Project Documentation: AWS Cloud Infrastructure [V-Profile Project/Lift And Shift]

Date: December 22, 2025

Engineer: Abdullah Mudassar

Focus: DevOps / Cloud Administration

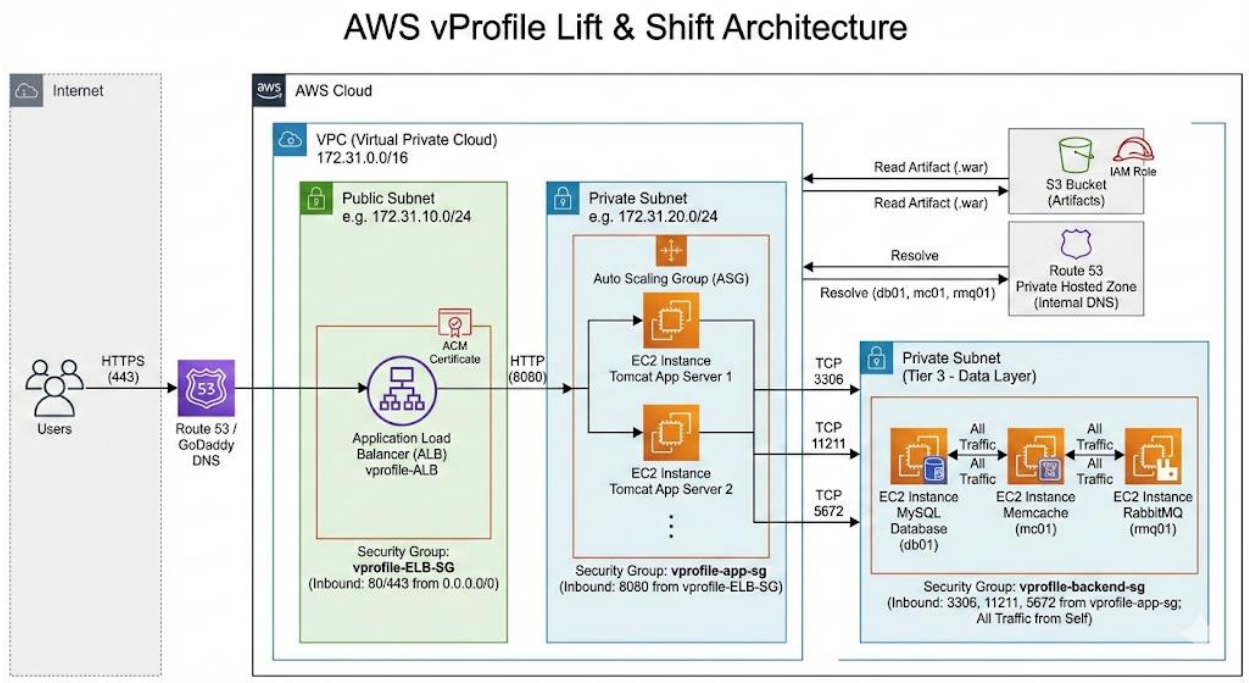
1. Project Introduction: vProfile Lift & Shift

The **vProfile** project is a production-level migration of a multi-tier Java web application stack. The goal is to move a complex environment consisting of a web server, application server, database, and messaging broker from a local environment to **AWS infrastructure**.

Instead of rewriting the application, we use a **Lift and Shift (Rehosting)** strategy. This approach allows a company to move to the cloud quickly to gain immediate benefits like **High Availability**, **Pay-as-you-go pricing**, and **Auto Scaling**, without the time-consuming process of changing the application code.

---

2. Technical Architecture Diagram



---

3. Architecture Explanation (The Flow)

The architecture is divided into three distinct layers to ensure security and performance:

Tier 1: The User & Public Access (Web Tier)

- **DNS (GoDaddy/Route 53):** The user enters your website URL. DNS translates this URL into the address of your **Application Load Balancer (ALB)**.
- **Load Balancer (ALB):** This is the entry point. It receives traffic on **Port 443 (HTTPS)**. It uses an SSL certificate from **AWS Certificate Manager (ACM)** to ensure the connection is encrypted and secure.

