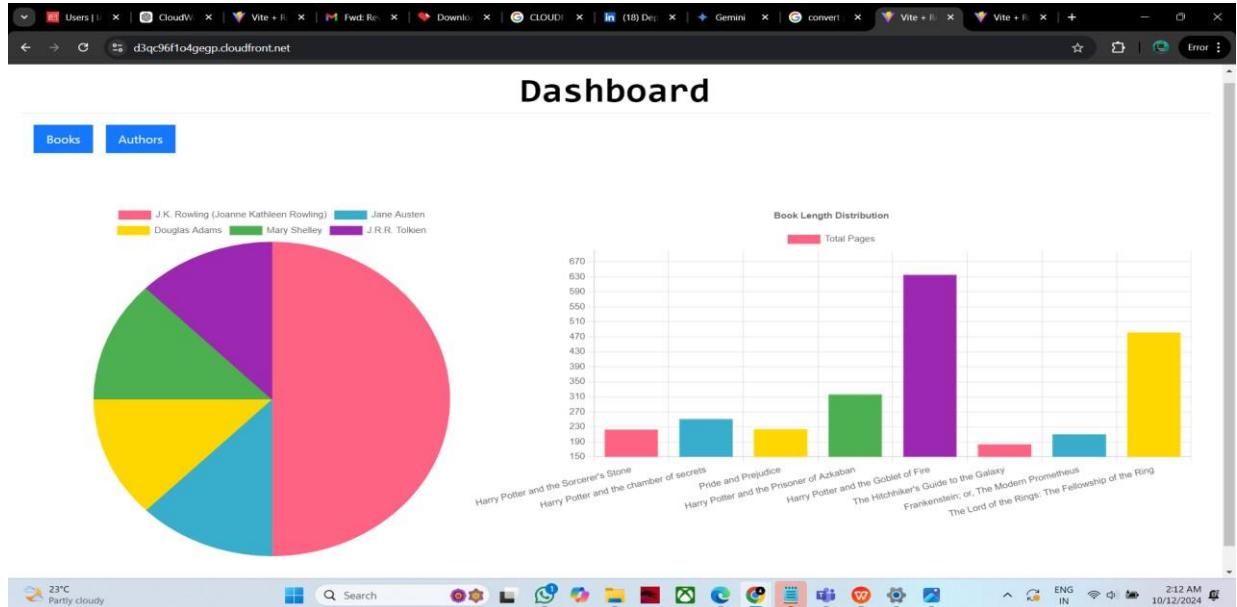
The system's scalability allows horizontal or vertical growth in response to demand, ensuring consistent performance as the user base expands. With centralized management tools and best practices for server administration, administrative overhead is minimized. High availability was ensured by employing multi-AZ deployments, automated failovers, and load balancing mechanisms, resulting in a highly resilient infrastructure with 99.99% uptime.

Additionally, cost efficiency was achieved through the use of AWS pricing models and resource optimization techniques, maintaining an effective balance between performance and expenditure. Security considerations, including encryption, least-privilege access, and regular audits, were prioritized throughout the project to safeguard the infrastructure.

Overall, this project demonstrates the effective use of AWS to build a robust, scalable, and secure cloud infrastructure, enabling the seamless operation of critical services for a large, centralized network. The design lays a strong foundation for future expansion and operational agility, while also ensuring cost-effective and secure cloud operations.

6. RESULT

Task to Designing and Implementing a Scalable Cloud Infrastructure on AWS for Ubuntu Server Administration in a centralized large network successfully accomplished.



The 'MANAGE BOOKS' page lists five books with their details:

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	

MANAGE AUTHORS

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 2004 TV comedy, a 2005 video game, a 2005 feature film, and a 2009 annual film.	2024-05-29T00:00:00.000Z	2024-05-29T00:00:00.000Z	

23°C Partly cloudy ENG IN 2:12 AM 10/12/2024