3 TIER ARCHITECTURE

Server full Architecture

Ramnath P
ramnathraja935@gmail.com
+91 9791438315

Table Of Content

Table Of Content

1.Abstract

This project explores the implementation of a three-tier serverful architecture, designed to enhance scalability, maintainability, and performance for web applications. The architecture consists of three distinct layers: the presentation layer, the application layer, and the database layer. The presentation layer, responsible for user interface interactions, ensures a responsive and dynamic user experience. The application layer handles business logic and processes client requests, enabling robust and efficient data manipulation. Finally, the database layer manages data storage and retrieval, ensuring data integrity and security.

By utilizing a serverful approach, each tier can be independently scaled and optimized based on demand, allowing for improved resource management and reduced latency. This architecture facilitates the separation of concerns, making it easier to maintain and update individual components without disrupting the overall system. Additionally, it supports integration with third-party services and APIs, enhancing functionality and adaptability.

This project demonstrates the effectiveness of a three-tier serverful architecture through a case study, highlighting its advantages in real-world applications. The findings indicate significant improvements in performance and user satisfaction, showcasing the architecture as a viable solution for modern web development needs.

2. Introduction:

A serverless 3-tier architecture on AWS is a popular way to implement a multi-tier architecture for applications. It separates an application's functionality into three layers: presentation, business logic, and data. This architecture is suitable for a variety of applications, including web apps, mobile apps, and enterprise systems.

3. Features:

3.1 Automatic scaling

Serverless technologies automatically scale up or down based on traffic without the need for manual configuration.

3.2 High availability

Serverless technologies are built to be highly available.

3.3 Separation of functionality

The three layers of a 3-tier architecture separate an application's functionality, which enables scalability, modularity, and flexibility.

3.4 Components

A serverless 3-tier architecture on AWS uses a combination of components, including Amazon S3, Amazon CloudFront, Amazon API Gateway, and AWS Lambda.

4. Layers in 3 tier Architecture:

4.1 Presentation tier:

The layer where users interact with the application

4.2 Application tier:

Also known as the logic tier, this layer processes data collected by the presentation tier

4.3 Data tier:

Also known as the database tier, this layer stores and manages the application's data

5. 3 tier Architecture Sample Diagram:

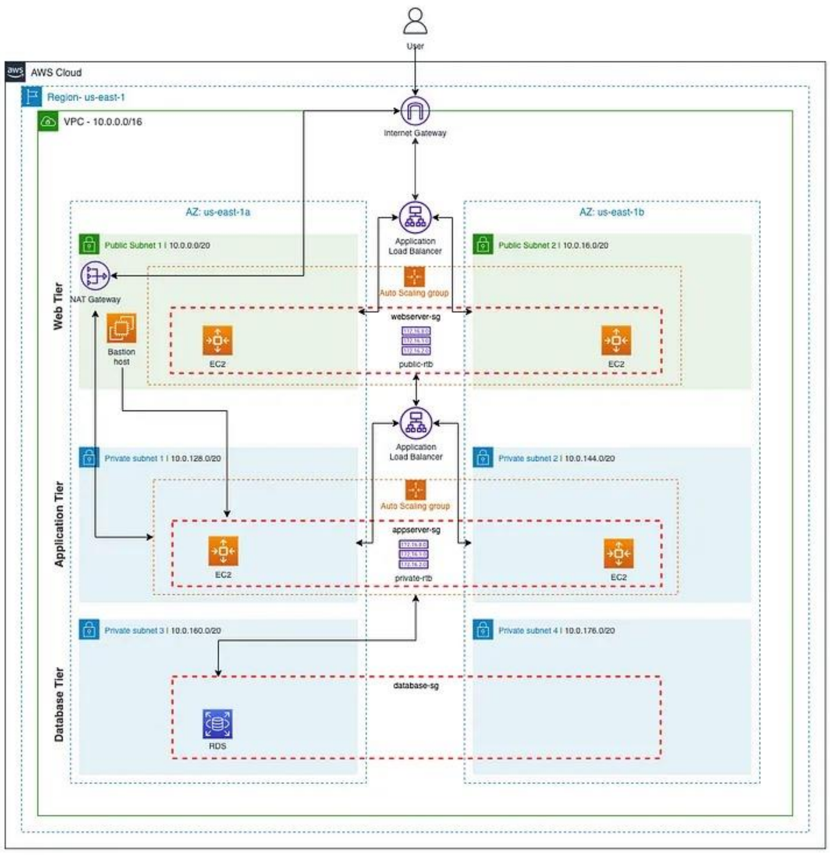
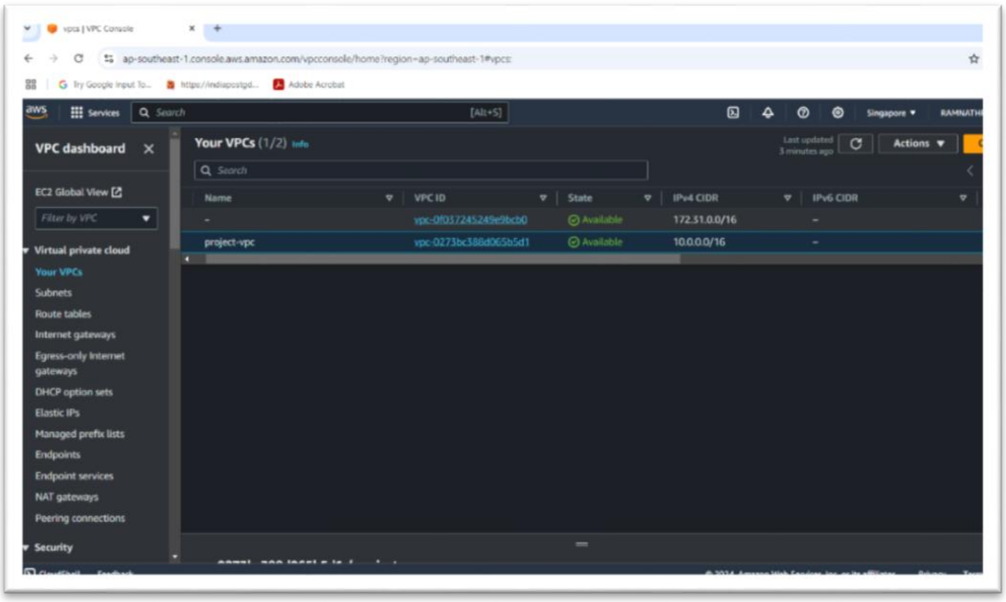


Fig 1. 3 Tier Architecture

6. Virtual Private Cloud:

A Virtual Private Cloud (VPC) is a secure, isolated section of a cloud provider's network where users can launch resources in a virtualized environment. It allows for customizable network configurations, including subnets, IP address ranges, and routing tables, while enabling secure communication between resources. VPCs enhance security by providing private IP addresses and the ability to define access controls through security groups and network ACLs. They are ideal for deploying applications that require a secure and controlled networking environment. Created a VPC named as Project-VPC for this project.



Your VPCs (1/2)

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR
-	vpc-0f037245249e9bcb0	Available	172.31.0.0/16	-
project-vpc	vpc-0273bc388d065b5d1	Available	10.0.0.0/16	-

Fig 2. VPC Created for this project

7. Subnets:

A subnet (subnetwork) is a smaller, segmented part of a larger network. It divides the network into manageable sections, each with its own range of IP addresses. Subnets improve network performance by reducing congestion, enhance security through isolation, and simplify management. They enable efficient routing and can contain broadcast traffic, making networks more organized and efficient. The Subnets are as follows:

