BUILDING A SCALABLE AND SECURE WEB APPLICATION ON AWS WAF

Introduction

- **Amazon VPC** allows you to create an isolated network environment within AWS for hosting applications securely

.

- **Subnets** divide the VPC into logical segments (public/private) for better organization and security.
- **Internet Gateway (IGW)** enables communication between resources in the VPC and the internet.
- **Route Tables** control traffic flow between subnets, the internet, and other networks.
- **EC2 Instances** act as compute resources where your application runs.
- **Application Load Balancer (ALB)** distributes incoming traffic across multiple EC2 instances, ensuring scalability and availability.
- **AWS WAF (Web Application Firewall)** protects the application from common web exploits like SQL injection, XSS, and malicious bots.

Benefits

✅ **Security:**
Fine-grained network control with VPC, subnets, and security groups.
WAF filters malicious requests, reducing attack surface.

✅ **High Availability:**
ALB distributes traffic across multiple AZs, preventing single points of failure.

✅ **Scalability:**
Add or remove EC2 instances easily based on load, supported by ALB.

✅ **Centralized Control:**
One Web ACL can protect multiple resources (ALBs, API Gateways).

✅ **Visibility:**
WAF provides request logs, CloudWatch metrics, and insights into threats.

Drawbacks / Considerations

⚠️ **Cost:**
VPC components are free, but ALB, EC2, and WAF rules incur hourly and per-request charges.

⚠️ **Complexity:**
Proper planning is required for CIDR ranges, routing, and security group rules.

⚠️ **Latency:**
WAF inspection can add a slight delay to requests, especially with many rules.

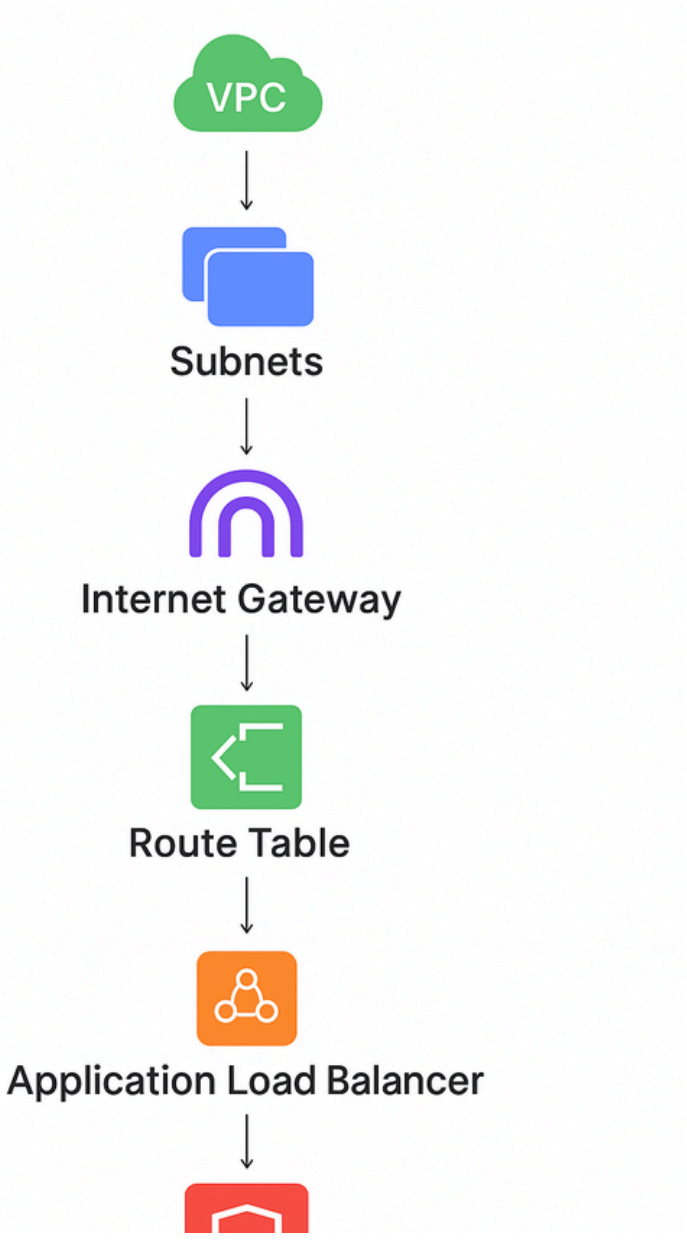
⚠️ **Rule Tuning:**
False positives may block legitimate users unless rules are tested in COUNT mode first.

Real-World Use Case

E-commerce Platform:
A company hosting its online store in AWS deploys EC2 instances in private subnets, uses an ALB in public subnets to distribute traffic, and applies WAF rules to protect against attacks like SQL injection, XSS, and DDoS attempts.
Customers access the site through the ALB's DNS, and malicious requests are blocked at the WAF level before reaching the application.

Workflow

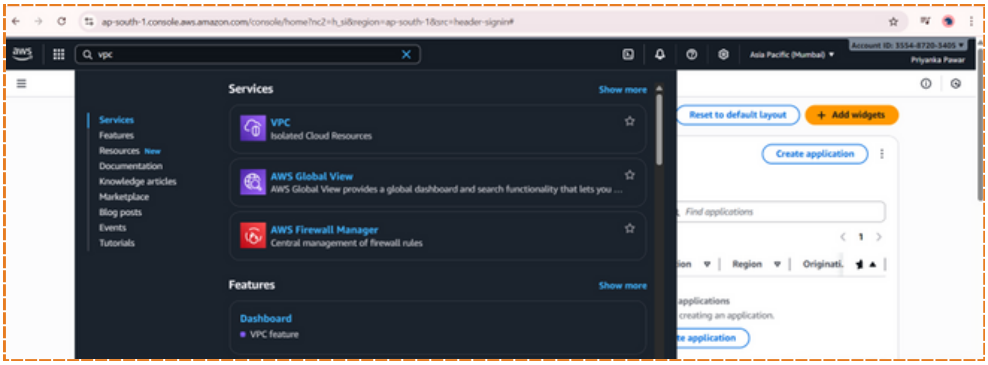
VPC – Creates an isolated network environment for your application.
Subnets – Divide the VPC into segments (public for ALB, private for EC2 if needed).
I**nternet Gateway** – Connects the VPC to the internet.
Route Table – Defines routing rules so public subnets can send traffic to the IGW.
EC2 Instances – Run your application code or web server.
Application Load Balancer (ALB) – Distributes incoming traffic across EC2 instances, improving availability.
A**WS WAF** – Filters and blocks malicious traffic before it reaches the ALB/EC2.



