

AWS Project: Secure Multi-AZ Web Application with ALB, WAF, and SSL

Designed and implemented a highly available, secure, and scalable web application architecture on AWS using VPC, subnets, NAT Gateway, Application Load Balancer (ALB), WAF, Route 53, and SSL certificates. The architecture ensures that only traffic from the ALB can access private EC2 instances, and security controls are applied using WAF rules.

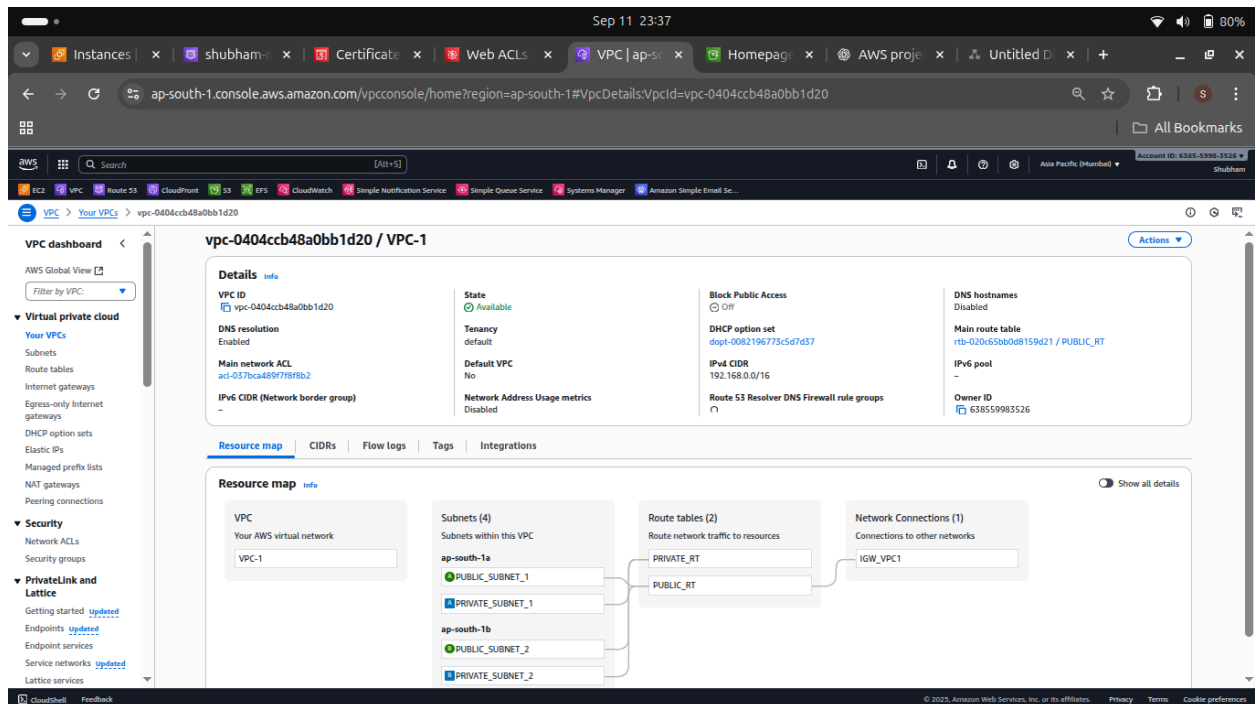
Tools & Services Used

- **AWS VPC**
- **EC2 (Elastic Compute Cloud)**
- **NAT Gateway**
- **Application Load Balancer (ALB)**
- **Target Groups**
- **AWS WAF**
- **Route 53**

- **AWS Certificate Manager (ACM)**
- **CloudWatch Logs (optional monitoring)**
- **Linux instances for web server setup**

STEP 1

Created VPC



STEP 2

Created two public subnets

Sep 11 21:47

Instances | EC2 | ap-south-1 x VPC | ap-south-1 x Distributions | CloudFront x Homepage | S3 | ap-south-1 x +

ap-south-1.console.aws.amazon.com/vpconsole/home?region=ap-south-1#CreateSubnet:

aws Search [Alt+S] Asia Pacific (Mumbai) Account ID: 6385-5998-3526 Shubham

EC2 VPC Route 53 CloudFront S3 EFS CloudWatch Simple Notification Service Simple Queue Service Systems Manager Amazon Simple Email Se...

VPC > Subnets > Create subnet

Specify the CIDR blocks and Availability Zone for the subnet.

Subnet 1 of 1

Subnet name
Create a tag with a key of 'Name' and a value that you specify.
PUBLIC_SUBNET_1
The name can be up to 256 characters long.

Availability Zone [Info](#)
Choose the zone in which your subnet will reside, or let Amazon choose one for you.
Asia Pacific (Mumbai) / ap-south-1a

IPv4 VPC CIDR block [Info](#)
Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.
192.168.0.0/16

IPv4 subnet CIDR block
192.168.0.0/24 256 IPs

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Sep 11 21:47

Instances | EC2 | ap-south-1 x VPC | ap-south-1 x Distributions | CloudFront x Homepage | S3 | ap-south-1 x +

ap-south-1.console.aws.amazon.com/vpconsole/home?region=ap-south-1#CreateSubnet:

aws Search [Alt+S] Asia Pacific (Mumbai) Account ID: 6385-5998-3526 Shubham

EC2 VPC Route 53 CloudFront S3 EFS CloudWatch Simple Notification Service Simple Queue Service Systems Manager Amazon Simple Email Se...

VPC > Subnets > Create subnet

Specify the CIDR blocks and Availability zone for the subnet.

Subnet 1 of 1

Subnet name
Create a tag with a key of 'Name' and a value that you specify.
PUBLIC_SUBNET_2
The name can be up to 256 characters long.

Availability Zone [Info](#)
Choose the zone in which your subnet will reside, or let Amazon choose one for you.
Asia Pacific (Mumbai) / ap-south-1b

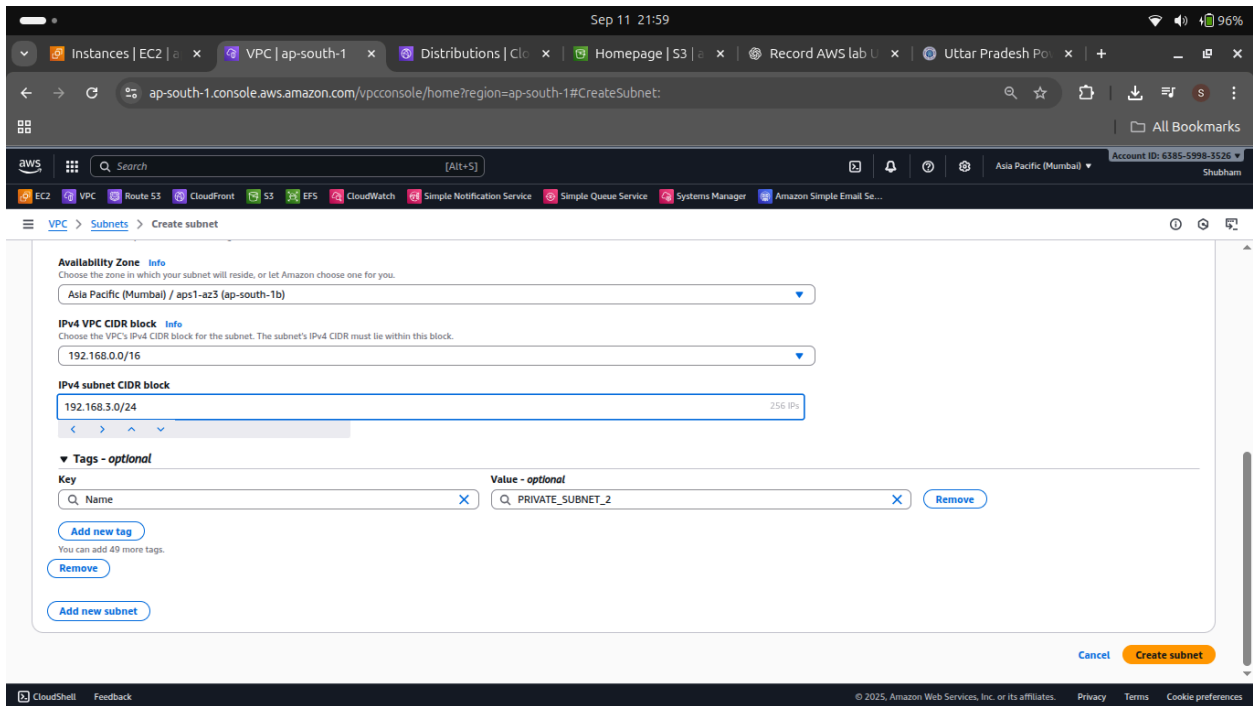
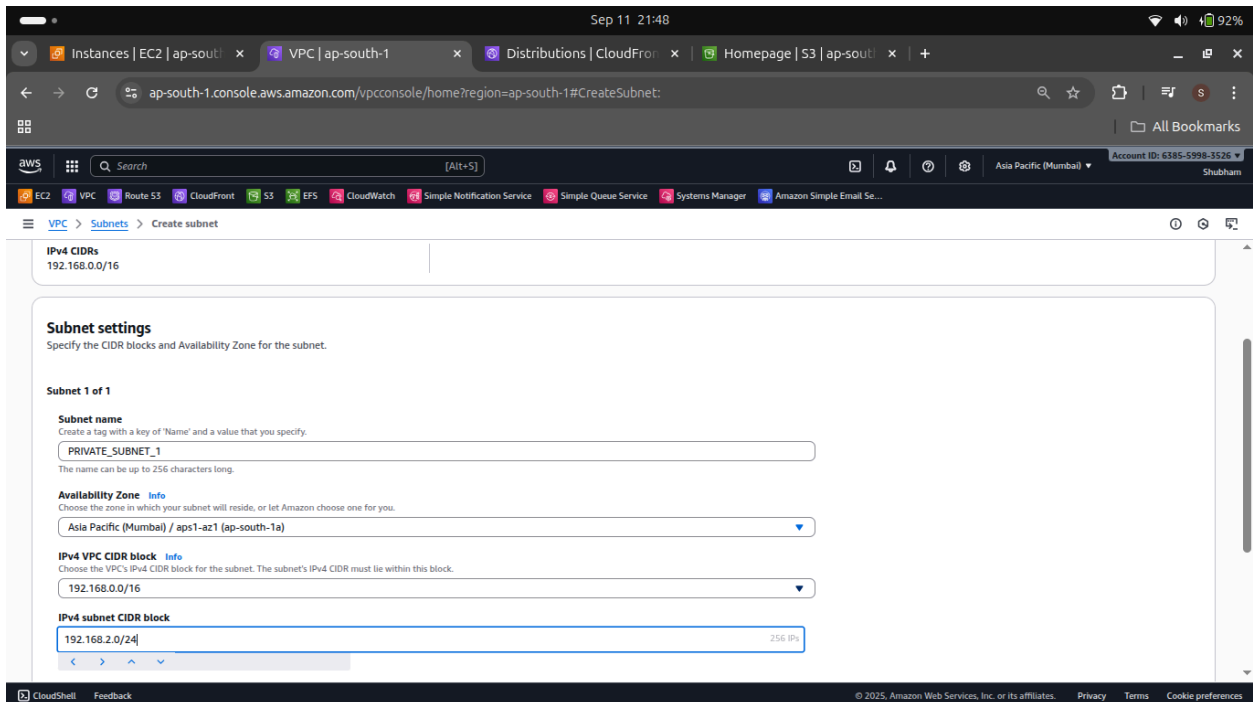
IPv4 VPC CIDR block [Info](#)
Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.
192.168.0.0/16