Project-Subnet-public1

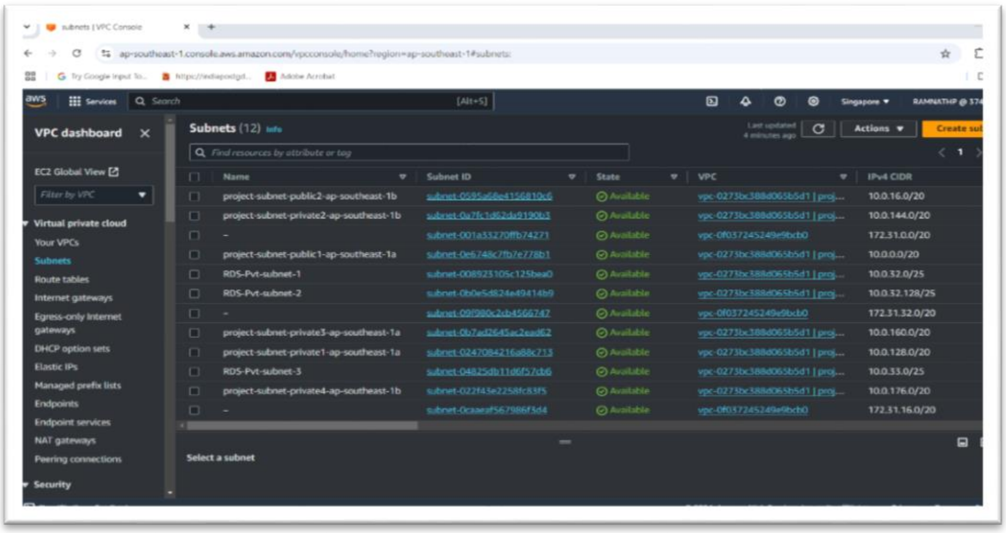
Project-Subnet-public2

Project-Subnet-private1

Project-Subnet-private2

Project-Subnet-private3

Project-Subnet-private4



Subnets (12)

Name	Subnet ID	State	VPC	IPv4 CIDR
project-subnet-public2-ap-southeast-1b	subnet-0595a68e4156810c6	Available	vpc-0273bc388d065b5d1 | proj...	10.0.16.0/20
project-subnet-private2-ap-southeast-1b	subnet-0a7fc1d62da9190b3	Available	vpc-0273bc388d065b5d1 | proj...	10.0.144.0/20
–	subnet-001a33270ffb74271	Available	vpc-0f037245249e9bcb0	172.31.0.0/20
project-subnet-public1-ap-southeast-1a	subnet-0e6748c7fb7e778b1	Available	vpc-0273bc388d065b5d1 | proj...	10.0.0.0/20
RDS-Pvt-subnet-1	subnet-008923105c125bea0	Available	vpc-0273bc388d065b5d1 | proj...	10.0.32.0/25
RDS-Pvt-subnet-2	subnet-0b0e5d824e49414b9	Available	vpc-0273bc388d065b5d1 | proj...	10.0.32.128/25
–	subnet-09f980c2cb4566747	Available	vpc-0f037245249e9bcb0	172.31.32.0/20
project-subnet-private3-ap-southeast-1a	subnet-0b7ad2645ac2ead62	Available	vpc-0273bc388d065b5d1 | proj...	10.0.160.0/20
project-subnet-private1-ap-southeast-1a	subnet-0247084216a88c713	Available	vpc-0273bc388d065b5d1 | proj...	10.0.128.0/20
RDS-Pvt-subnet-3	subnet-04825db11d6f57cb6	Available	vpc-0273bc388d065b5d1 | proj...	10.0.33.0/25
project-subnet-private4-ap-southeast-1b	subnet-022f43e2258fc83f5	Available	vpc-0273bc388d065b5d1 | proj...	10.0.176.0/20
–	subnet-0caaeaf567986f3d4	Available	vpc-0f037245249e9bcb0	172.31.16.0/20

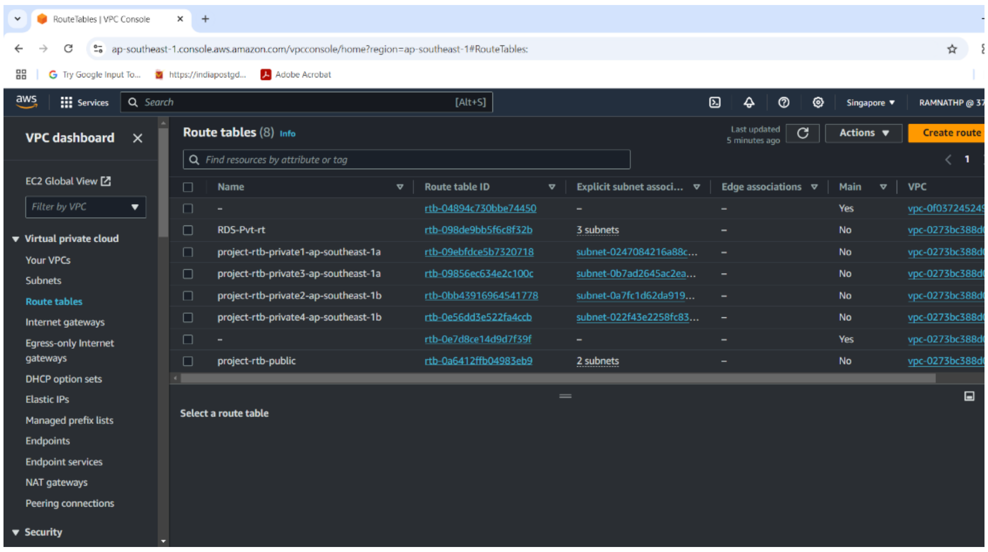
Select a subnet

Fig 3. Subnets created for this project

8. Route Table:

A route table is a set of rules used by routers to determine the best path for forwarding packets within a network. It contains entries that specify destination IP addresses, associated subnet masks, and the next hop or gateway for each route. Route tables help manage traffic, ensuring data is sent efficiently and correctly between different networks or subnets. They are essential for both local and internet routing, enabling effective communication between devices. The Subnets are connected to their respective route tables,

Project-rtb-public	- -> Connected to the 2 Public Subnet
Project-rtb-private1	- -> Connected to the Private Subnet1
Project-rtb-private2	- -> Connected to the Private Subnet2
Project-rtb-private3	- -> Connected to the Private Subnet3
Project-rtb-private4	- -> Connected to the Private Subnet4



Route tables (8) Info

Name	Route table ID	Explicit subnet associ...	Edge associations	Main	VPC
–	rtb-04894c730bbe74450	–	–	Yes	vpc-0f03724524
RDS-Pvt-rt	rtb-098de9bb5f6c8f32b	3 subnets	–	No	vpc-0273bc388d
project-rtb-private1-ap-southeast-1a	rtb-09ebfdce5b7320718	subnet-0247084216a88c...	–	No	vpc-0273bc388d
project-rtb-private3-ap-southeast-1a	rtb-09856ec634e2c100c	subnet-0b7ad2645ac2ea...	–	No	vpc-0273bc388d
project-rtb-private2-ap-southeast-1b	rtb-0bb43916964541778	subnet-0a7fc1d62da919...	–	No	vpc-0273bc388d
project-rtb-private4-ap-southeast-1b	rtb-0e56dd3e522fa4ccb	subnet-022f43e2258fc83...	–	No	vpc-0273bc388d
–	rtb-0e7d8ce14d9d7f39f	–	–	Yes	vpc-0273bc388d
project-rtb-public	rtb-0a6412ffb04983eb9	2 subnets	–	No	vpc-0273bc388d

Select a route table

Fig 4. Route tables created for this Project

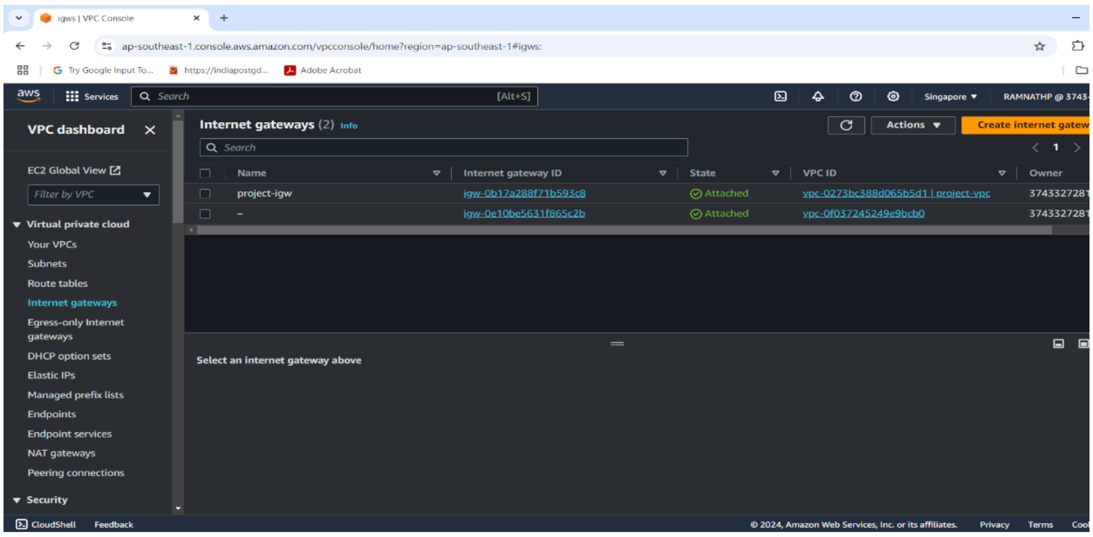
9. Internet Gateway:

An Internet Gateway is a horizontally scaled, redundant, and highly available component in a Virtual Private Cloud (VPC) that allows communication between instances in the VPC and the internet. It serves two main purposes:

Outbound Traffic: It enables resources in the VPC to access the internet.