Console Steps

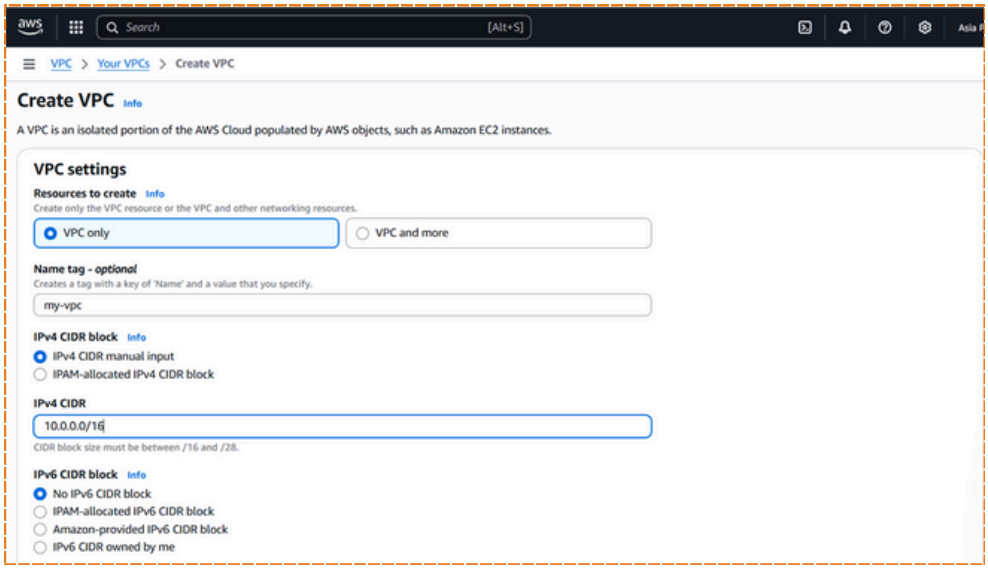
1) Create VPC

VPC → Create VPC

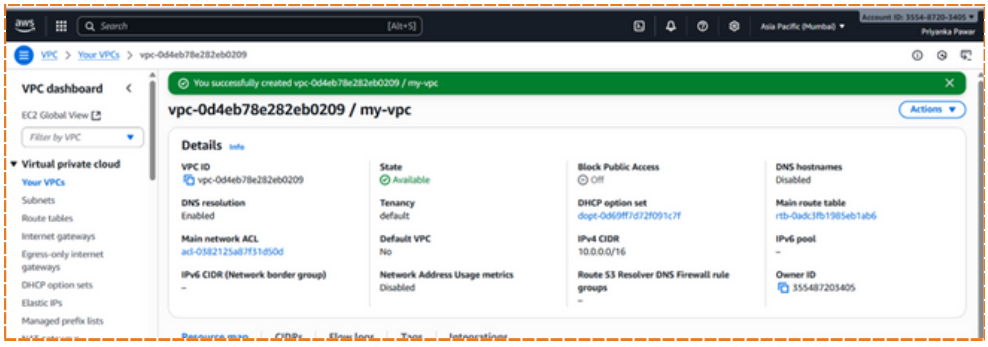


Name: my-vpc,
IPv4 CIDR: 10.0.0.0/16

Enable DNS hostnames/support.



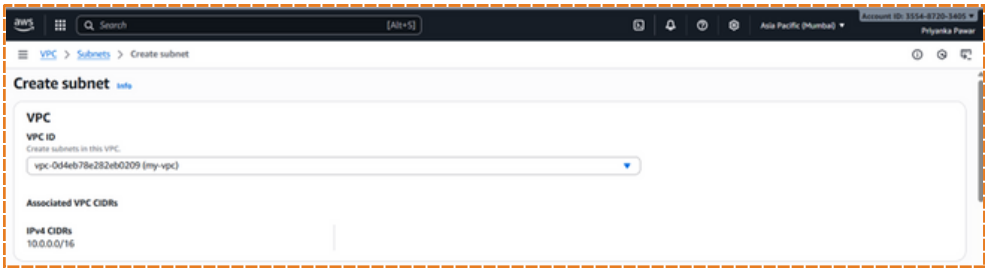
Check VPC sucessfully created



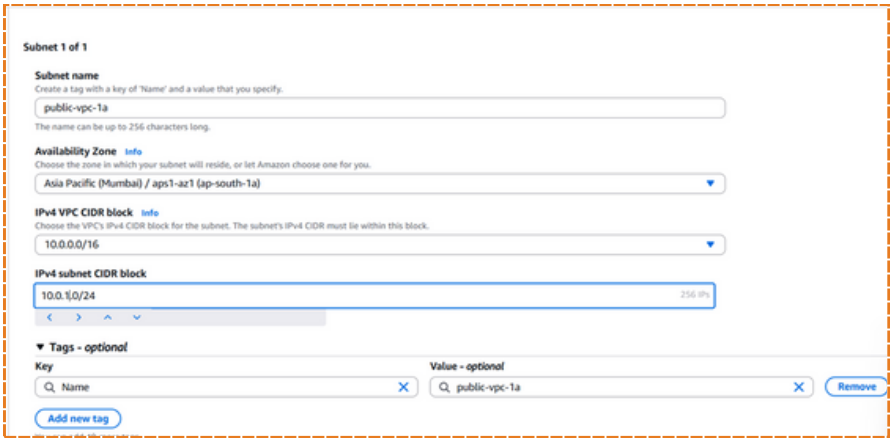
2) Create Subnets (2 public)

VPC >> Subnets >> Create Subnet

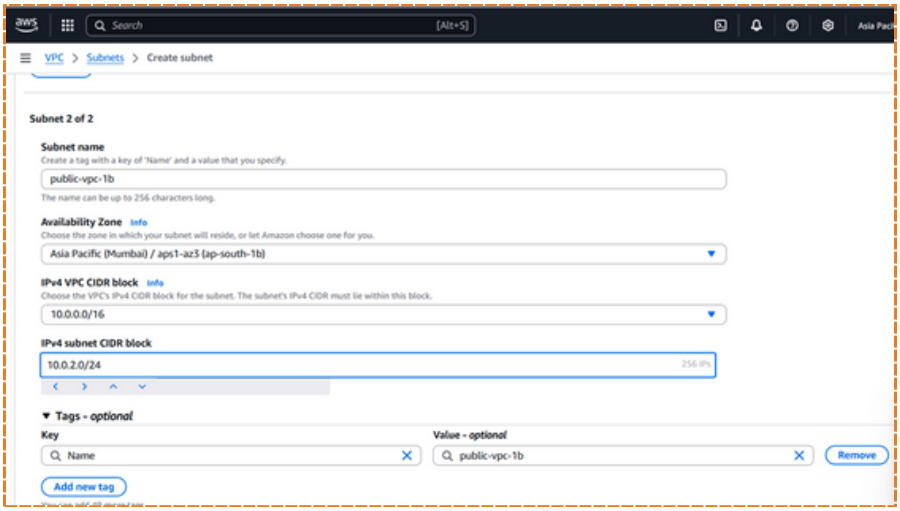
Select VPC ID {my-vpc}



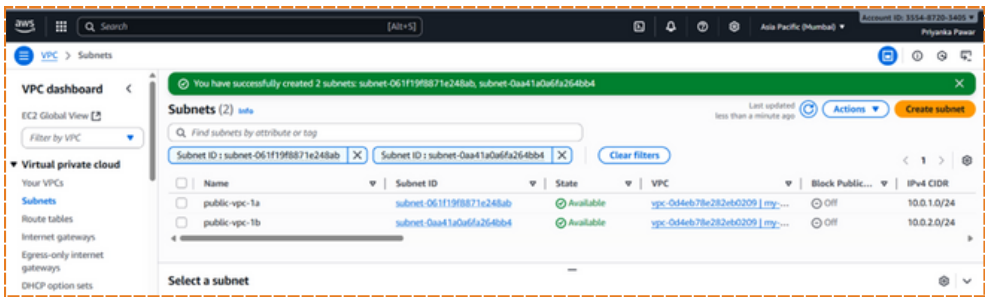
Subnet 1:
Subnet name : public-vpc-1a
Zone : (ap-south-1a)
IPv4 VPC CIDR : 10.0.0.0/16
IPv4 subnet : 10.0.1.0/24



Subnet 2:
Subnet name : public-vpc-1b
Zone : (ap-south-1b)
IPv4 VPC CIDR : 10.0.0.0/16
IPv4 subnet : 10.0.2.0/24

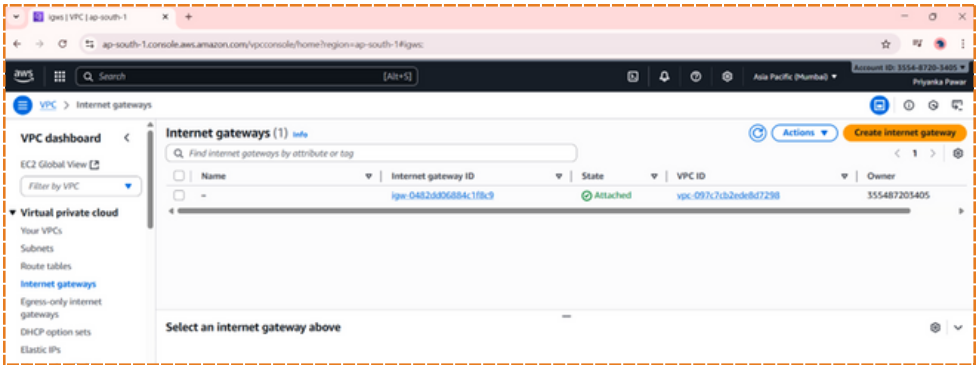


Check Subnets sucessfully created



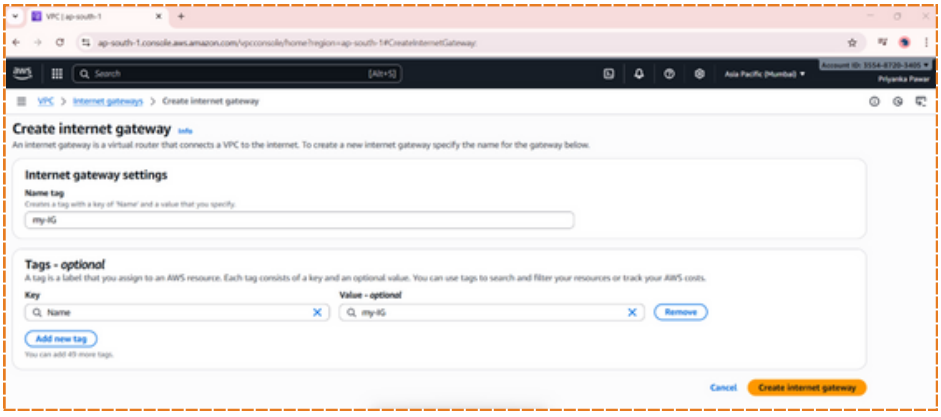
3) Internet Gateway (IGW)