IPv4 subnet CIDR block
192.168.1.0/24 256 IPs

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

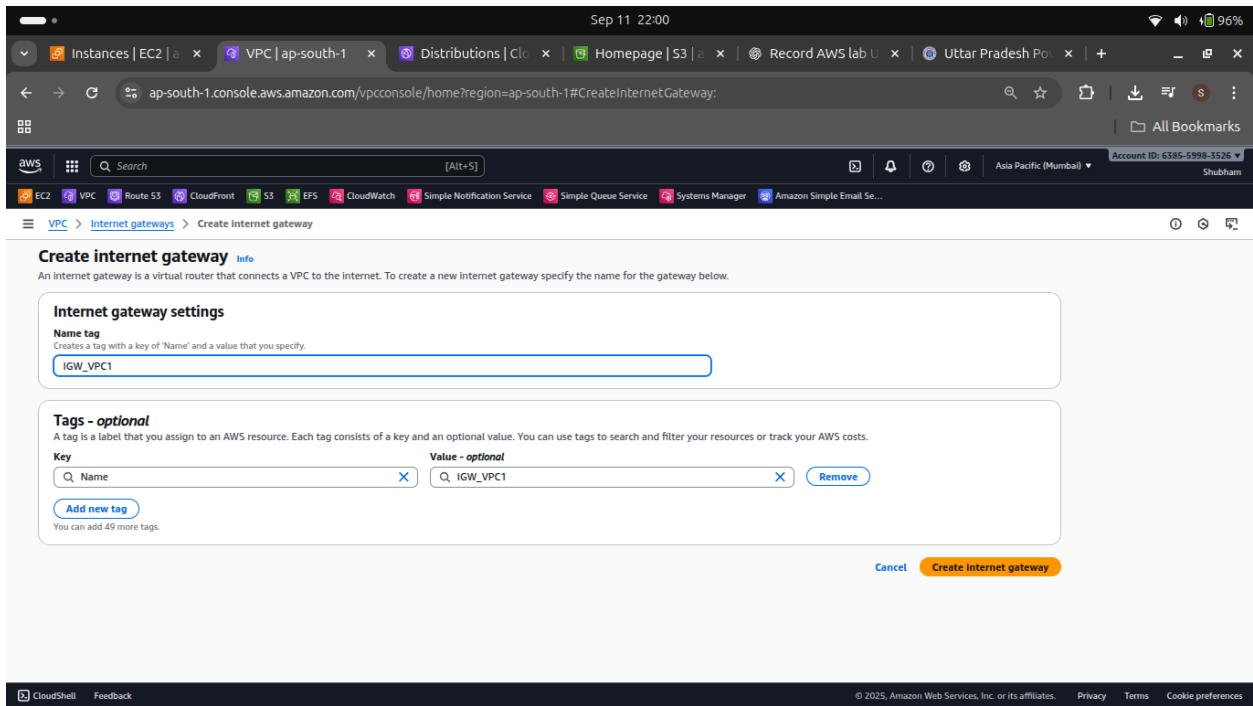
STEP 3

Created two private subnets



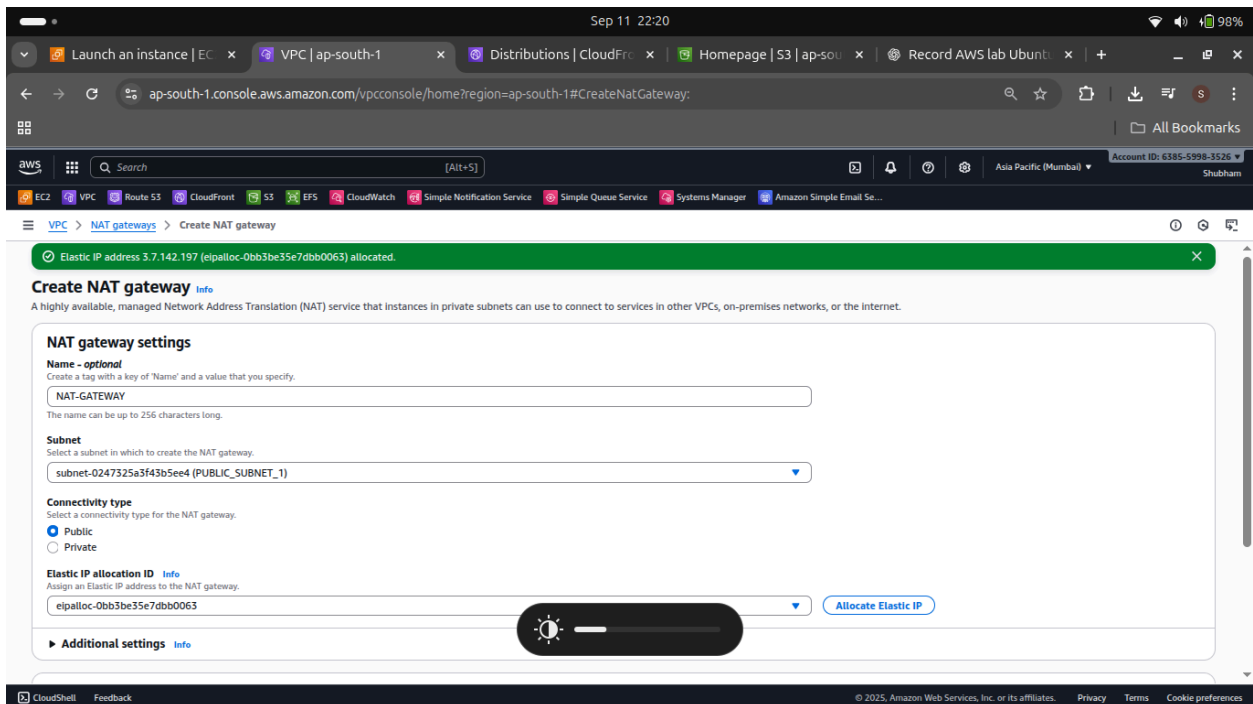
STEP 4

“Created an Internet Gateway and attached it to the VPC to provide internet access to the public subnets.