Inbound Traffic: It allows external users to reach resources in the VPC, such as web servers, through public IP addresses.

The Internet Gateway ensures secure and efficient routing of traffic between the VPC and the internet.For the Internet Connectivity for the public subnets we have connected the Public Subnet where the public instances are connected to the Internet Gateway.



Internet gateways (2) Info

Name	Internet gateway ID	State	VPC ID	Owner
project-igw	igw-0b17a288f71b593c8	Attached	vpc-0273bc388d065b5d1 | project-vpc	3743327281
–	igw-0e10be5631f865c2b	Attached	vpc-0f037245249e9bcb0	3743327281

Select an internet gateway above

Fig 5. Internet Gateway created for this project

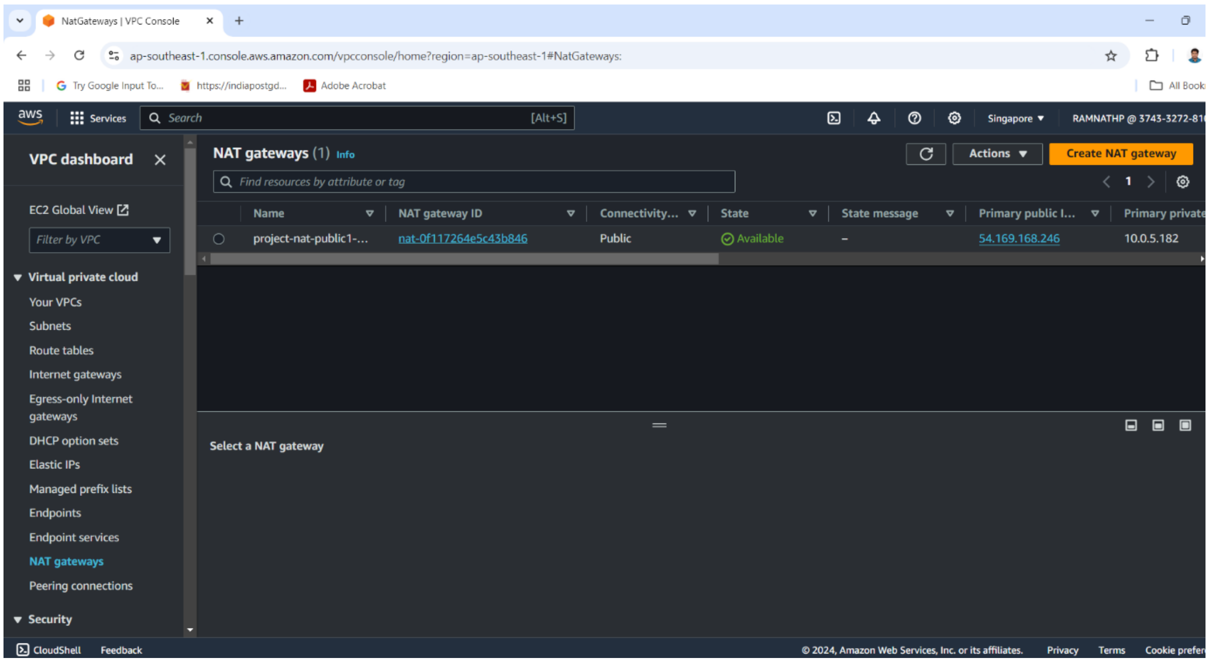
10. NAT Gateway:

NAT (Network Address Translation) is a process that modifies the IP address information in packet headers while they are in transit across a router or firewall. It serves two main purposes:

IP Address Conservation: NAT allows multiple devices on a private network to share a single public IP address, helping conserve the limited number of available public IPs.

Security: By hiding internal IP addresses from external networks, NAT adds a layer of security, making it harder for attackers to reach devices on the private network.

In cloud environments, NAT Gateways facilitate outbound internet access for instances in private subnets while keeping them inaccessible from the internet. For the Internet Connectivity for the private subnets we have connected the private Subnet where the private instances are connected to the NAT(Network Address Translation) Gateway.



NAT gateways (1) Info

Name	NAT gateway ID	Connectivity...	State	State message	Primary public I...	Primary private
project-nat-public1-...	nat-0f117264e5c43b846	Public	Available	–	54.169.168.246	10.0.5.182

Fig 6. NAT Gateway created for this project

11. Bastion Server:

A bastion server is a secure gateway that acts as a point of entry to a private network, typically used to manage and access resources in a secure manner. Here are its key features:

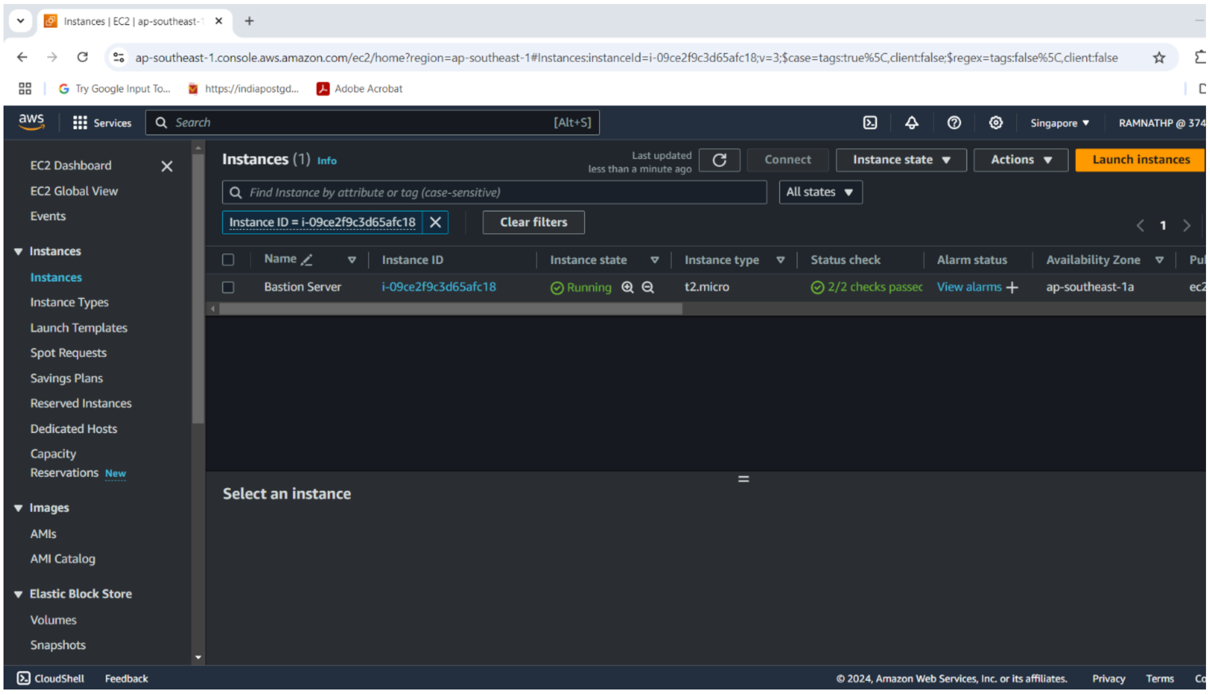
Access Control: Bastion servers provide controlled access to sensitive systems, allowing only authorized users to connect.