VPC >> Internet Gateways >> Create IGW

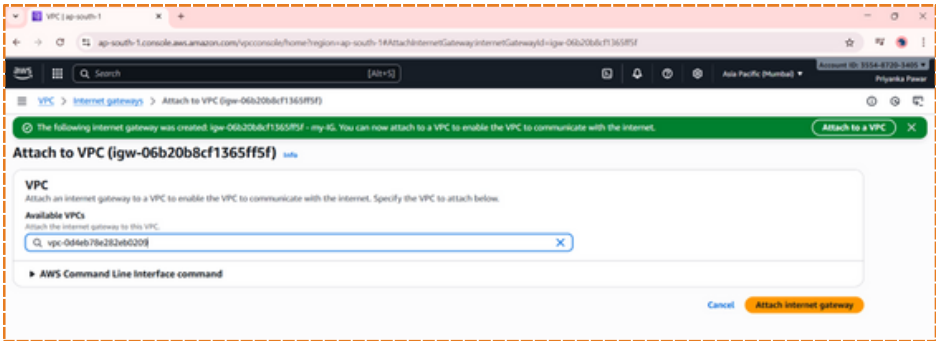


Create internet gateway

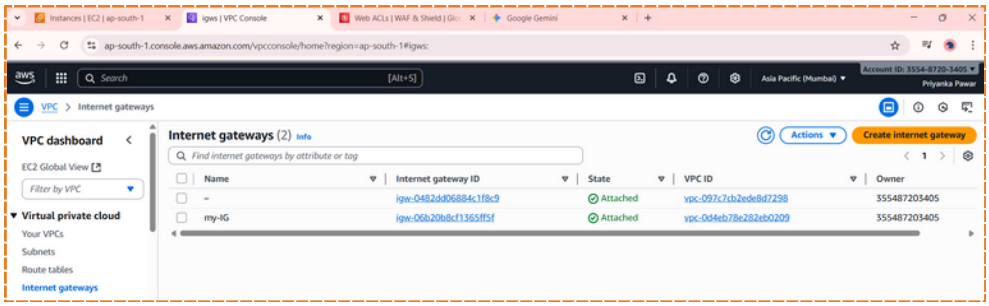
name : my-IG



Attach to VPC

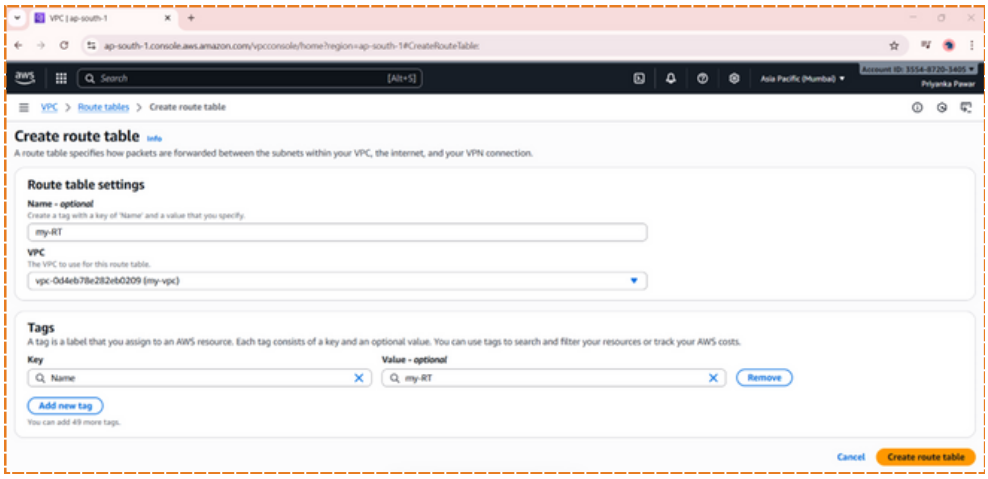


Internet Gateway created

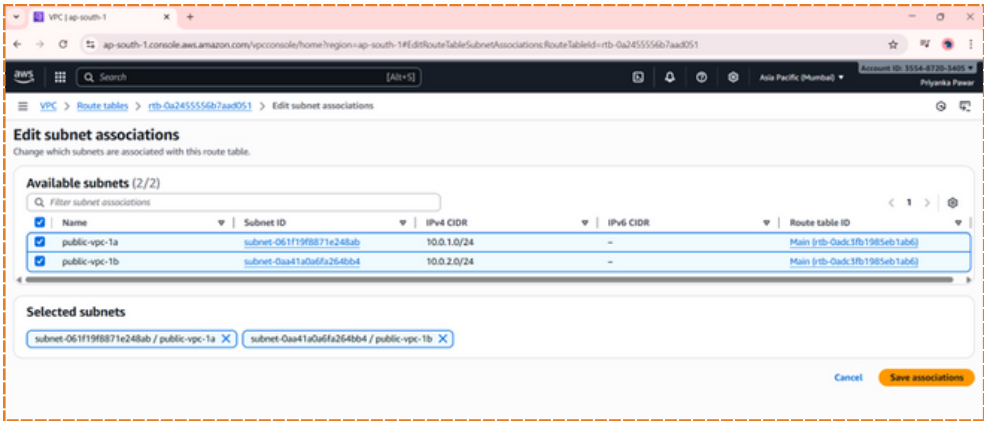


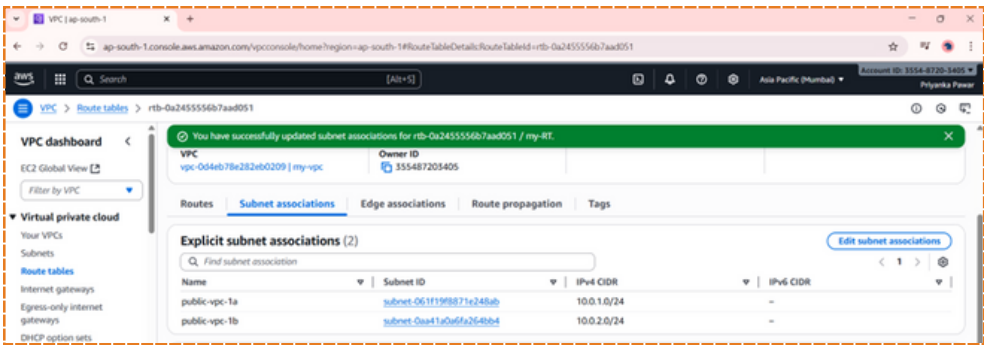
4) Route Table (public)

Route Tables >> Create >> Name: my-RT (VPC = my-vpc)