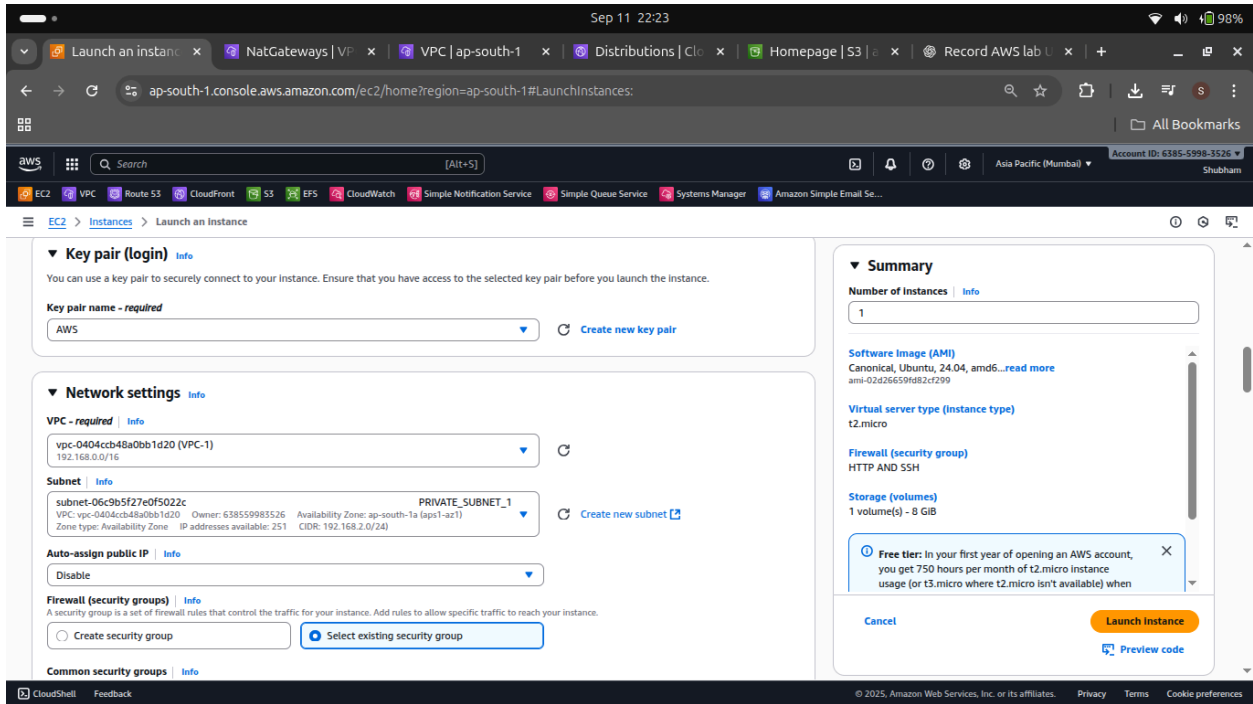
STEP 5

Created NAT gateway to provide outbound internet access to private subnets



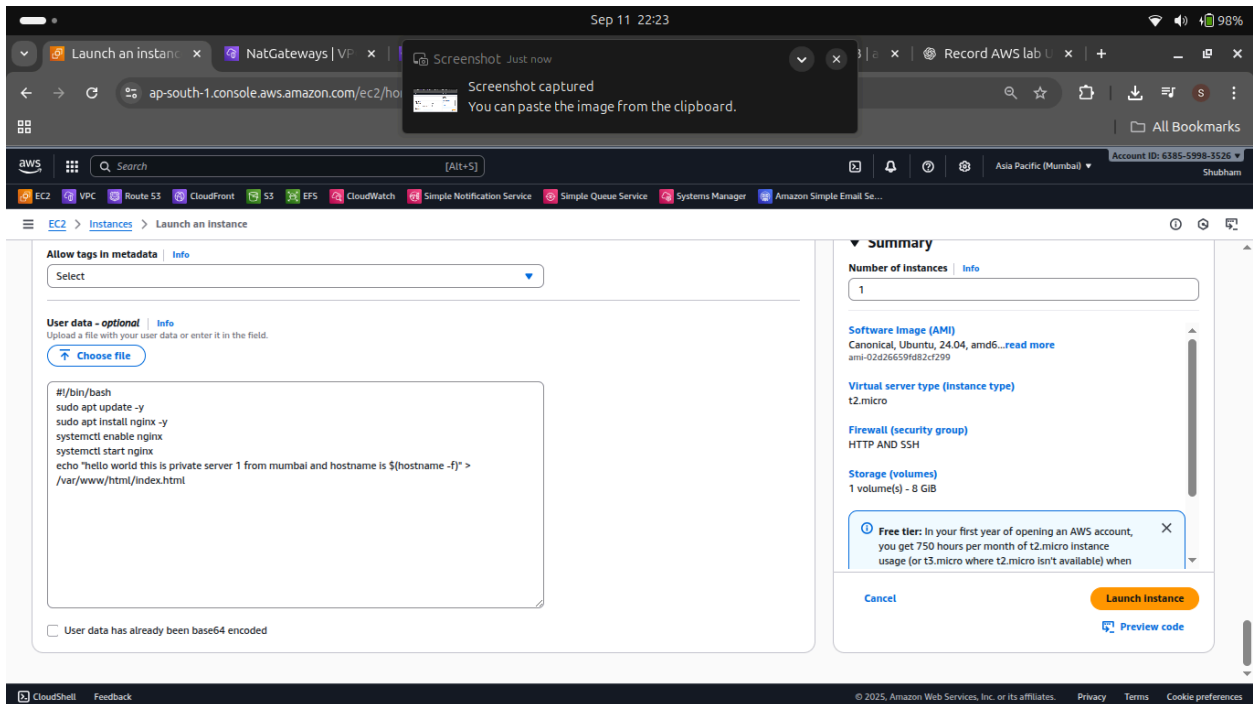
STEP 6

Launched two ec2 instances into private subnets



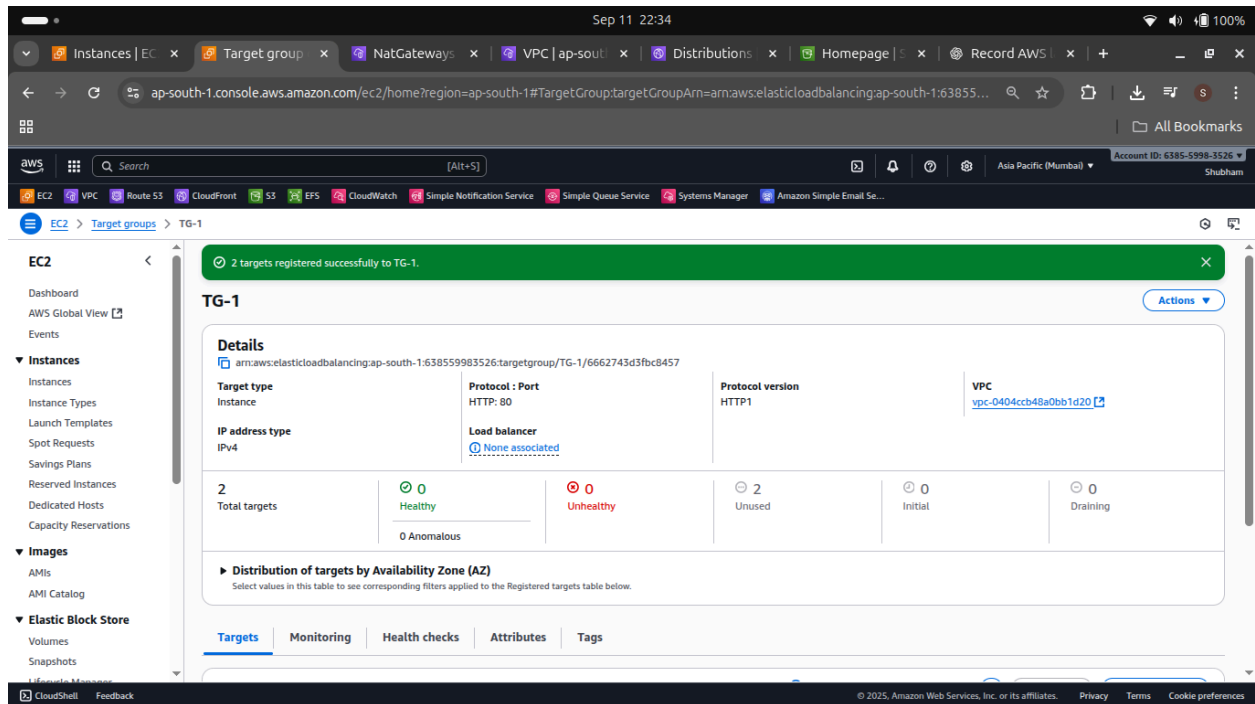
STEP 7

Added user data scripts on both private servers to install nginx web server



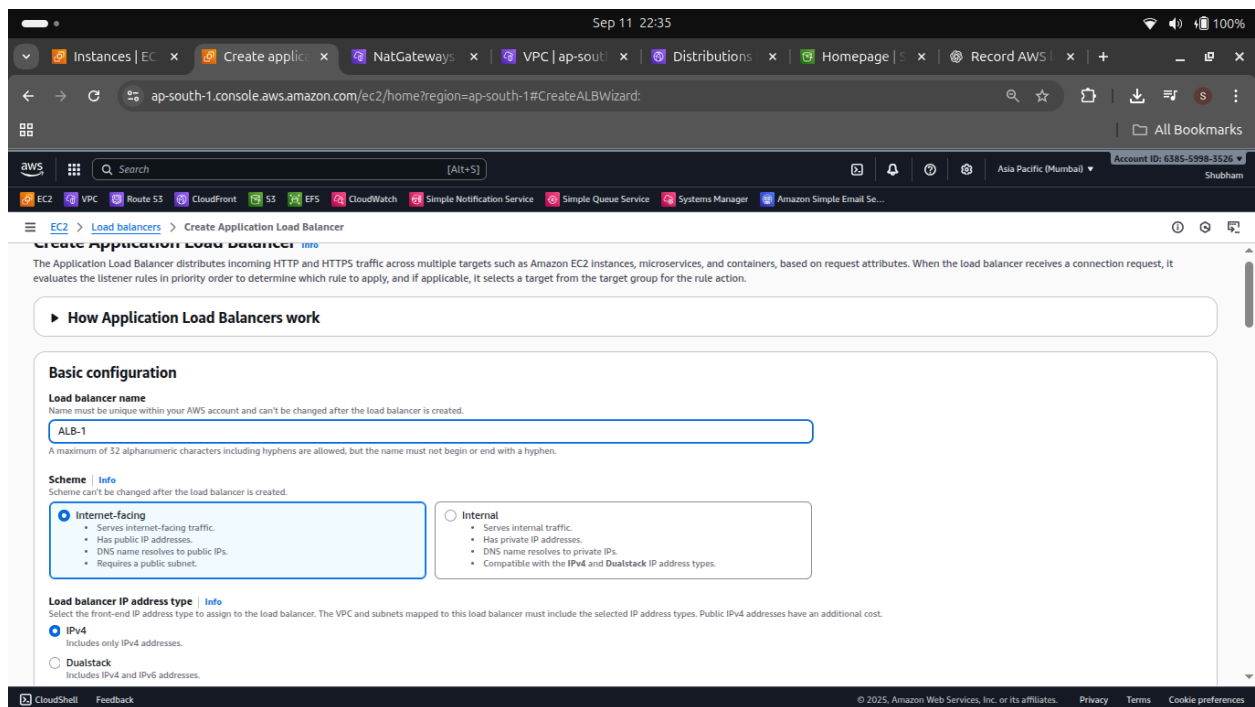
STEP 8

Created Target group