Tier 2: The Application Logic (App Tier)

- **Tomcat Servers:** Your Java code runs here on **EC2 instances**.
- **Auto Scaling Group (ASG):** These servers aren't static. If 1,000 users visit at once, the ASG automatically adds more Tomcat servers. If the traffic drops, it removes them to save money.
- **Port 8080:** The Load Balancer talks to Tomcat specifically on this port.

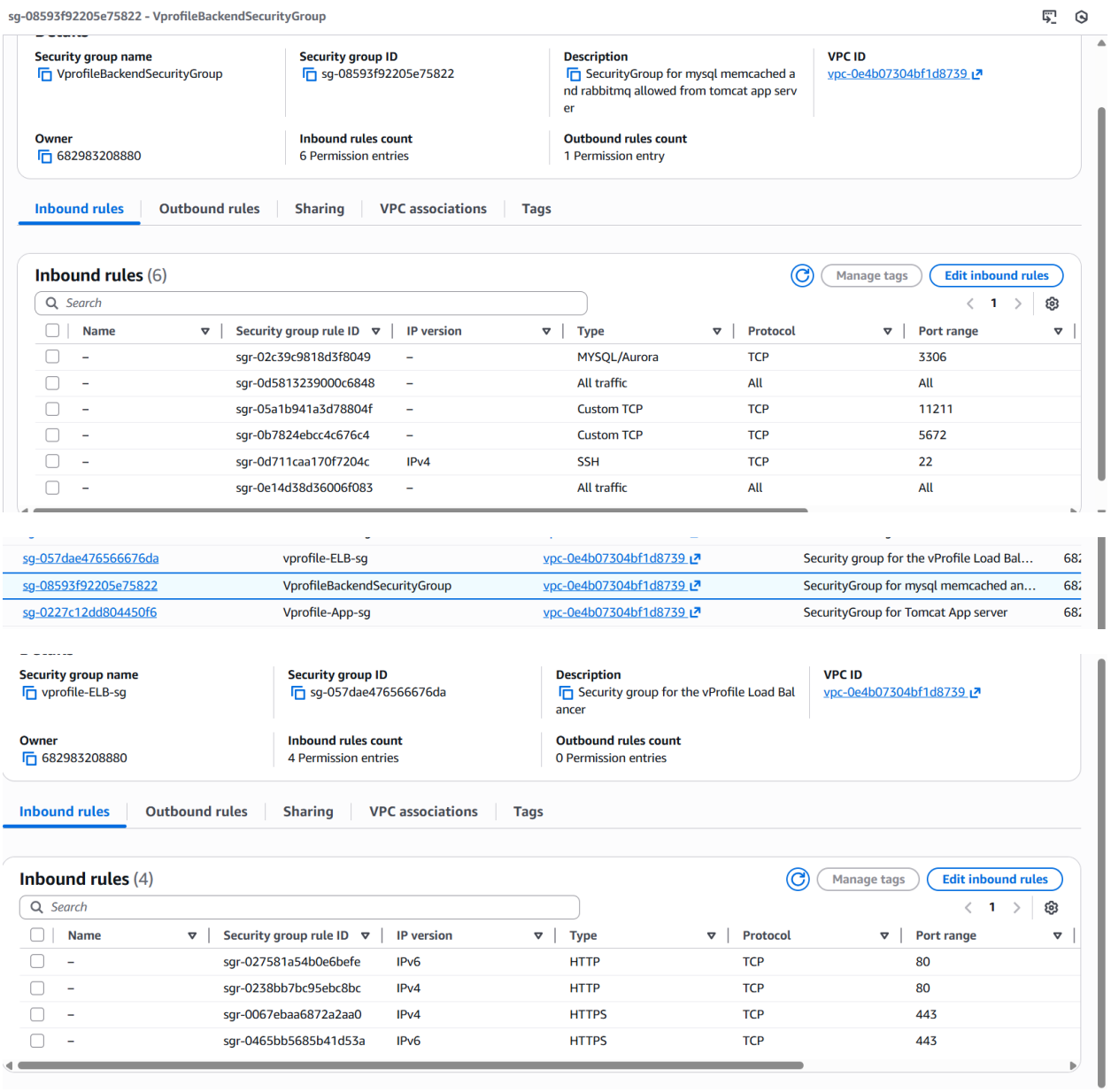
Tier 3: The Data & Support Services (Backend Tier)