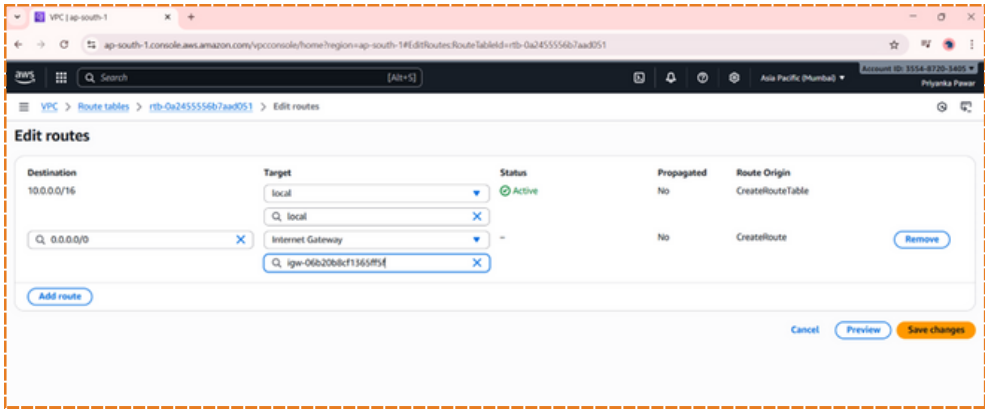


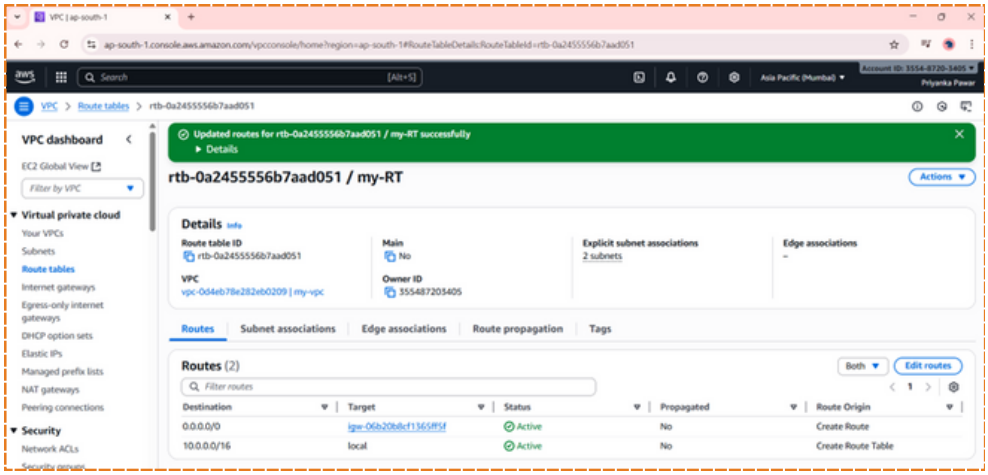
Subnet associations >> Select both public subnets.



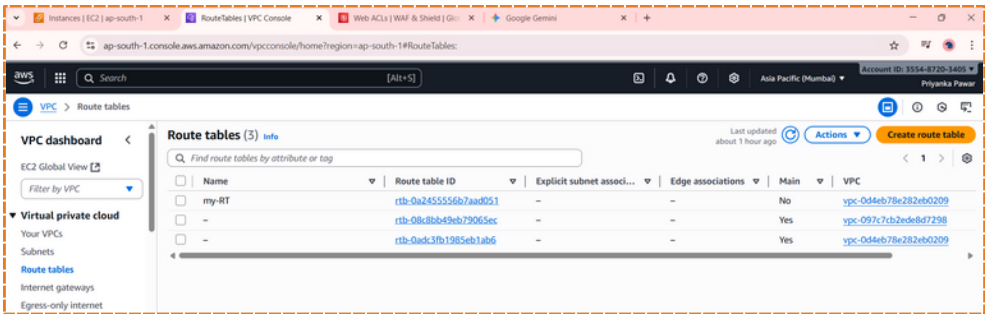


Edit routes → Add 0.0.0.0/0 → Target: Internet Gateway



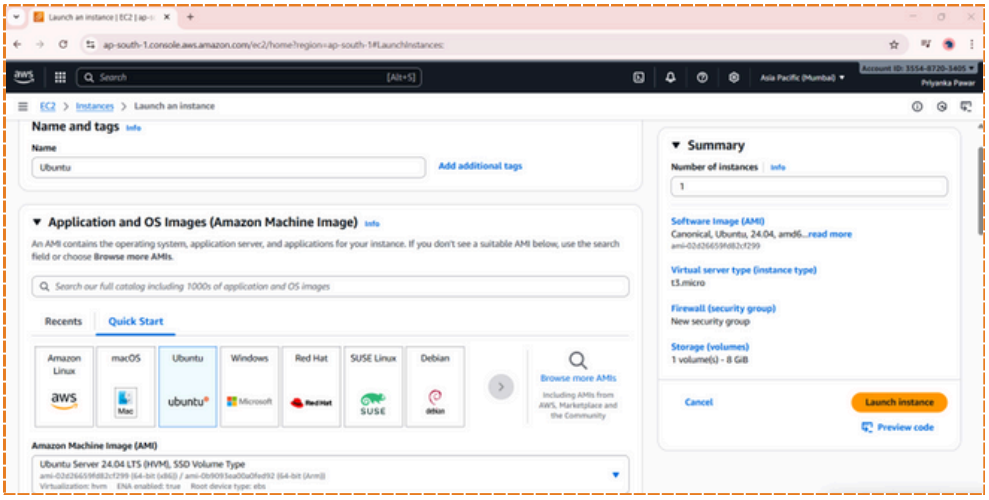


Route table created

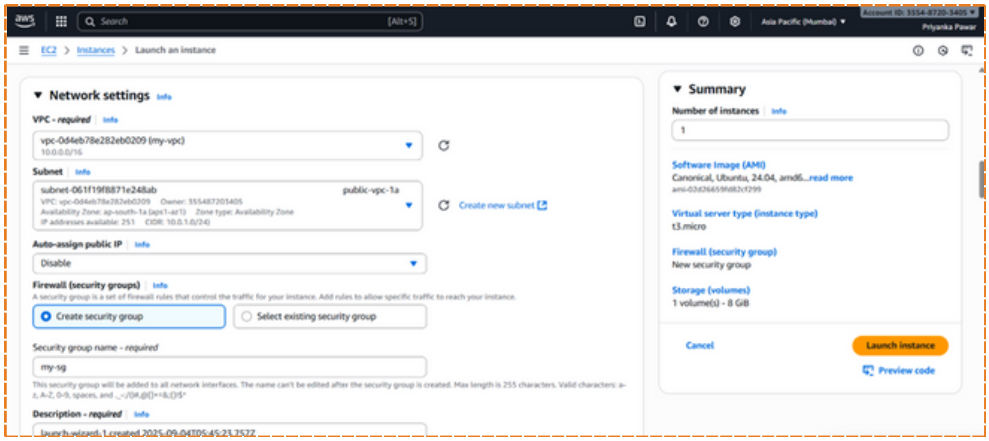


5) Launch EC2 Instances (web servers)

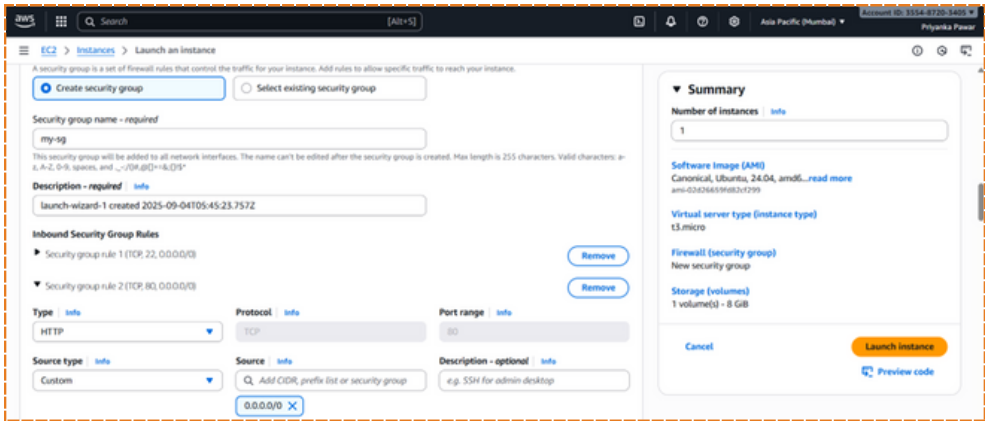
EC2 → Launch instance (Ubuntu).