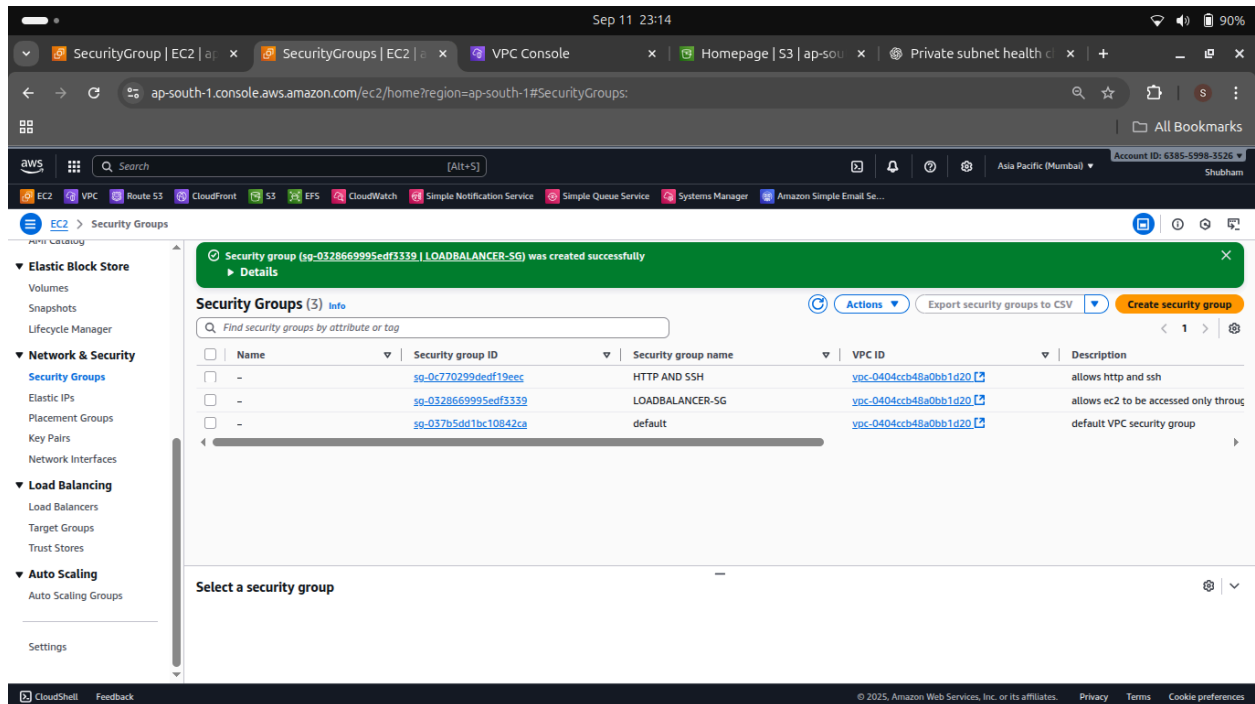
STEP 9

Created Application Load Balancer



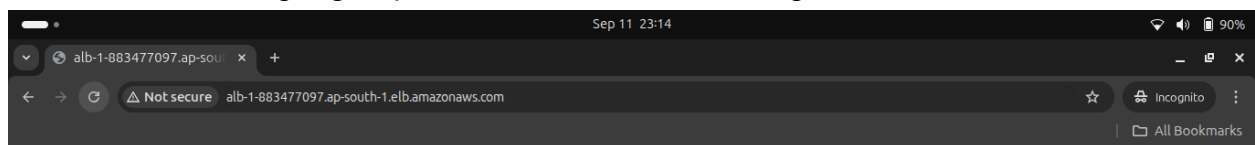
STEP 10

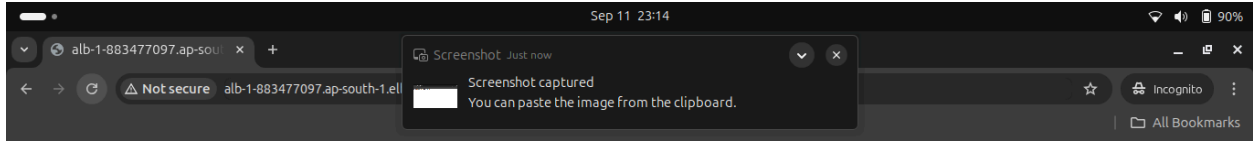
Configured security groups and created two security groups 1 for private instances and 1 for load balancer and only traffic to reach ec2 instances through load balancer and ssh using my ip



STEP 11

After successful target group and load balancer reaching instances can see the result