- **MySQL (DB):** The relational database where all user data is stored (**Port 3306**).
- **Memcached:** A caching service that stores frequent data in memory so the app doesn't have to keep asking the database, making the site faster (**Port 11211**).
- **RabbitMQ:** A message broker that helps different parts of the application talk to each other without getting overwhelmed (**Port 5672**).
- **Route 53 (Private Zone):** An internal "phonebook" that lets the Tomcat servers find the Backend servers using names (like db01.vprofile.local) instead of IP addresses.

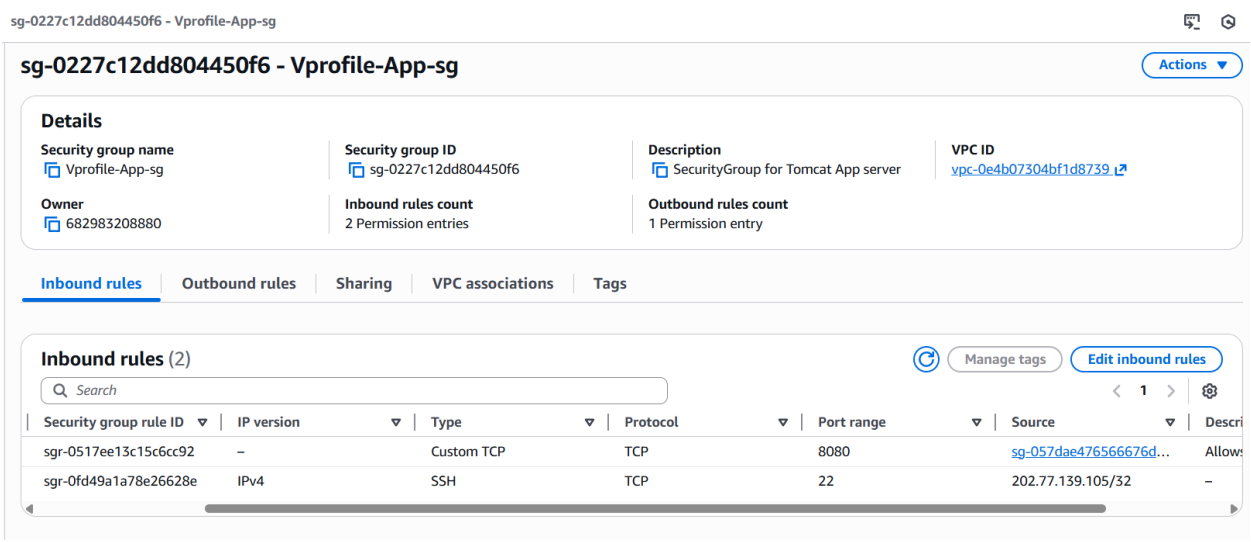
The Security Shield (Security Groups)

- Each tier has its own **Security Group (Firewall)**.
- The **Backend Tier** is completely hidden from the internet. It will **only** accept requests if they come directly from the **App Tier**. This is the core of cloud security.