Secure Management: They often host tools for administrative tasks, such as SSH (Secure Shell) or RDP (Remote Desktop Protocol), enabling secure remote management of internal servers.

Monitoring: Bastion servers can log all access attempts and activities, aiding in auditing and security monitoring.

Isolation: They are typically placed in a public subnet, providing a buffer between the public internet and the private network, thus enhancing security.

Overall, bastion servers serve as a crucial security layer, ensuring safe access to resources in a protected environment.Bastion Server has been created for login to the application server. This works by first login to the Bastion server and then login to the application server.



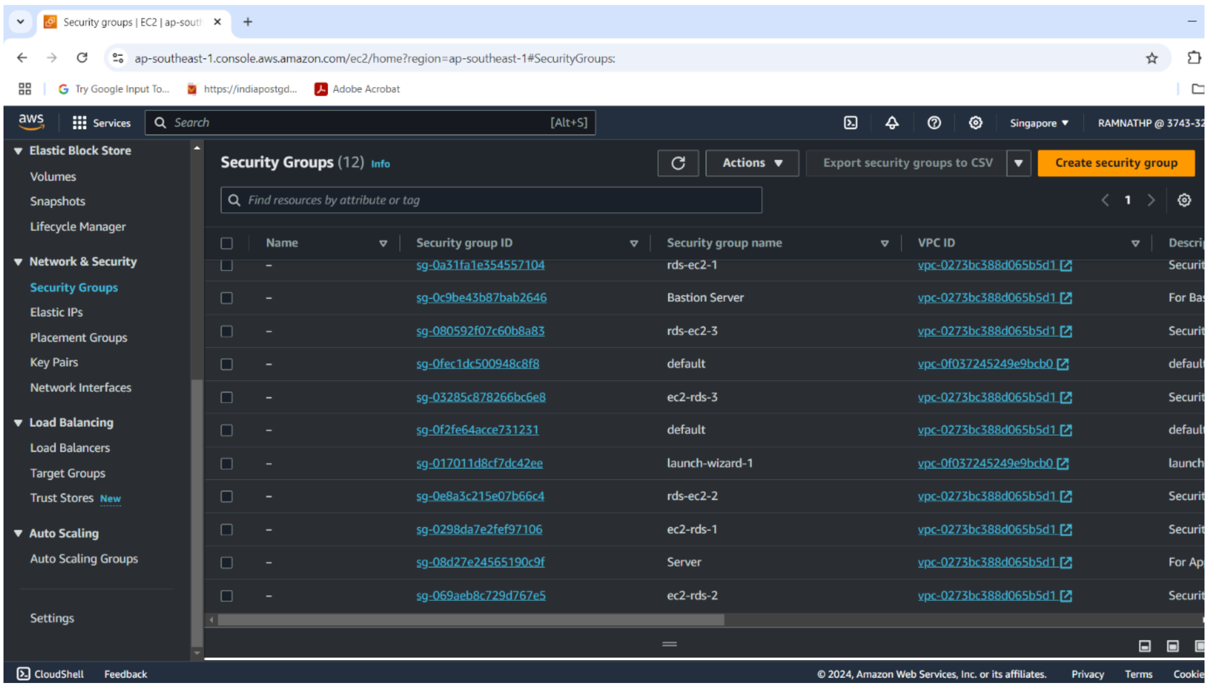
ap-southeast-1.console.aws.amazon.com/ec2/home?region=ap-southeast-1#Instances:instanceId=i-09ce2f9c3d65afc18;v=3;$case=tags:true%5C,client:false;$regex=tags:false%5C,client:false

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Pu
Bastion Server	i-09ce2f9c3d65afc18	Running	t2.micro	2/2 checks passec	View alarms +	ap-southeast-1a	ec2

Fig 7. Bastion Server created for this project

12. Security Group:

The Security group is used for securing the servers. This has both inbound and outbound rules. Inbound rules will be incoming traffic and outbound rules will be used for outgoing trafiic. The Bastion server inbound rule is set ssh port 22 as custom anywhere so that any incoming traffic can come to the Bastion server. The Server inbound rule is set ssh port 22 as the security group of the Bastion server so that from bastion server alone we can login to the app servers in private subnet.

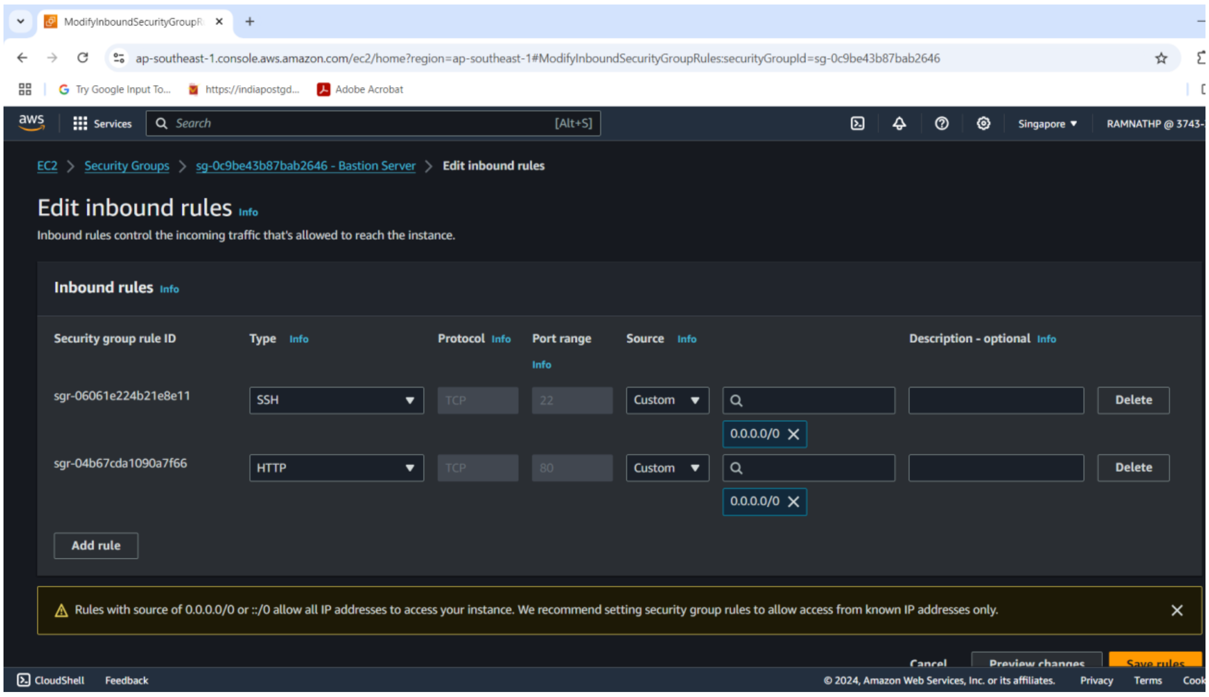


ap-southeast-1.console.aws.amazon.com/ec2/home?region=ap-southeast-1#SecurityGroups:

Security Groups (12)

Name	Security group ID	Security group name	VPC ID	Descri
–	sg-0a31fa1e354557104	rds-ec2-1	vpc-0273bc388d065b5d1	Securit
–	sg-0c9be43b87bab2646	Bastion Server	vpc-0273bc388d065b5d1	For Ba
–	sg-080592f07c60b8a83	rds-ec2-3	vpc-0273bc388d065b5d1	Securit
–	sg-0fec1dc500948c8f8	default	vpc-0f037245249e9bcb0	defaul
–	sg-03285c878266bc6e8	ec2-rds-3	vpc-0273bc388d065b5d1	Securit
–	sg-0f2fe64acce731231	default	vpc-0273bc388d065b5d1	defaul
–	sg-017011d8cf7dc42ee	launch-wizard-1	vpc-0f037245249e9bcb0	launch
–	sg-0e8a3c215e07b66c4	rds-ec2-2	vpc-0273bc388d065b5d1	Securit
–	sg-0298da7e2fef97106	ec2-rds-1	vpc-0273bc388d065b5d1	Securit
–	sg-08d27e24565190c9f	Server	vpc-0273bc388d065b5d1	For Ap
–	sg-069aeb8c729d767e5	ec2-rds-2	vpc-0273bc388d065b5d1	Securit