Network Settings :
VPC : (my-vpc)
Subnet : public-vpc-1a
Create security group
name : my-sg

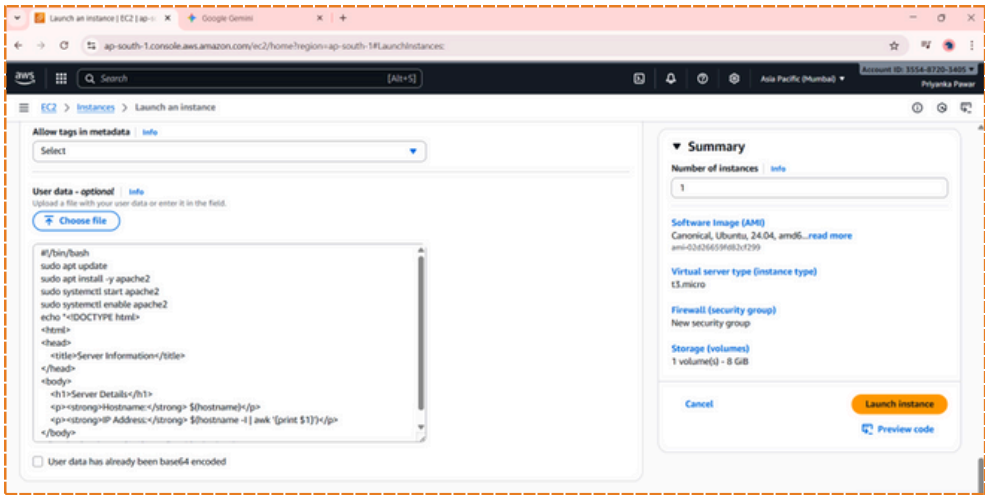


Inbound Security Group add Type : HTTP
Source Type : Custom
Source : 0.0.0.0/0



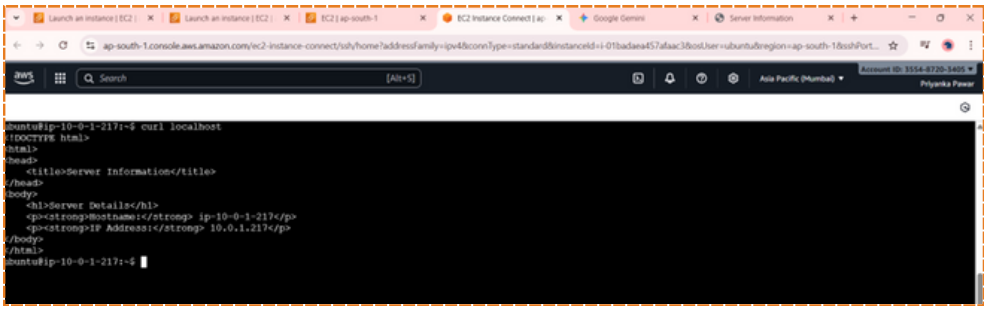
Advanced Settings add these in Execute Shell

```
#!/bin/bash
sudo apt update
sudo apt install -y apache2
sudo systemctl start apache2
sudo systemctl enable apache2
echo "<!DOCTYPE html>
<html>
<head>
  <title>Server Information</title>
</head>
<body>
  <h1>Server Details</h1>
  <p><strong>Hostname:</strong> $(hostname)</p>
  <p><strong>IP Address:</strong> $(hostname -I | awk '{print $1}')</p>
</body>
</html>" | sudo tee /var/www/html/index.html
```



The end result is a running web server (Apache) that hosts a simple, single-page website. This website, which can be accessed from any web browser, displays the hostname and IP address of the server it's running on.

Connect to instance and do curl localhost

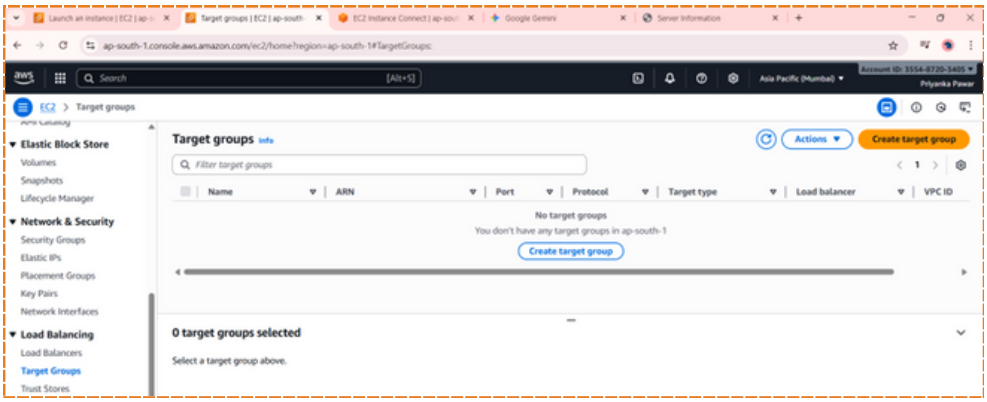


Copy instance publlic IP and past it in browser

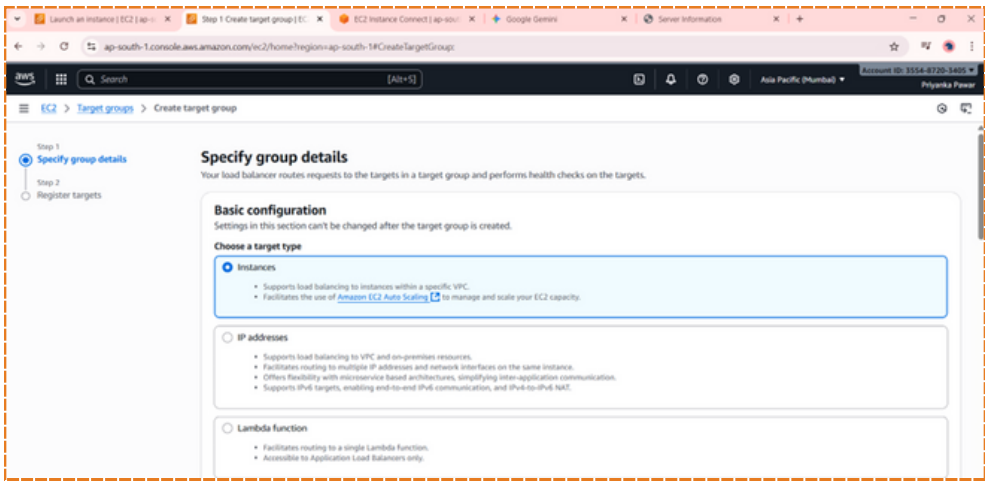
http://13.232.75.159

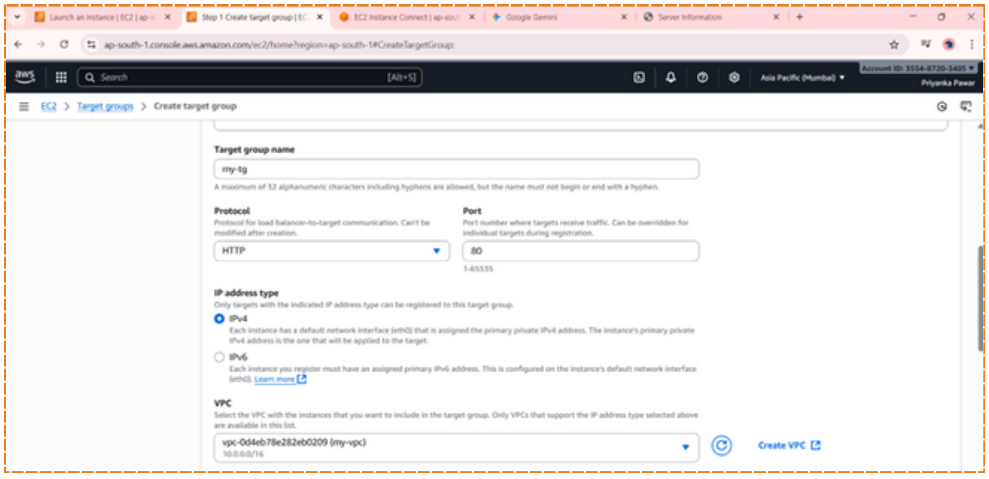
6) Target Group (for ALB)