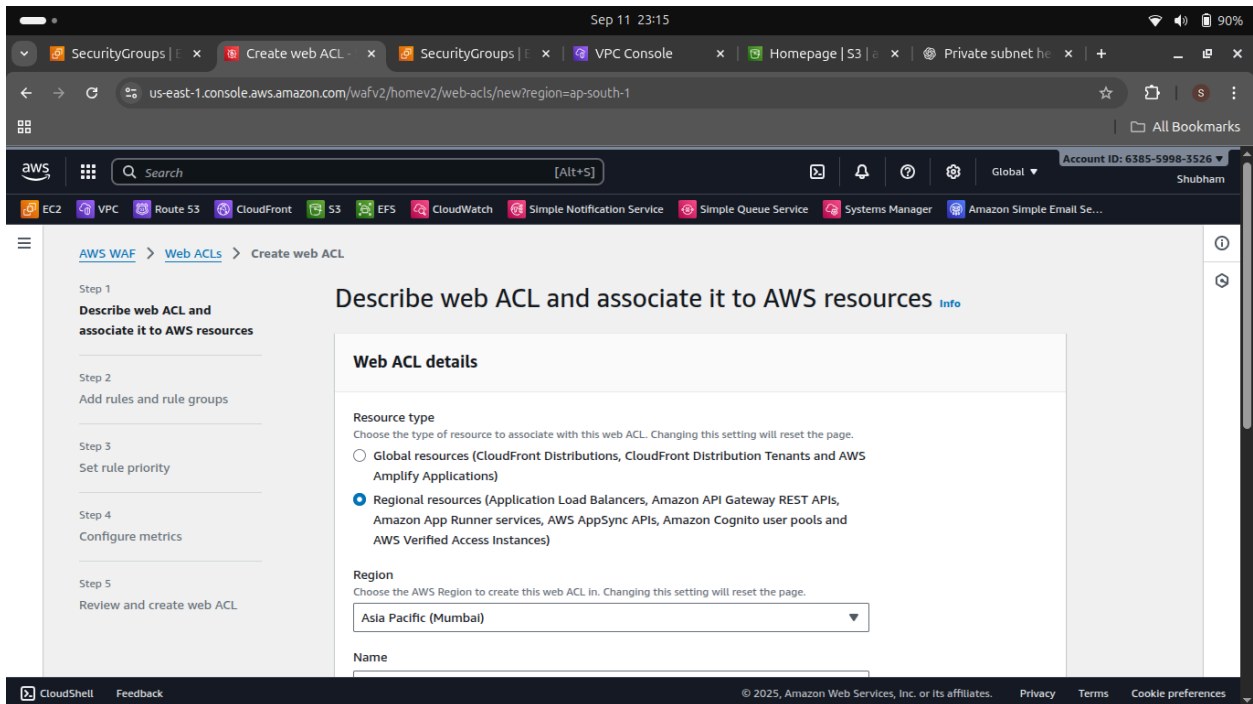


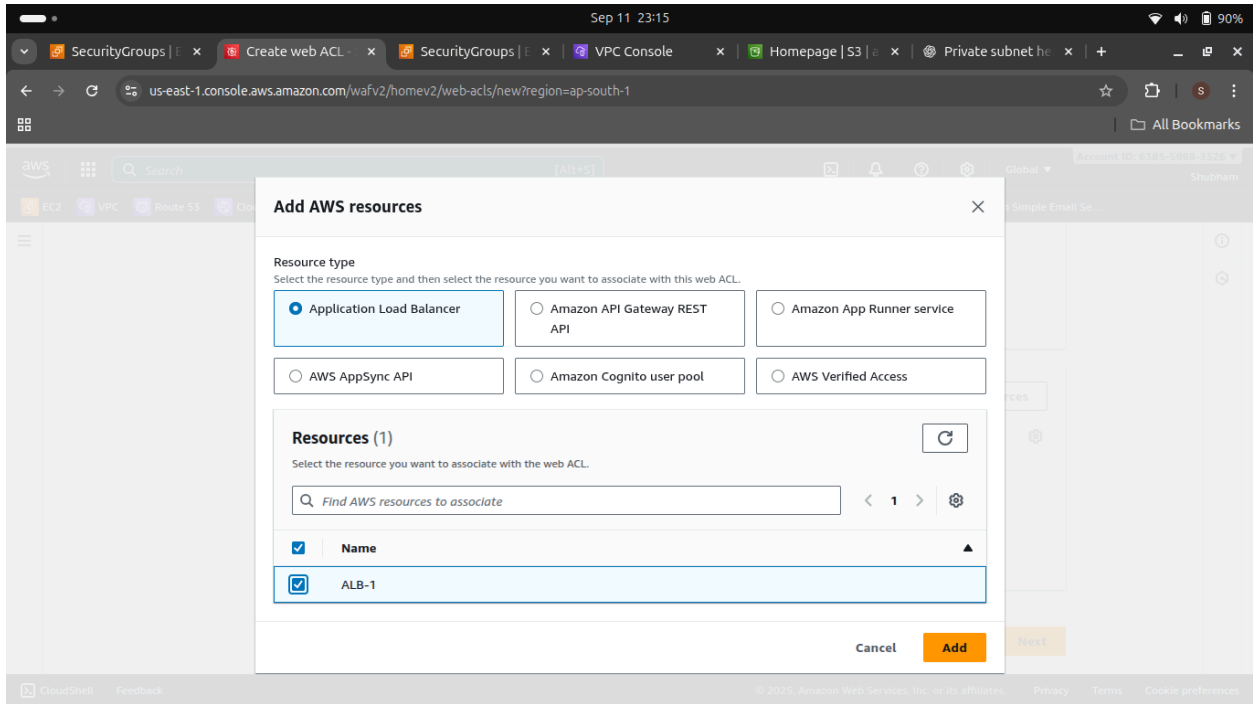


hello world this is private server 2 from mumbai and hostname is ip-192-168-3-175

STEP 12

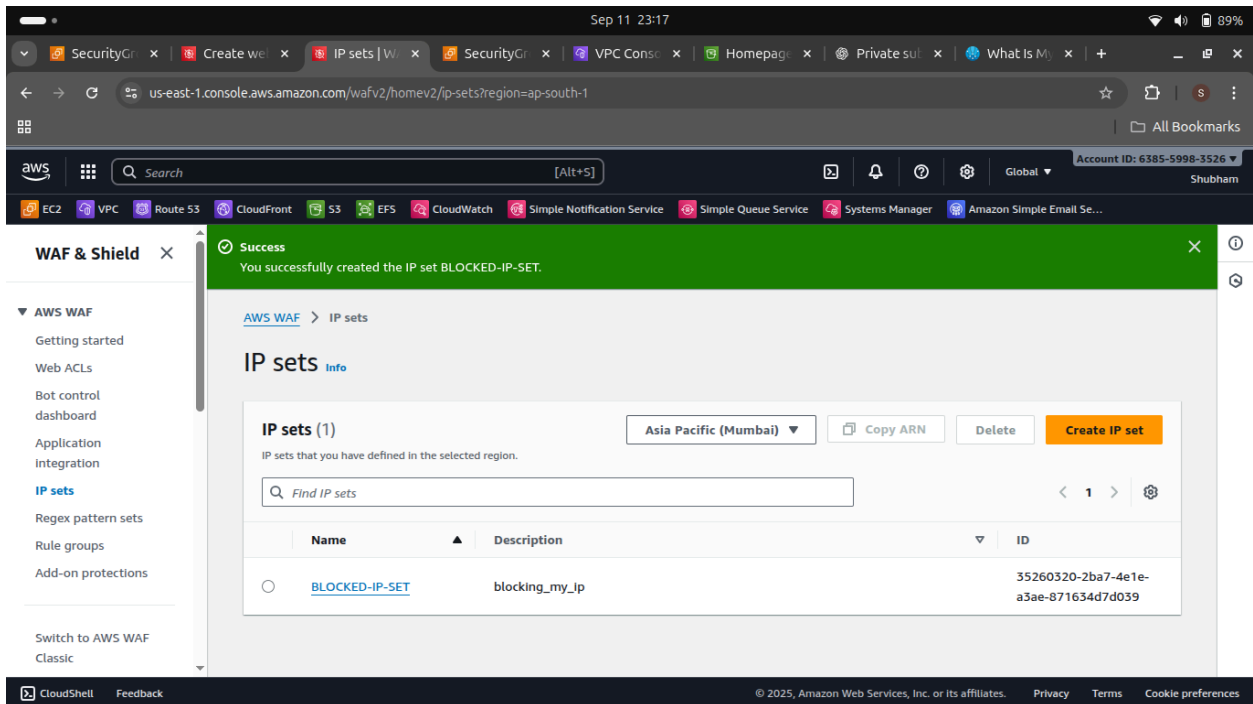
Now creating WAF and attaching to the load balancer

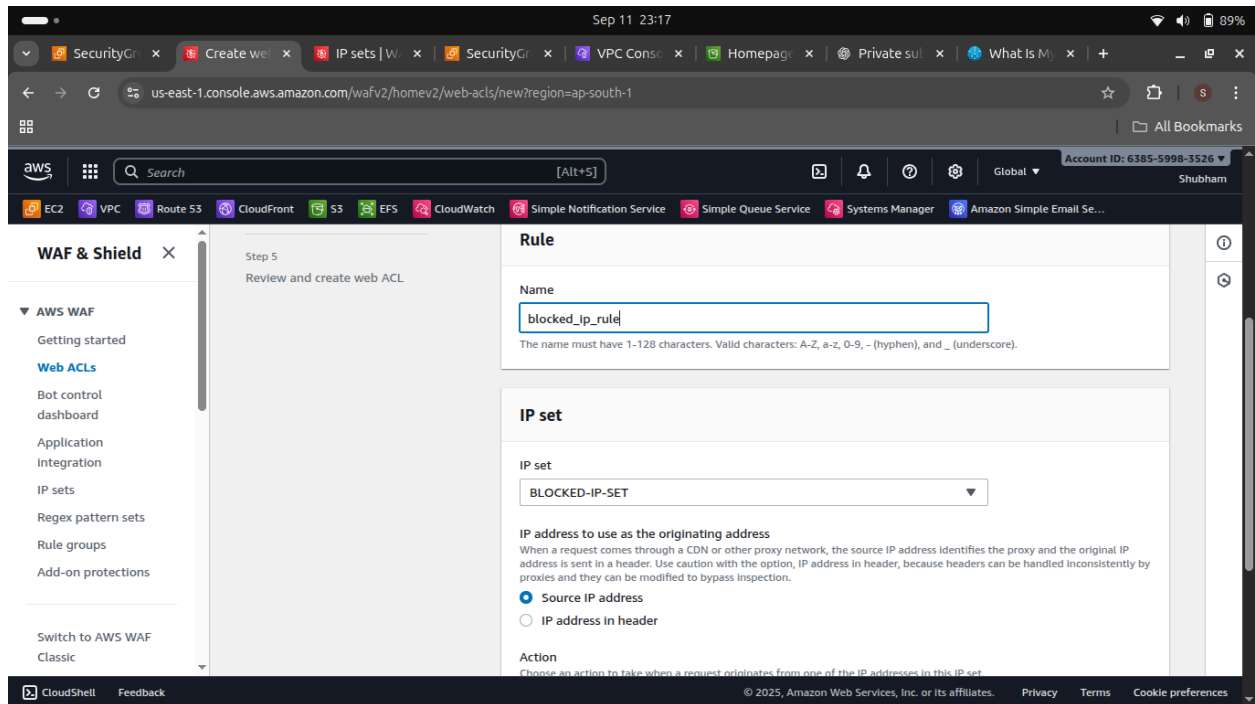




STEP 13

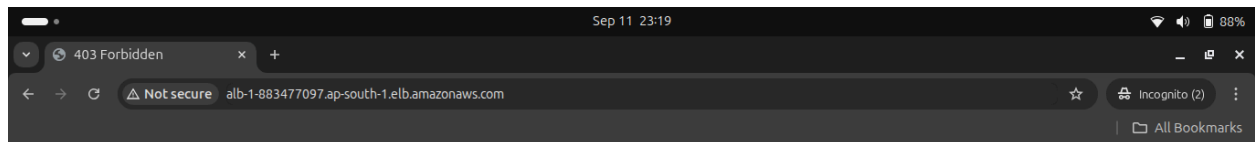
Created IP SET to set blocking from my ip