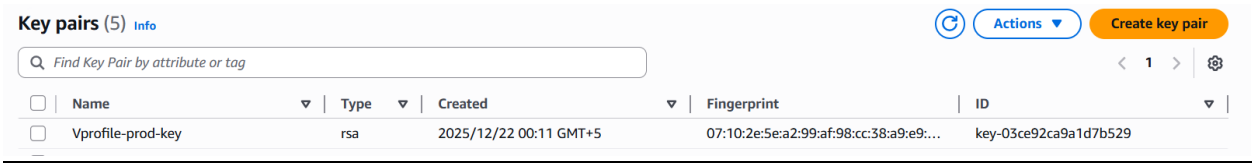
1. **CREATING SECURITY GROUPS:**


sg-08593f92205e75822 - VprofileBackendSecurityGroup
Security group name
VprofileBackendSecurityGroup
Security group ID
sg-08593f92205e75822
Description
SecurityGroup for mysql memcached and rabbitmq allowed from tomcat app server
VPC ID
vpc-0e4b07304bf1d8739
Owner
682983208880
Inbound rules count
6 Permission entries
Outbound rules count
1 Permission entry
Inbound rules
Outbound rules
Sharing
VPC associations
Tags
Inbound rules (6)
Manage tags
Edit inbound rules
Search
Name
Security group rule ID
IP version
Type
Protocol
Port range
– sgr-02c39c9818d3f8049 – MYSQL/Aurora TCP 3306
– sgr-0d5813239000c6848 – All traffic All All
– sgr-05a1b941a3d78804f – Custom TCP TCP 11211
– sgr-0b7824ebcc4c676c4 – Custom TCP TCP 5672
– sgr-0d711caa170f7204c IPv4 SSH TCP 22
– sgr-0e14d38d36006f083 – All traffic All All

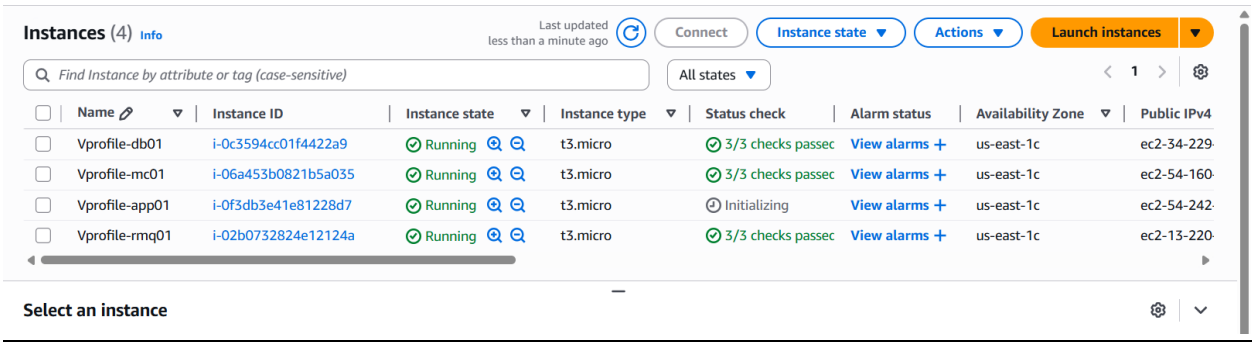
sg-057dae476566676da vprofile-ELB-sg vpc-0e4b07304bf1d8739 Security group for the vProfile Load Bal...
sg-08593f92205e75822 VprofileBackendSecurityGroup vpc-0e4b07304bf1d8739 SecurityGroup for mysql memcached an...
sg-0227c12dd804450f6 Vprofile-App-sg vpc-0e4b07304bf1d8739 SecurityGroup for Tomcat App server
Security group name
vprofile-ELB-sg
Security group ID
sg-057dae476566676da
Description
Security group for the vProfile Load Balancer
VPC ID
vpc-0e4b07304bf1d8739
Owner
682983208880
Inbound rules count
4 Permission entries
Outbound rules count
0 Permission entries
Inbound rules
Outbound rules
Sharing
VPC associations
Tags
Inbound rules (4)
Manage tags
Edit inbound rules
Search
Name
Security group rule ID
IP version
Type
Protocol
Port range
– sgr-027581a54b0e6befe IPv6 HTTP TCP 80
– sgr-0238bb7bc95ebc8bc IPv4 HTTP TCP 80
– sgr-0067ebaa6872a2aa0 IPv4 HTTPS TCP 443
– sgr-0465bb5685b41d53a IPv6 HTTPS TCP 443