Fig 8.1 Security Group created for this project



ap-southeast-1.console.aws.amazon.com/ec2/home?region=ap-southeast-1#ModifyInboundSecurityGroupRules:securityGroupId=sg-0c9be43b87bab2646

EC2 > Security Groups > sg-0c9be43b87bab2646 - Bastion Server > Edit inbound rules

Edit inbound rules

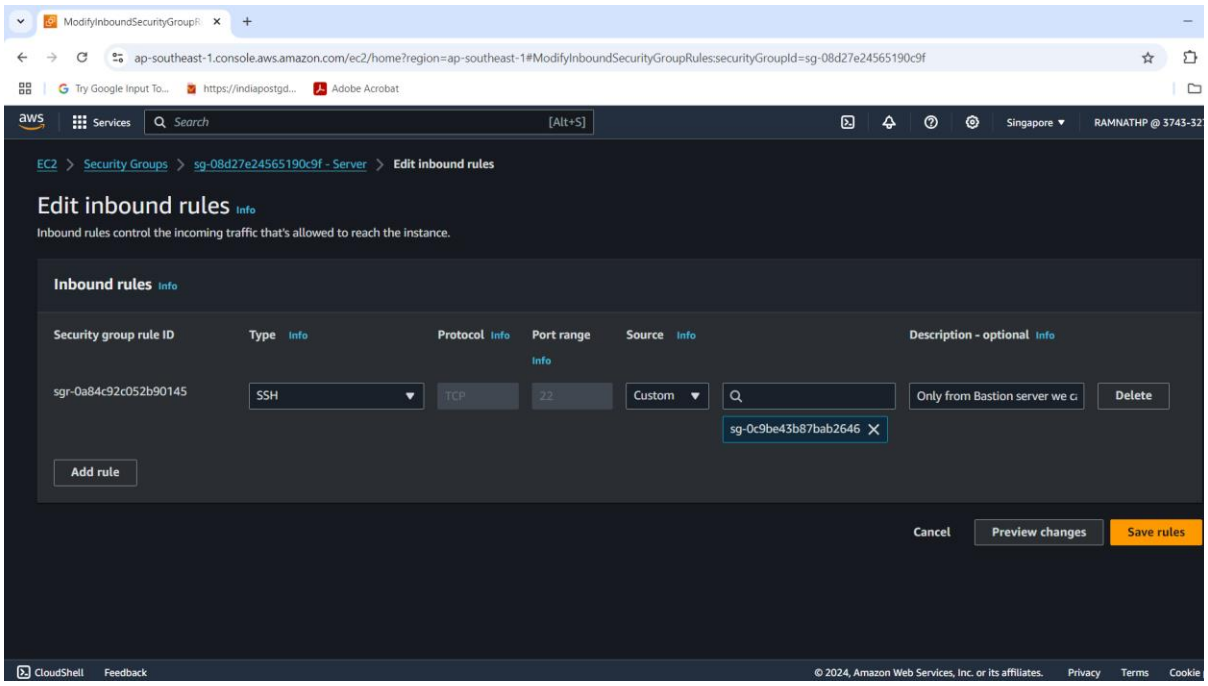
Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules

Security group rule ID	Type	Protocol	Port range	Source		Description - optional
sgr-06061e224b21e8e11	SSH	TCP	22	Custom	0.0.0.0/0	
sgr-04b67cda1090a7f66	HTTP	TCP	80	Custom	0.0.0.0/0	

Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Fig 8.2. Inbound rules for Bastion server & Web server


ap-southeast-1.console.aws.amazon.com/ec2/home?region=ap-southeast-1#ModifyInboundSecurityGroupRules:securityGroupId=sg-08d27e24565190c9f

EC2 > Security Groups > sg-08d27e24565190c9f - Server > Edit inbound rules

Edit inbound rules

Inbound rules control the incoming traffic that's allowed to reach the instance.

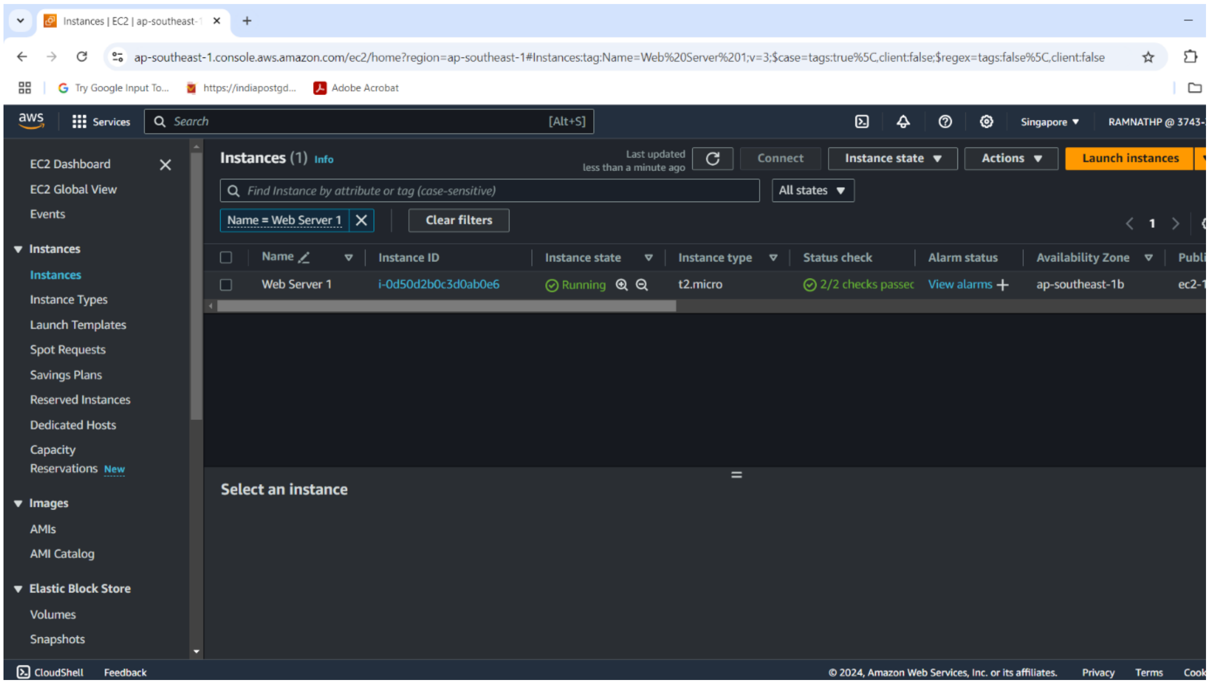
Inbound rules

Security group rule ID	Type	Protocol	Port range	Source		Description - optional
sgr-0a84c92c052b90145	SSH	TCP	22	Custom	sg-0c9be43b87bab2646	Only from Bastion server we c