EC2 >>Load Balancing >> Target Groups >> Create

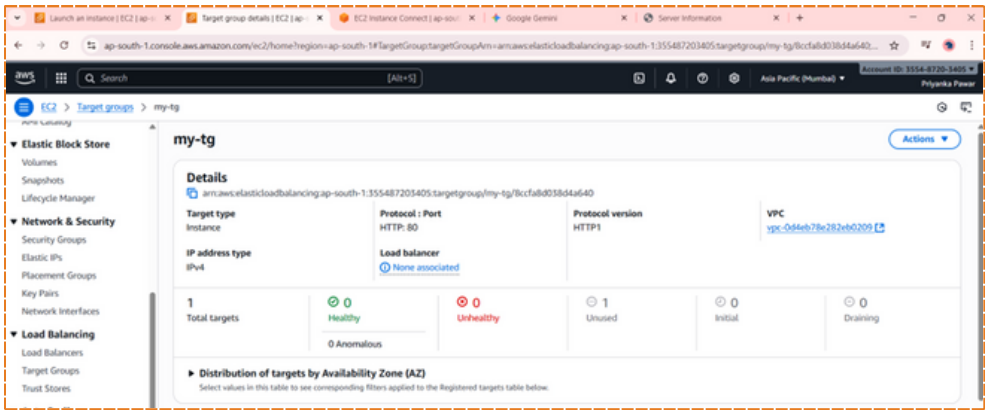


Type: Instances, Protocol: HTTP:80, VPC: my-vpc, Health check: /



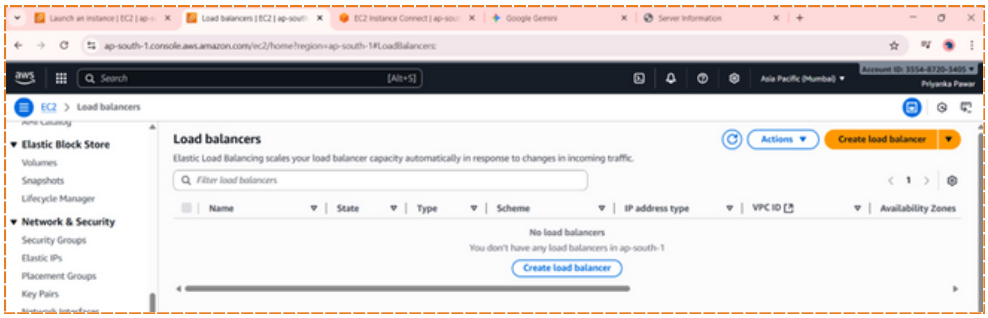


Health check

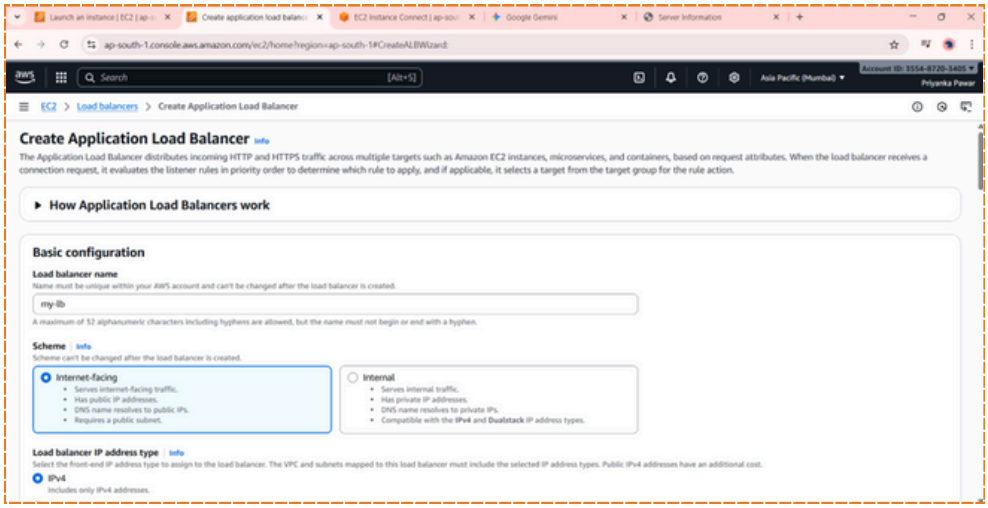


7) Application Load Balancer (ALB)

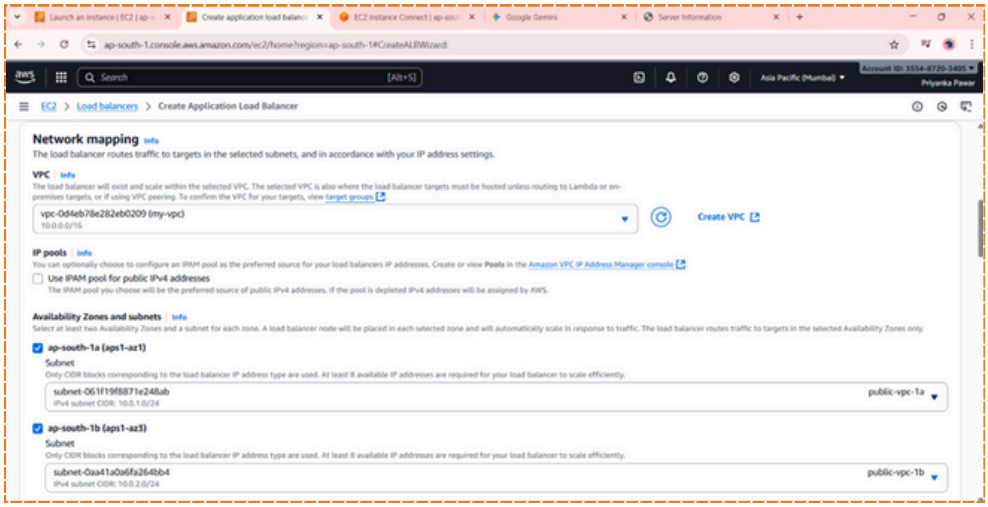
EC2 → Load Balancers → Create Load Balancer → Application LB