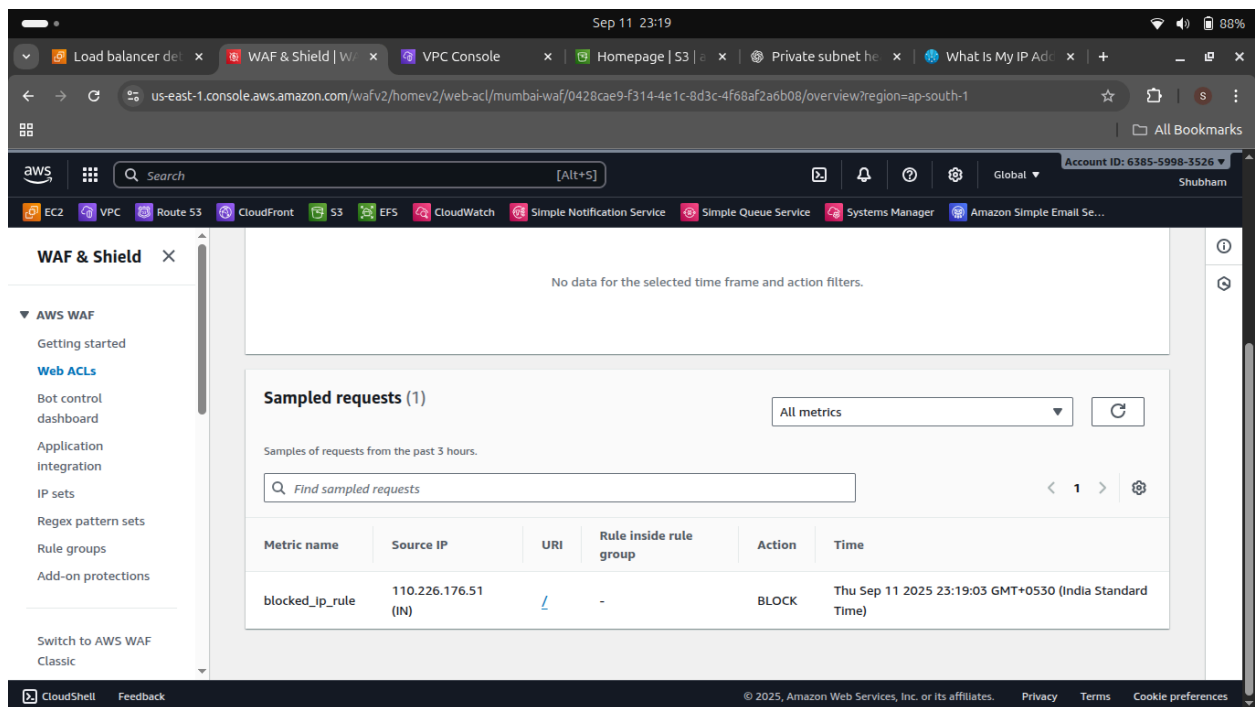


STEP 14

“Verified that the Load Balancer is inaccessible from my IP due to the WAF rules blocking traffic.”

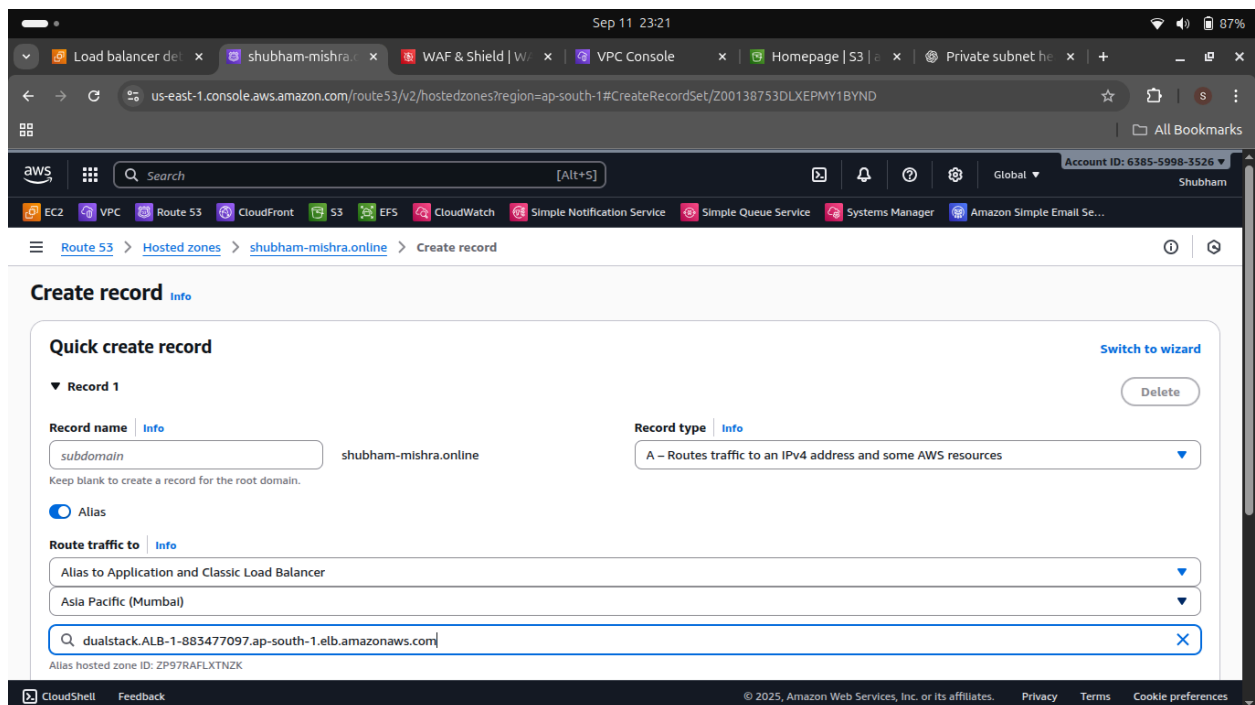


403 Forbidden



STEP 15

Now mapping this loadbalancer to route53 as alias record



STEP 16

Now configuring another rate based rule in waf

The screenshot shows the AWS WAF console interface for editing a rule. The rule name is 'blocked_ip_rule'. The rule type is 'Rate-based rule'. The rate limit is set to 10. The evaluation window is set to 1 minute (60 seconds). The request aggregation is set to Source IP address.

Name
blocked_ip_rule
The name must have 1-128 characters. Valid characters: A-Z, a-z, 0-9, - (hyphen), and _ (underscore).

Type
☐ Regular rule
☒ Rate-based rule
Limits request rates for requests that match your criteria. Applies the action to matching requests when the limit is reached, and removes the action when the rate falls below the limit.

Rate-limiting criteria [Learn more](#)

Rate limit
The maximum number of requests to allow during the specified time window that satisfy your criteria. You can narrow the scope of the requests using a scope-down statement. You can group requests by component types for count aggregation. You must provide at least one aggregation component or a scope-down statement.
10
Rate limit must be between 10 and 2,000,000,000.

Evaluation window
The amount of time to use for request counts.
1 minute (60 seconds)
The default time span is 5 minutes. Valid values are 1, 2, 5, and 10 minutes.

Request aggregation
Select the web request components to use for request aggregation. AWS WAF groups, counts, and rate limits requests based on this criteria.
☒ Source IP address
Use only the IP address from the web request origin. If a web request goes through one or more proxies or load balancers, this will contain the address of the last proxy, and not the originating address of the client.

The screenshot shows the AWS WAF console interface for editing a rule. The rule name is 'blocked_ip_rule'. The rule type is 'Rate-based rule'. The rate limit is set to 10. The evaluation window is set to 1 minute (60 seconds). The request aggregation is set to Source IP address.

Name
blocked_ip_rule
The name must have 1-128 characters. Valid characters: A-Z, a-z, 0-9, - (hyphen), and _ (underscore).

Type
☐ Regular rule
☒ Rate-based rule
Limits request rates for requests that match your criteria. Applies the action to matching requests when the limit is reached, and removes the action when the rate falls below the limit.

Rate-limiting criteria [Learn more](#)

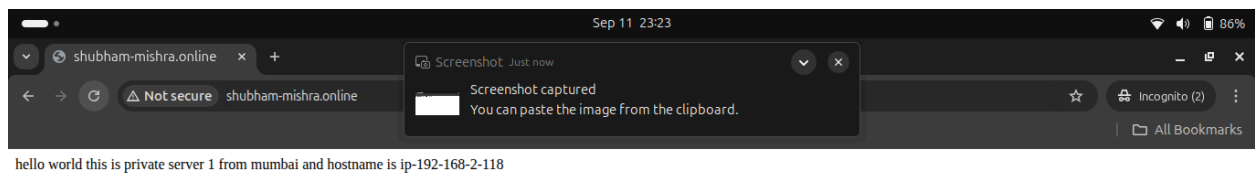
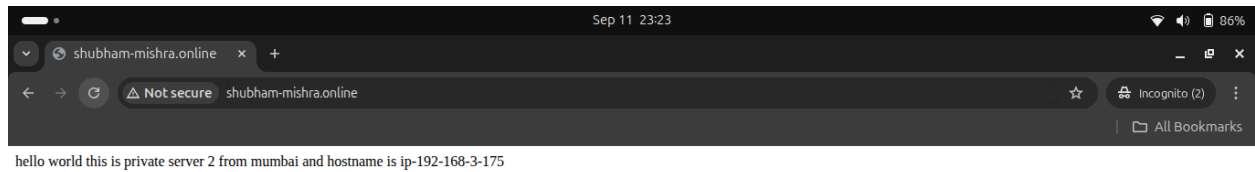
Rate limit
The maximum number of requests to allow during the specified time window that satisfy your criteria. You can narrow the scope of the requests using a scope-down statement. You can group requests by component types for count aggregation. You must provide at least one aggregation component or a scope-down statement.
10
Rate limit must be between 10 and 2,000,000,000.