Fig 8.3. Inbound rules created for App Server

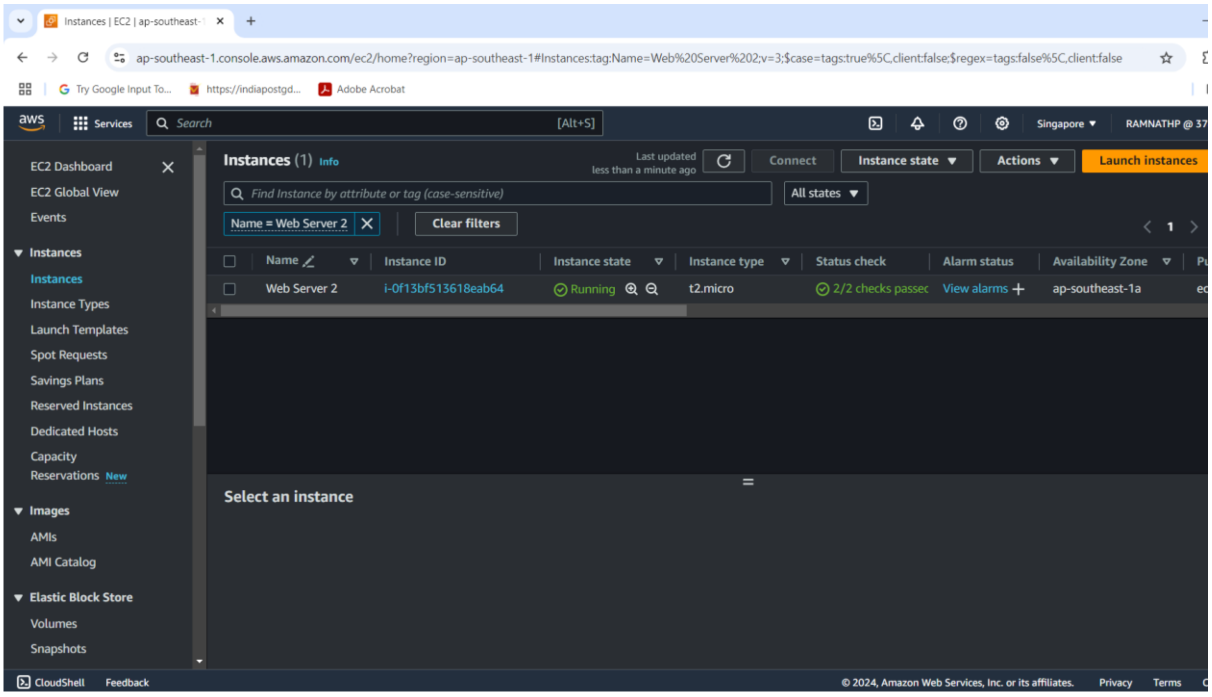
13. Web Servers:

A web server is a system that stores, processes, and delivers web content to users over the internet. It handles HTTP requests from clients (typically web browsers), serving static content like HTML pages, images, and CSS files, or dynamic content generated by web applications. Web servers can also manage other tasks such as SSL encryption for secure connections. Common examples include Apache, Nginx, and Microsoft IIS.Two Web servers are created in the Public Subnets (10.0.0.0/20 and 10.0.16.0/20) respectively. Here if we want we can create an extra security group for the web servers but here I have used the same Bastion server security group for the web server also.



Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
Web Server 1	i-0d50d2b0c3d0ab0e6	Running	t2.micro	2/2 checks passec	View alarms	ap-southeast-1b

Fig 9.1. Instance for Webserver1

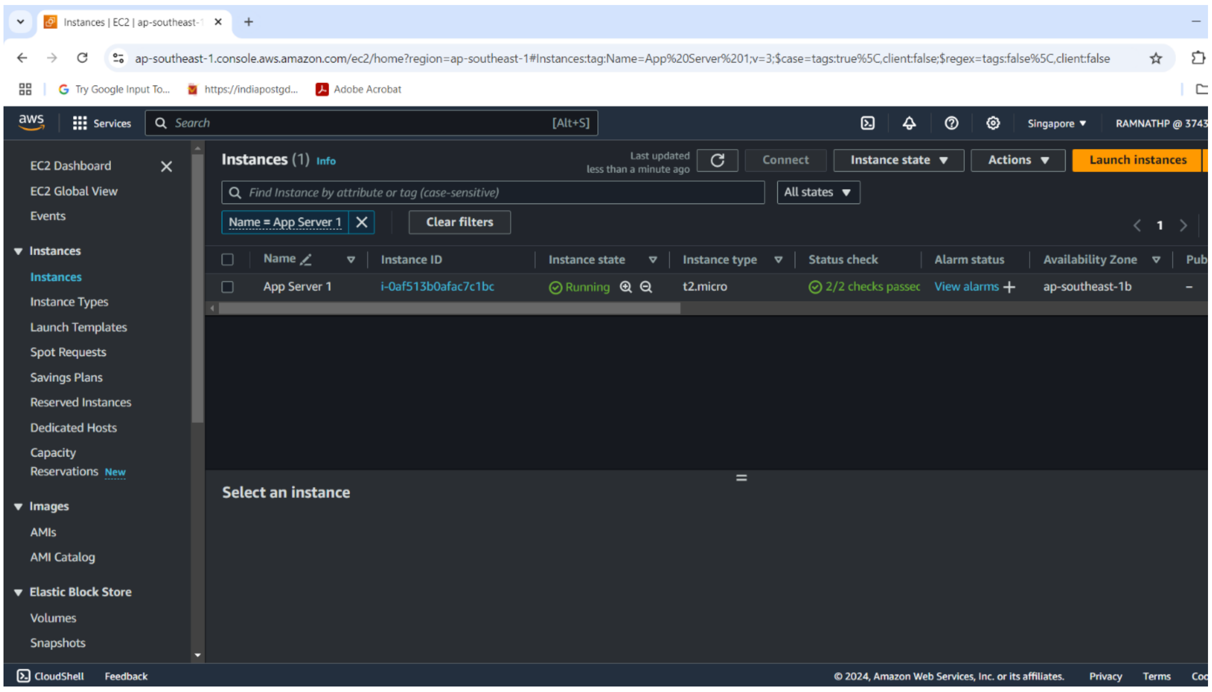


Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
Web Server 2	i-0f13bf513618eab64	Running	t2.micro	2/2 checks passec	View alarms	ap-southeast-1a

Fig 9.2. Instance for Webserver2

14. App Servers:

An application server is a software framework that provides a platform for running and managing web applications. It handles business logic and processes client requests, often interacting with databases and other services. Application servers facilitate the execution of dynamic content, enabling functionalities like session management, security, and transaction handling. Examples include Apache Tomcat, JBoss, and IBM WebSphere. They typically work in conjunction with web servers to deliver a complete web application experience.Two app servers are created in the private subnets(10.0.128.0/20 and 10.0.144.0/20) respectively.



ap-southeast-1.console.aws.amazon.com/ec2/home?region=ap-southeast-1#Instances:tag:Name=App%20Server%201;v=3;$case=tags:true%5C,client:false;$regex=tags:false%5C,client:false

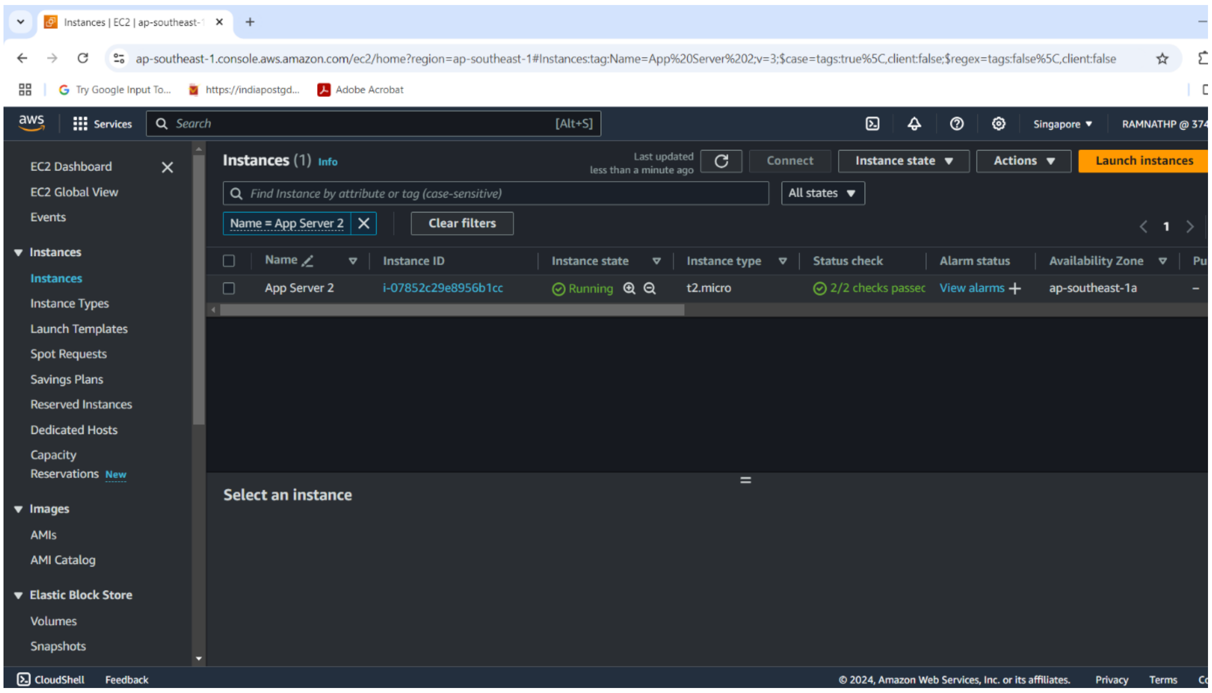
Instances (1)