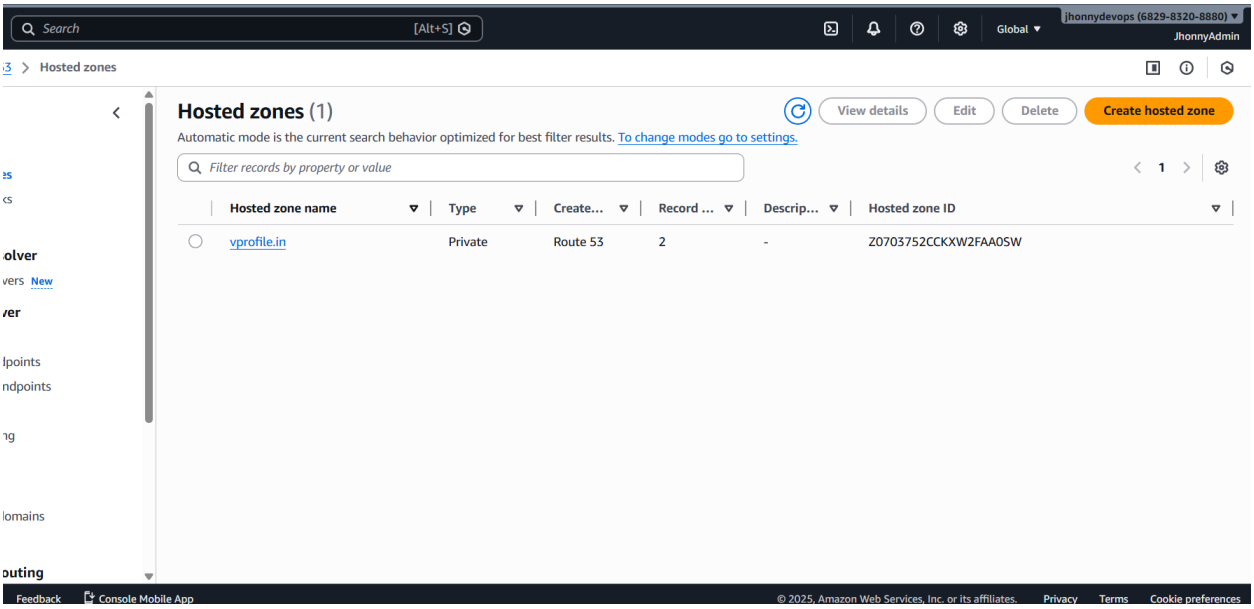
2. **Creating Key Pairs:**



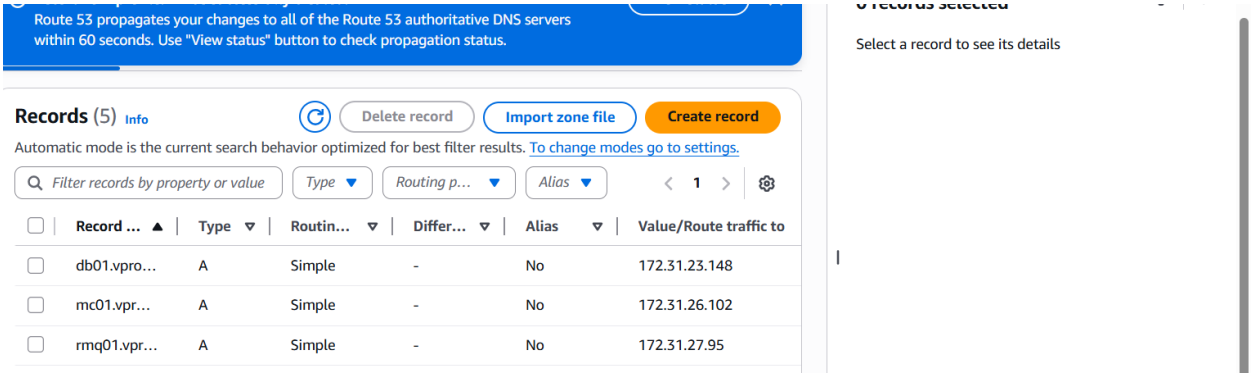
3. **Creating Ec2 Instances With user Data Scripts :**