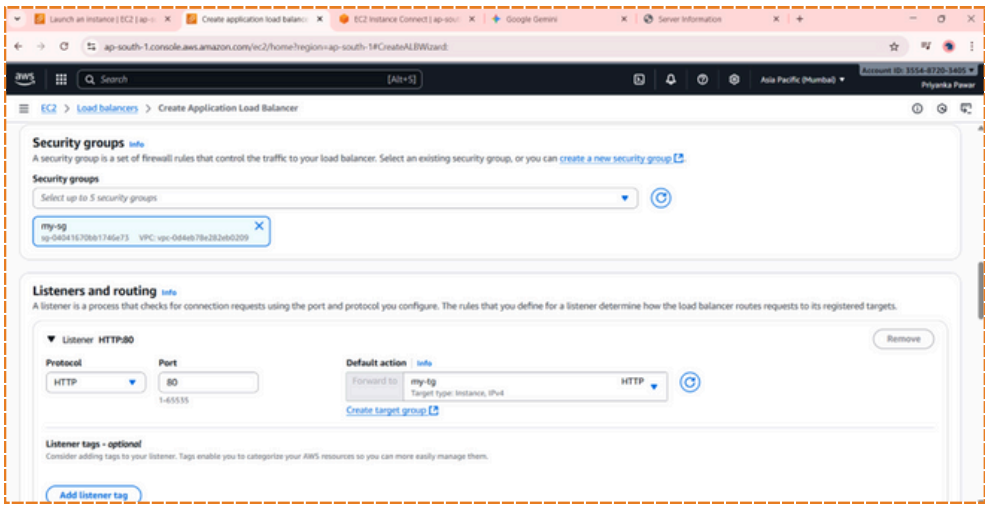
Scheme: Internet-facing, IP type: IPv4



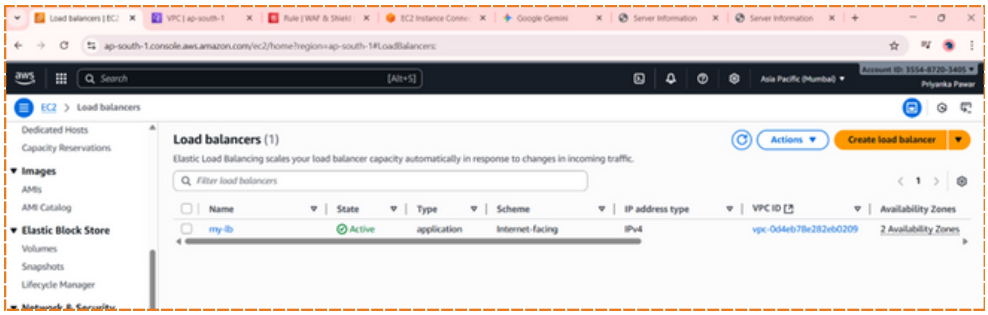
Subnets: select both public subnets

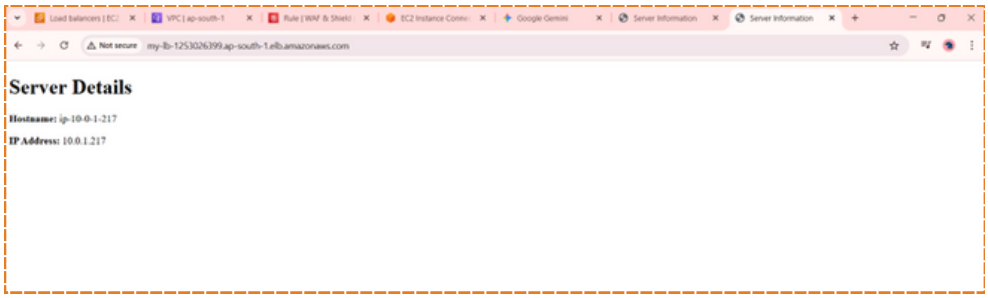


Security group: my-sg
Listener: HTTP:80 → Forward to your Target Group



After creation, hit the ALB DNS name in a browser.

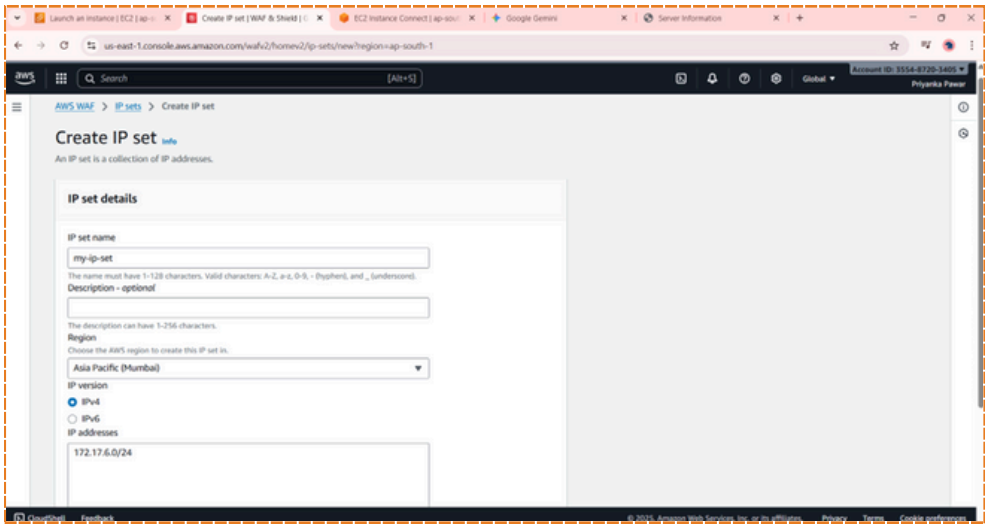




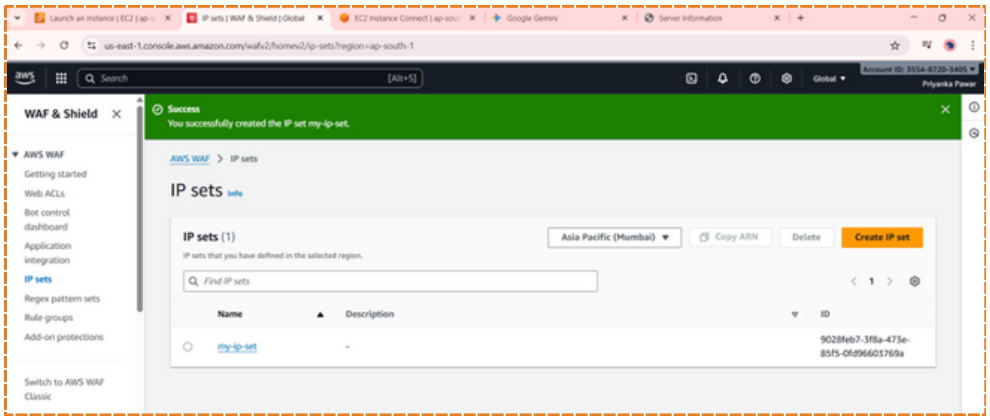
8) AWS WAF (protect the ALB)

IP sets >> Create IP set

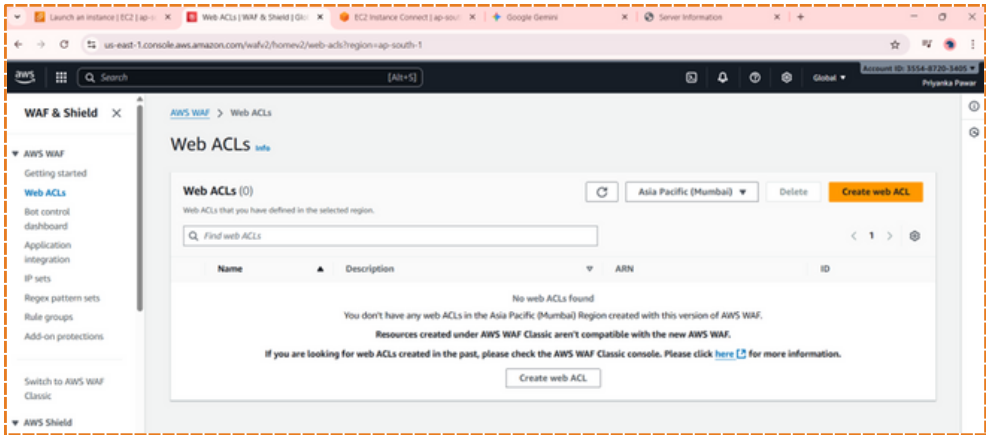
name : my-ip-set
Region : Asia Pacific (Mumbai)
IP Version : IPv4
IP address : 172.17.6.0/24 {this ip will block}