Name = App Server 1

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Pub
App Server 1	i-0af513b0afac7c1bc	Running	t2.micro	2/2 checks passec	View alarms	ap-southeast-1b	–

Select an instance

Fig 10.1. Instance for Appserver1



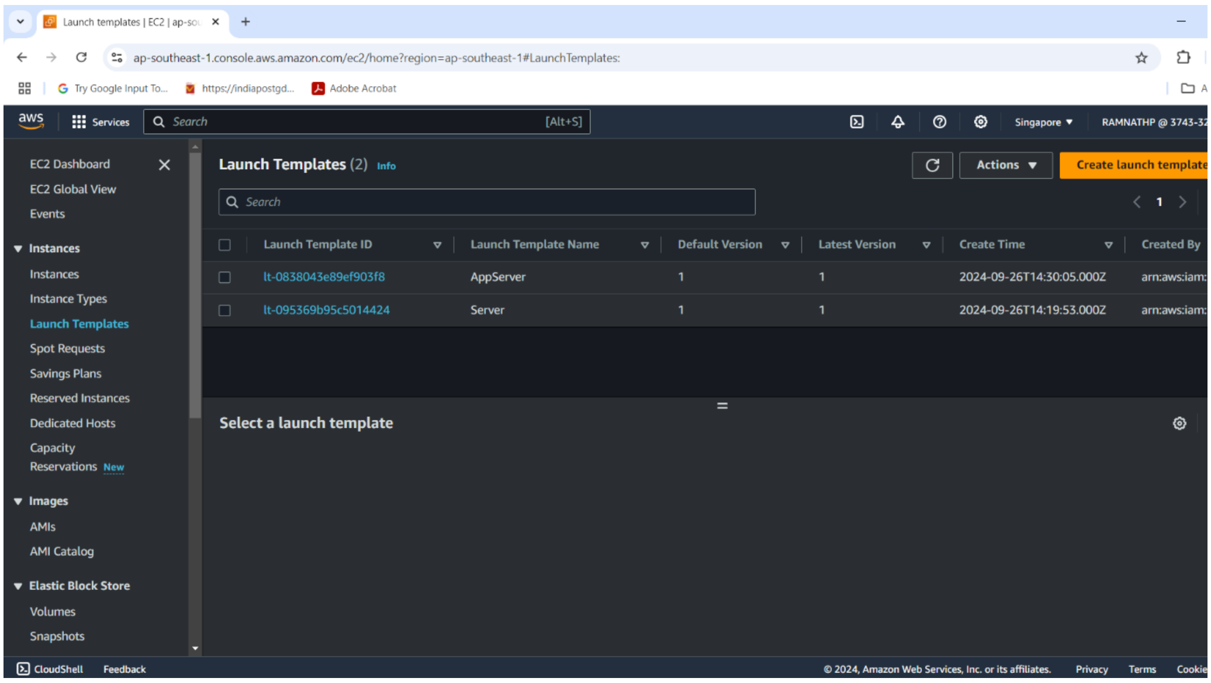
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Pu
App Server 2	i-07852c29e8956b1cc	Running	t2.micro	2/2 checks passec	View alarms +	ap-southeast-1a	–

Fig 10.2. Instance for Appserver2

To Login to these app servers we have to login to the Bastion Server and by adding the ssh key (Project key) to the Bastion server using WINSCP so that we can use the ssh -i "Project.pem" ec2-user@serverip command to login to the app servers.

15. Auto-Scaling Group (ASG):

An Auto Scaling Group (ASG) automatically adjusts the number of running instances based on demand. It can scale resources up or down in response to traffic and performance metrics, ensuring optimal performance and cost efficiency. ASGs also monitor instance health, replacing unhealthy ones to maintain application availability, and work with load balancers to distribute traffic effectively. Overall, they enhance the flexibility and resilience of cloud applications.Auto-Scaling Group is used to create/delete the instances based on the load. For this we have to create a template for web and app server.



Launch Templates (2)

Launch Template ID	Launch Template Name	Default Version	Latest Version	Create Time	Created By
lt-0838043e89ef903f8	AppServer	1	1	2024-09-26T14:30:05.000Z	arn:aws:iam:
lt-095369b95c5014424	Server	1	1	2024-09-26T14:19:53.000Z	arn:aws:iam:

Select a launch template

Fig 11. Launch Template for ASG

Server template is used for web server and AppServer template is used for App server. The desired capacity is set to 2 so that there will be 2 web and app servers in their respective subnets.

16. Load Balancer:

A load balancer is a device or software that distributes incoming network traffic across multiple servers to ensure no single server becomes overwhelmed. Key functions include:

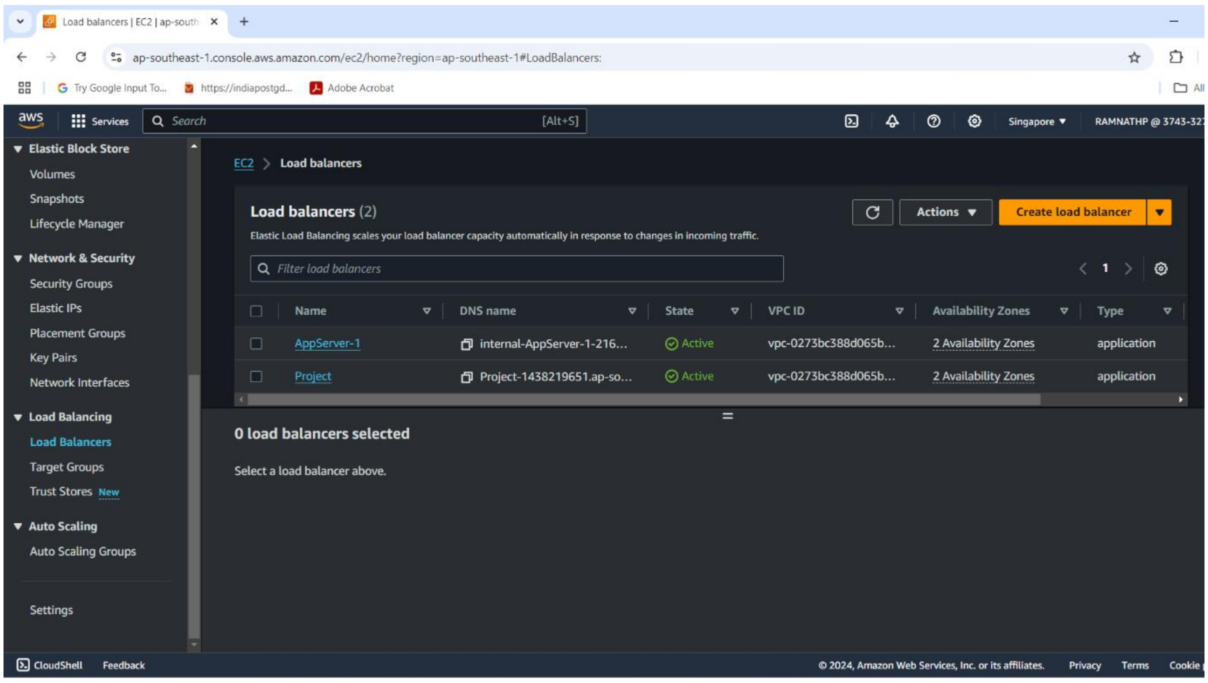
Traffic Distribution: It routes requests to various servers based on algorithms like round-robin or least connections.

High Availability: By spreading the load, it enhances application availability and reliability.

Health Monitoring: Load balancers regularly check the health of servers and reroute traffic away from any that are down.

Scalability: They facilitate scaling by allowing additional servers to be added seamlessly to handle increased traffic.

Overall, load balancers optimize resource use, improve response times, and ensure fault tolerance in applications.The Load Balancer is used to distribute the traffic to the instances. Two Load Balancers namely Project and AppServer-1 are created for web server and App server which are linked with auto-scaling group.