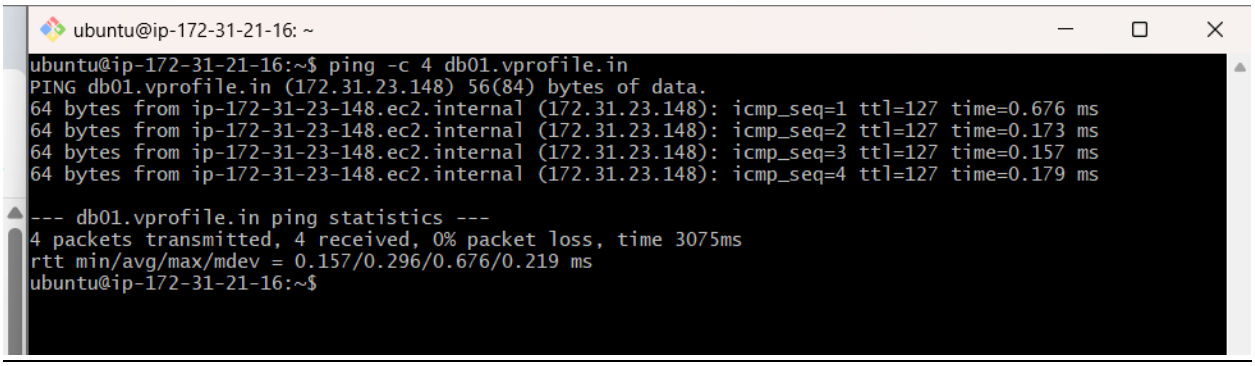
Create hosted Zone Using Route53:



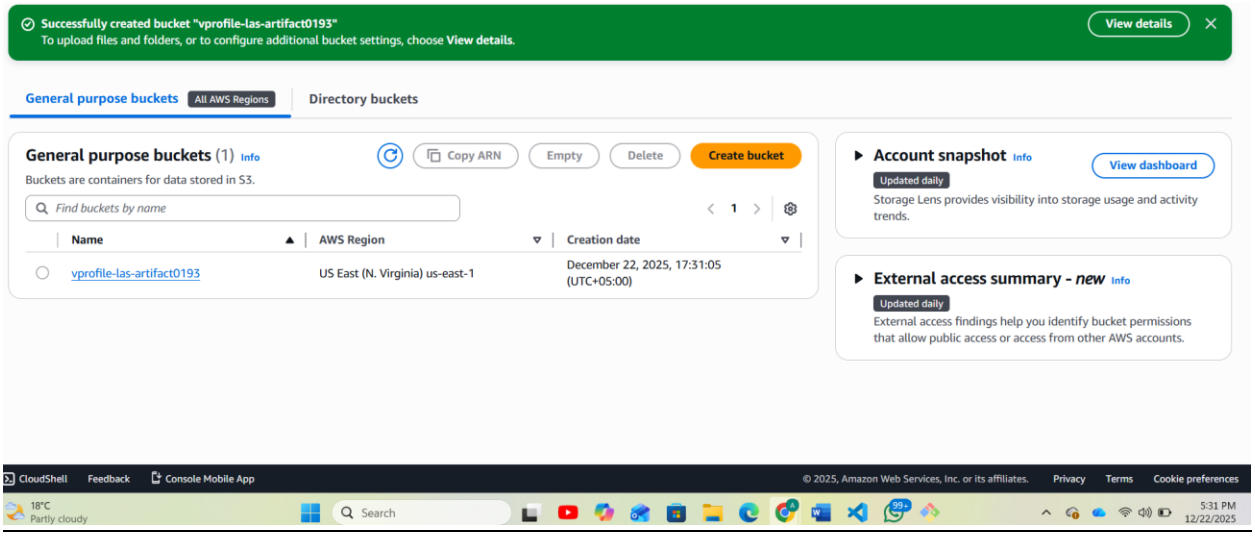
INTERNAL DNS RESOLUTION(Create a Record):