Evaluation window
The amount of time to use for request counts.
1 minute (60 seconds)
The default time span is 5 minutes. Valid values are 1, 2, 5, and 10 minutes.

Request aggregation
Select the web request components to use for request aggregation. AWS WAF groups, counts, and rate limits requests based on this criteria.
☒ Source IP address
Use only the IP address from the web request origin. If a web request goes through one or more proxies or load balancers, this will contain the address of the last proxy, and not the originating address of the client.

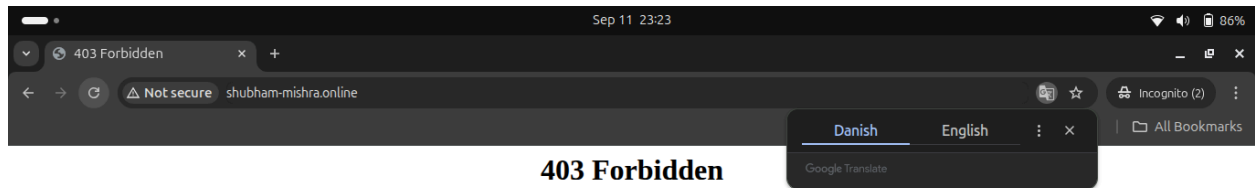
STEP 17

Now again i can access my webserver behind load balancer through my domain shubham-mishra.online



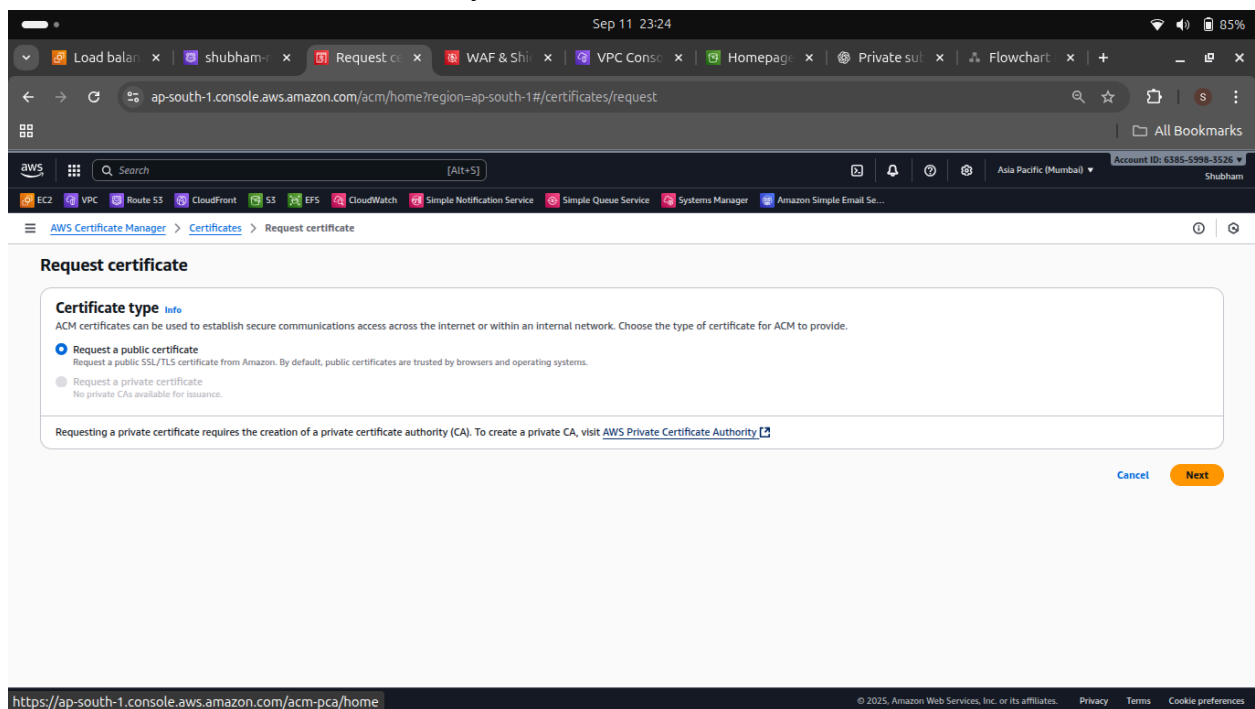
STEP 18

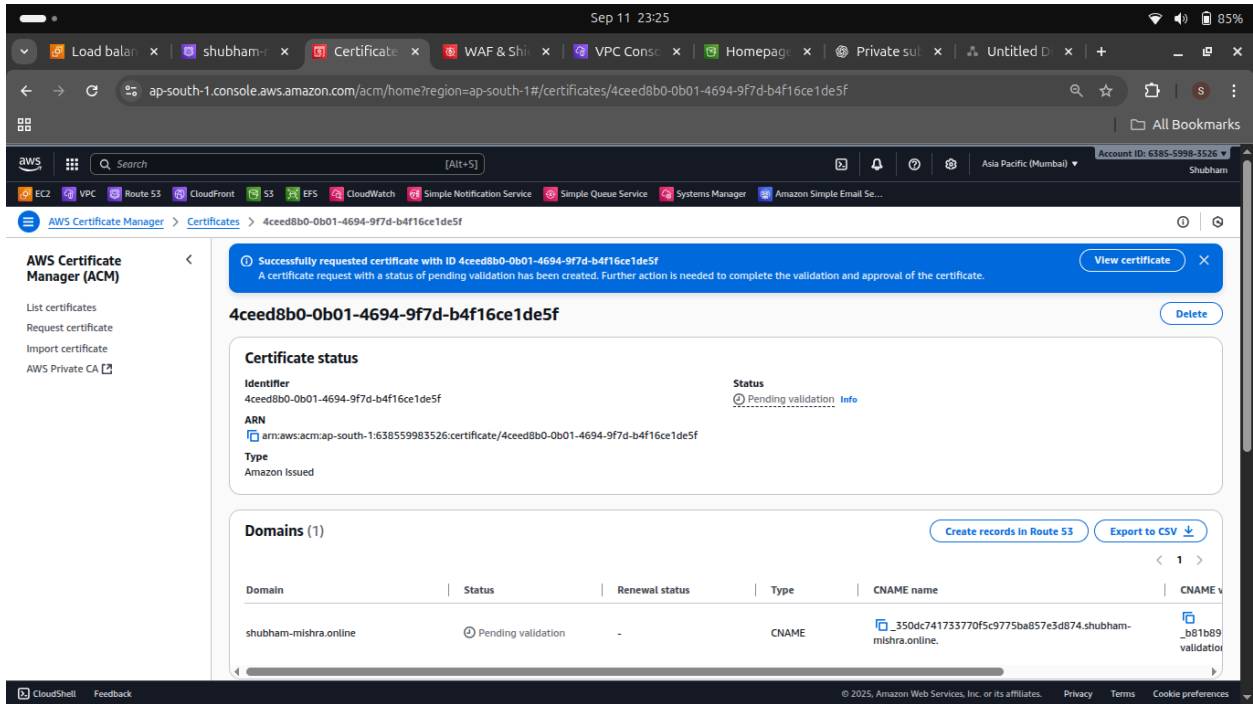
Now again after hitting rate limit of 10 request in 60 sec WAF blocked access



STEP 19

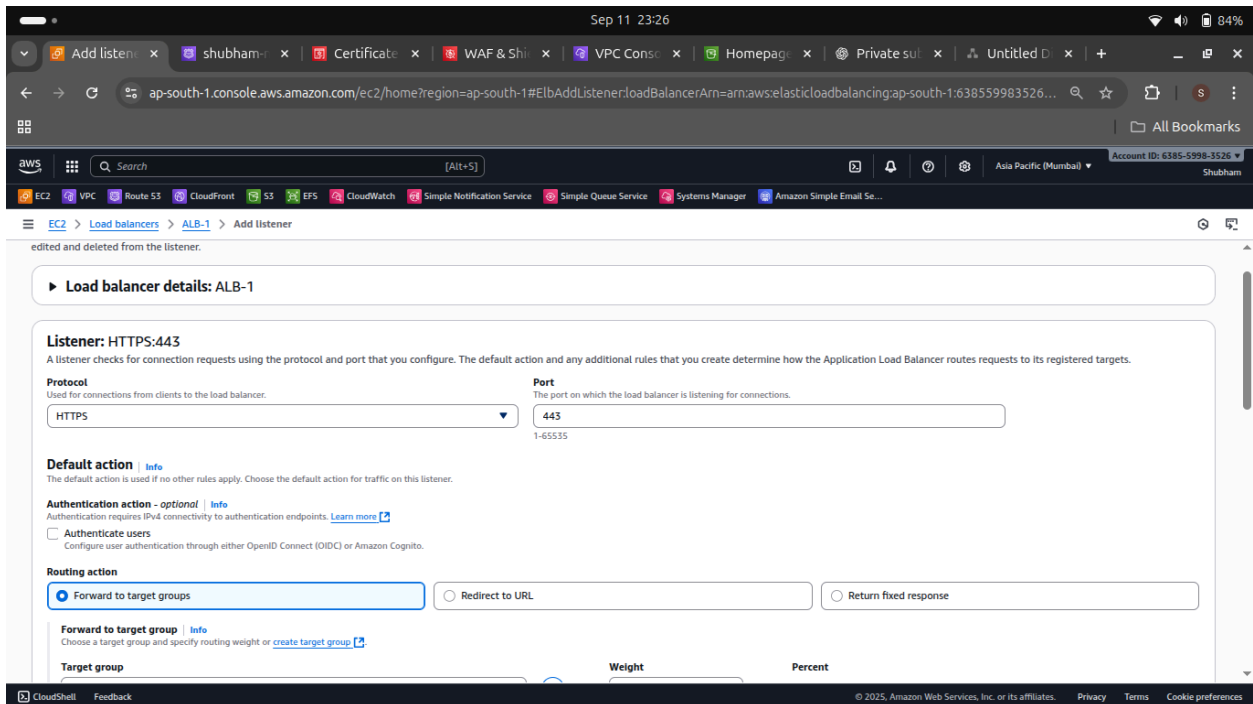
Now will add SSL certificate to my domain