Load balancers (2)

Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.

Name	DNS name	State	VPC ID	Availability Zones	Type
AppServer-1	internal-AppServer-1-216...	Active	vpc-0273bc388d065b...	2 Availability Zones	application
Project	Project-1438219651.ap-so...	Active	vpc-0273bc388d065b...	2 Availability Zones	application

0 load balancers selected

Select a load balancer above.

Fig 12. Load Balancer for Web and App layer

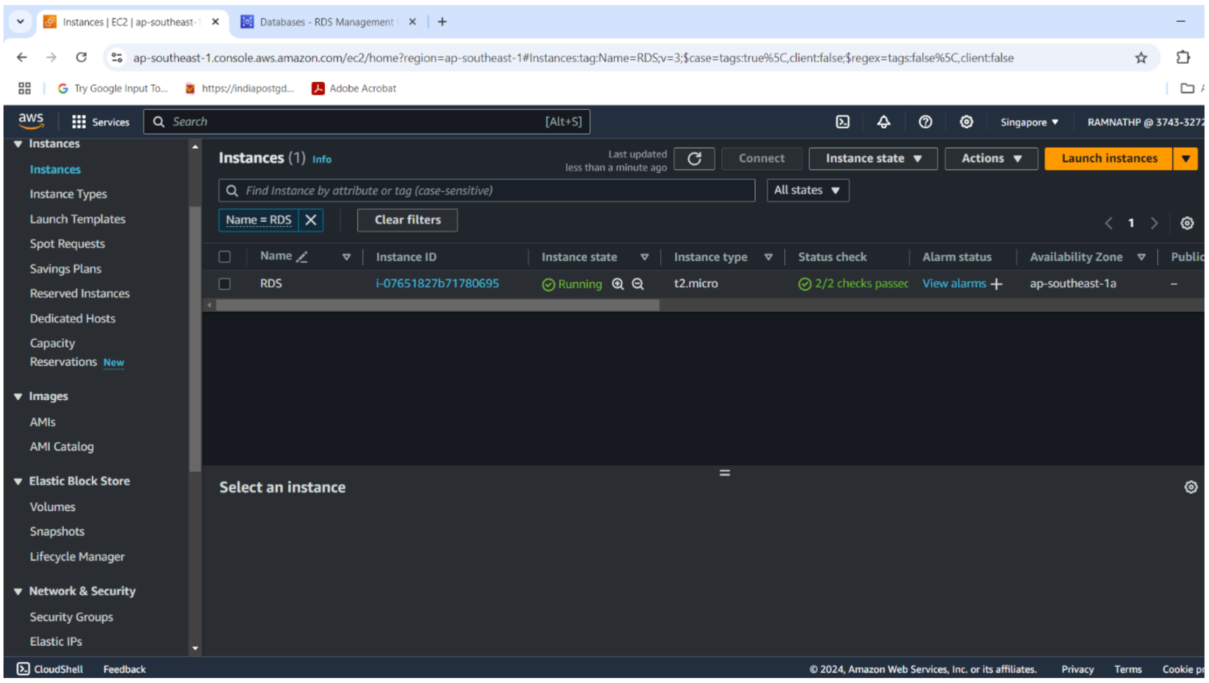
17. Database:

A database is an organized collection of data that can be easily accessed, managed, and updated. It allows users and applications to store, retrieve, and manipulate data efficiently. Databases can be classified into various types, such as:

Relational Databases: Use structured tables with predefined schemas (e.g., MySQL, PostgreSQL).

NoSQL Databases: Offer flexible data models, often for unstructured or semi-structured data (e.g., MongoDB, Cassandra).

Databases are essential for applications that require data storage, transaction management, and data integrity. They support various operations, including querying, updating, and reporting on data. Database-1 is created with mysql database engine with the username as admin and the password for the security. This Database is attached to an EC2 instance named RDS so that the database can be connected using this instance. This RDS instance is connected to the app server security group so that this can be access only through the Bastion server.

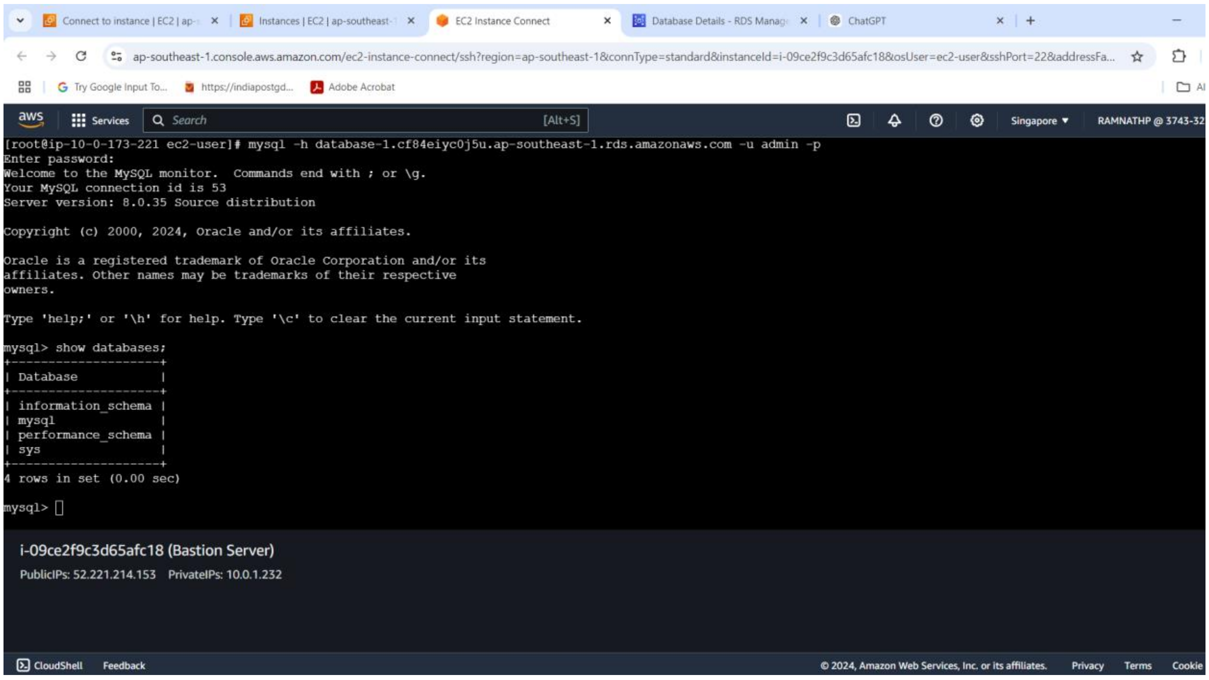


Instances (1)

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public
	RDS	i-07651827b71780695	Running	t2.micro	2/2 checks passec	View alarms	ap-southeast-1a	–

Fig 13. Instance created for RDS Database

For connecting to the database first we have to connect to the bastion server using the ssh-client command ssh -i "Project.pem" ec2-user@serverip and to connect the database using the command mysql -h endpoint -u admin -p and pass the password for the database set while creating the database.



```
[root@ip-10-0-173-221 ec2-user]# mysql -h database-1.cf84eiyc0j5u.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 53
Server version: 8.0.35 Source distribution

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

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

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

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

mysql> 
```

i-09ce2f9c3d65afc18 (Bastion Server)

PublicIPs: 52.221.214.153 PrivateIPs: 10.0.1.232

Fig 14. Connected to mysql using RDS Instance

18. Mysql Installation:

To install MySQL, follow these concise steps:

Download: Obtain the MySQL installer from the official MySQL website.

Run Installer: Execute the installer. Choose the setup type (Developer, Server, etc.).

Configuration: Follow the prompts to configure server settings, including root password and any additional options.

Install: Complete the installation process.

Start MySQL: After installation, start the MySQL service (if not automatically started).

Access MySQL: Use a command-line interface or a GUI tool (like MySQL Workbench) to connect to your MySQL server.

Verify Installation: Run `mysql --version` in the command line to confirm the installation.

This should set up MySQL on your system successfully!

18.1 Issue and Workaround:

While installing mysql we can face gpg key mismatch issue at that time we can add "--nogpgcheck" to ignore the gpg key mismatch issue.