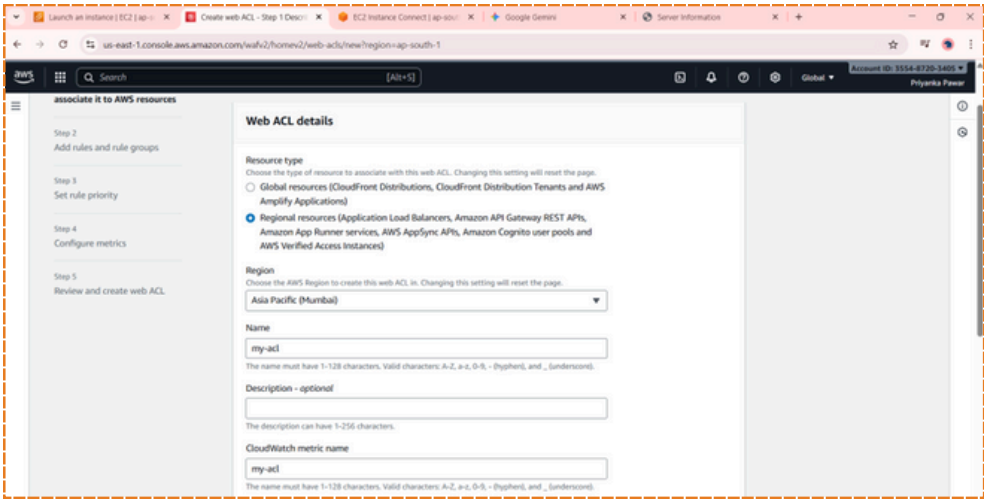
IP set Created



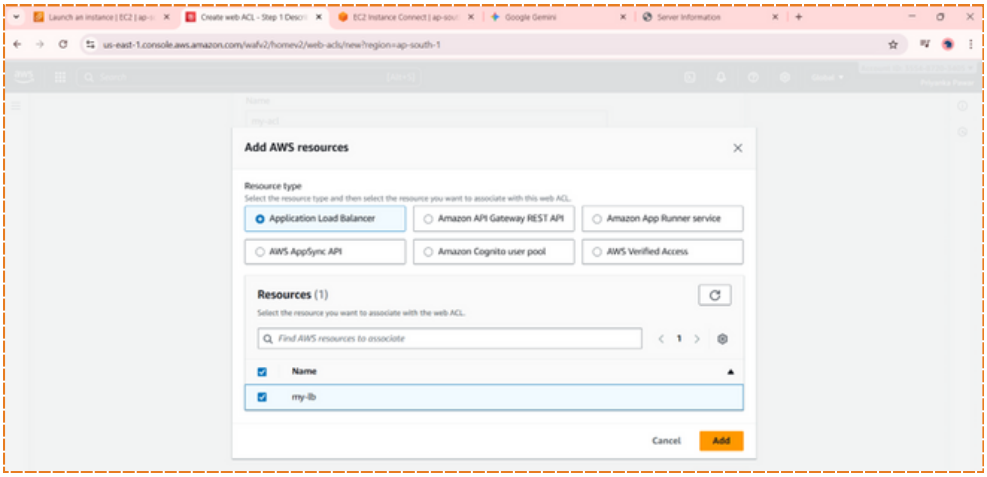
AWS WAF → Web ACLs → Create



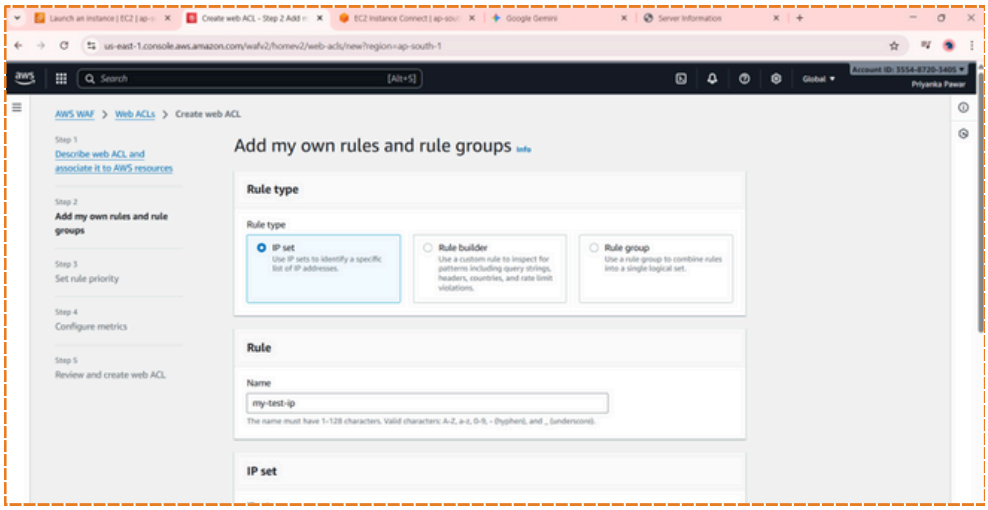
Scope: REGIONAL (same Region as ALB)
Default action: Allow



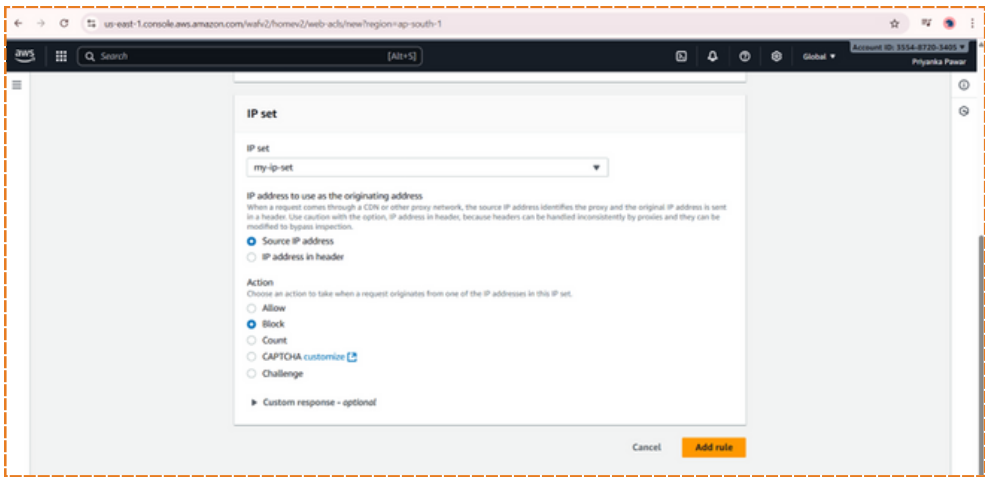
Add AWS resources
Resource type = Application Load Balancer



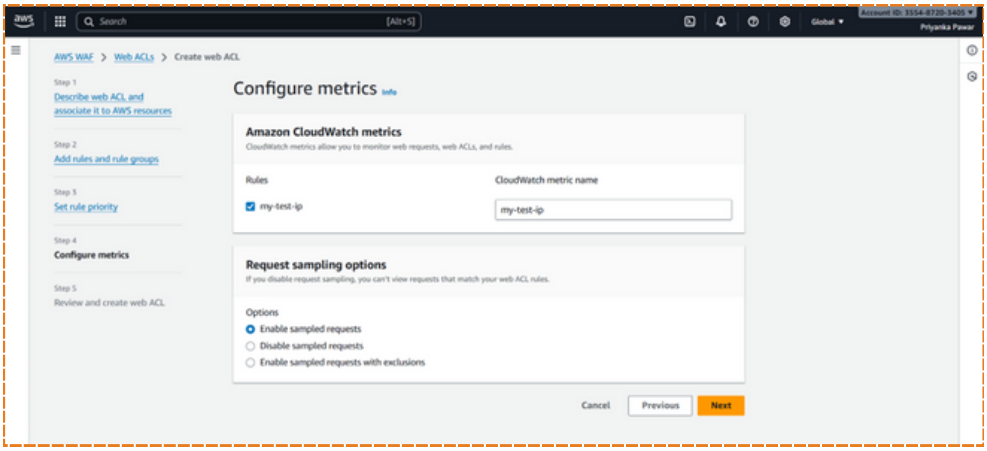
Add IP set



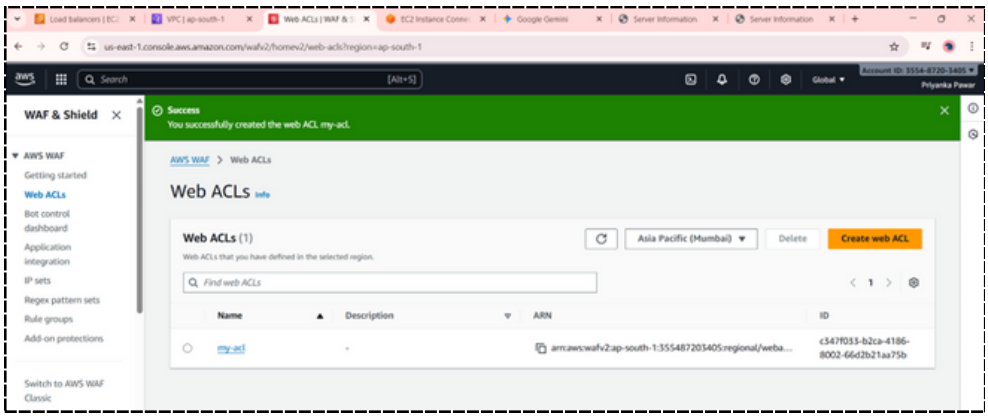
action: Block
Add rule



Configure metrics
Enable sampled requests



WAF tuning: start rules in Count mode, review logs, then switch to Block.



Conclusion

Building a secure and scalable application on AWS involves carefully designing a VPC, configuring public and private subnets, setting up an ALB for load distribution, and adding AWS WAF to protect against threats. This architecture is production-grade, offering security, high availability, and scalability.

When properly tuned, it minimizes downtime and protects applications from common vulnerabilities, making it ideal for businesses that need a secure, internet-facing application environment.


Thank
You