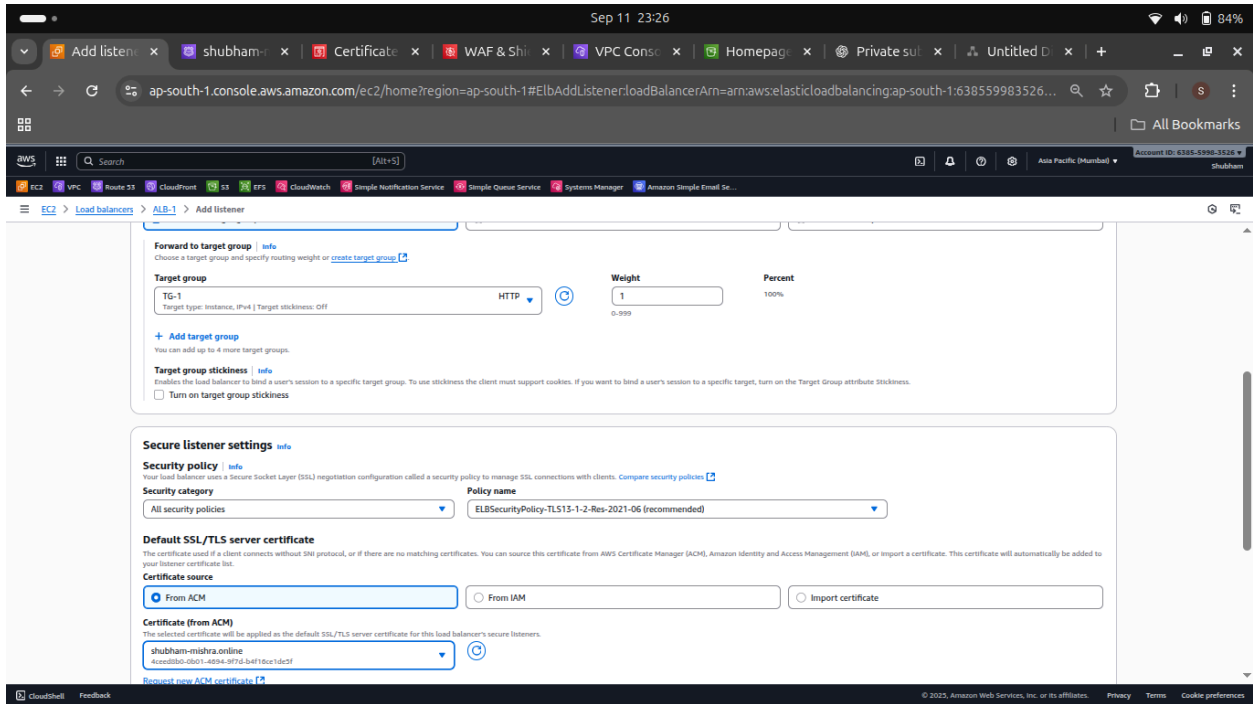




STEP 20

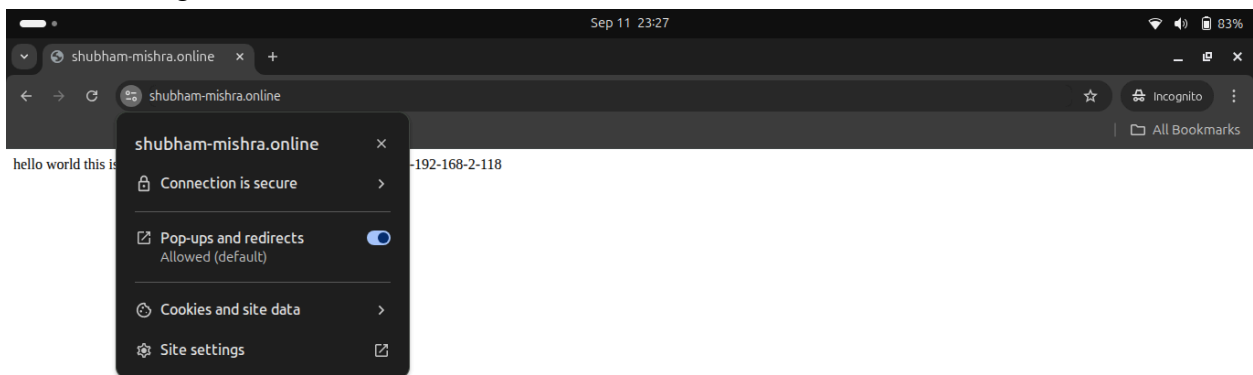
We will add https rule in loadbalancer to allows https access





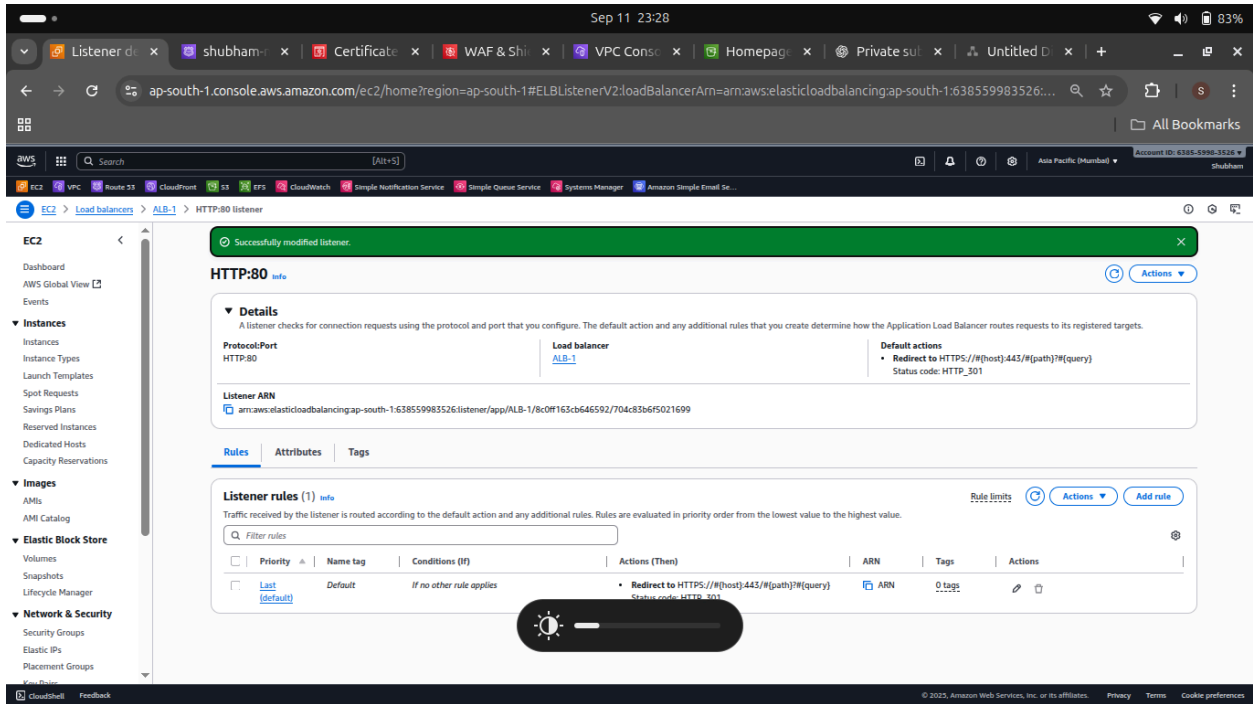
STEP 21

After successful validation we can access <https://shubham-mishra.online> which confirms ssl is working



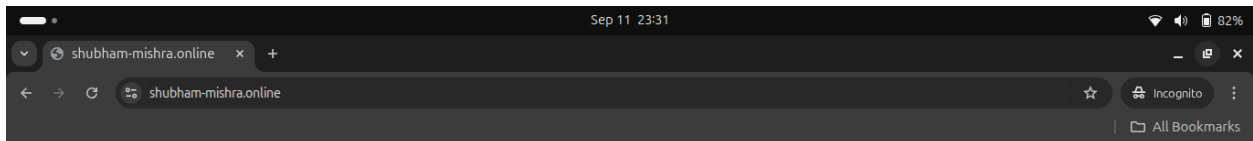
STEP 22

Now we will edit http rule in load balancer to redirect http request to https



STEP 23

We can see <http://shubham-mishra.online> request is redirected to <https://shubham-mishra.online>



This project showcases my ability to design and deploy a secure, scalable, and fault-tolerant AWS infrastructure by integrating networking, security, and

application delivery components. It reflects hands-on expertise in implementing cloud best practices suitable for production workloads.