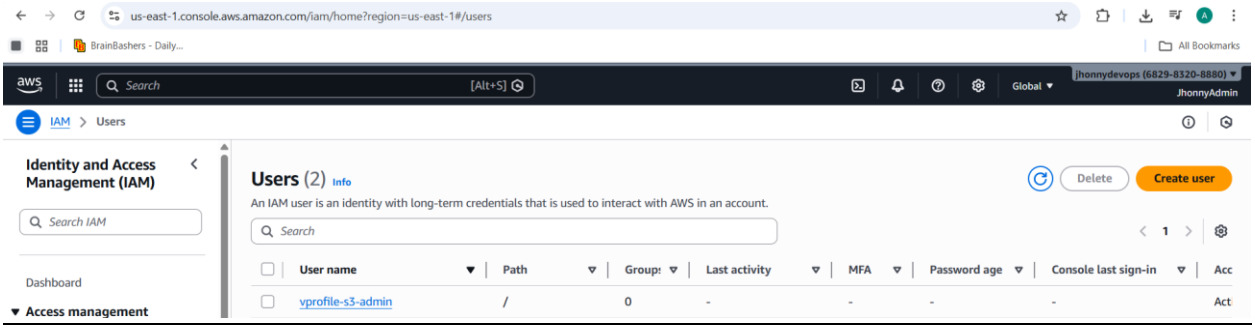
Checking Connection:



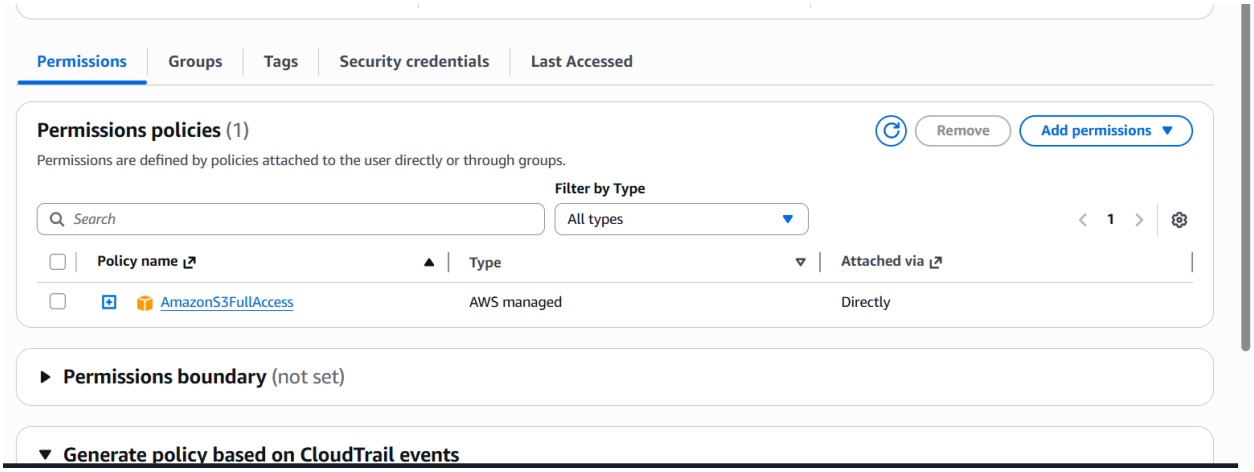
Creating S3 Bucket:



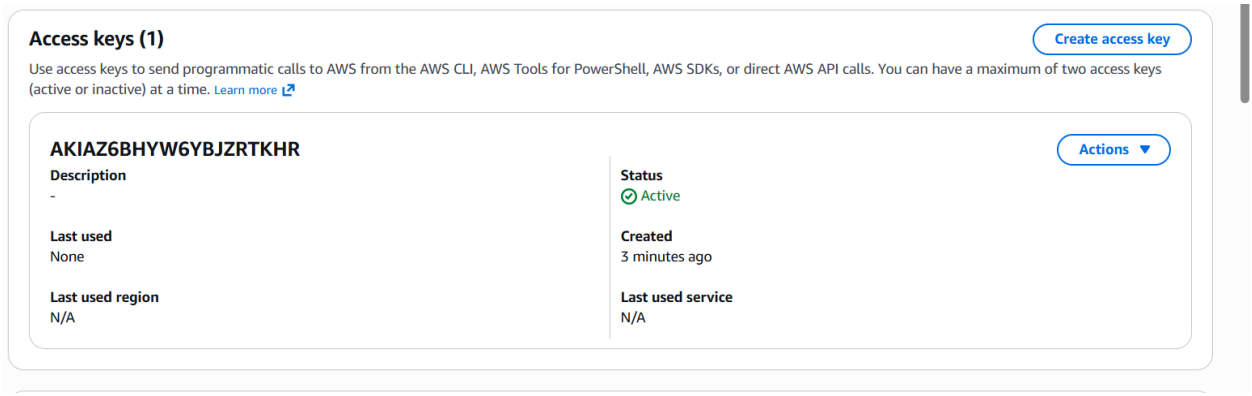
Create IAM user :



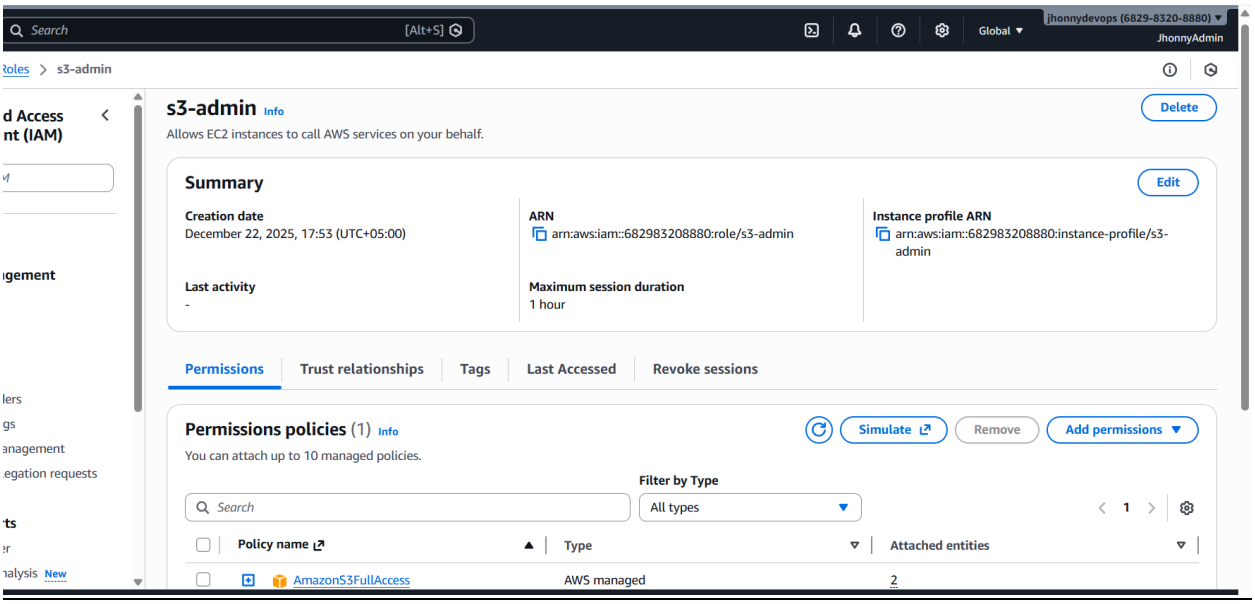
Attach policies:



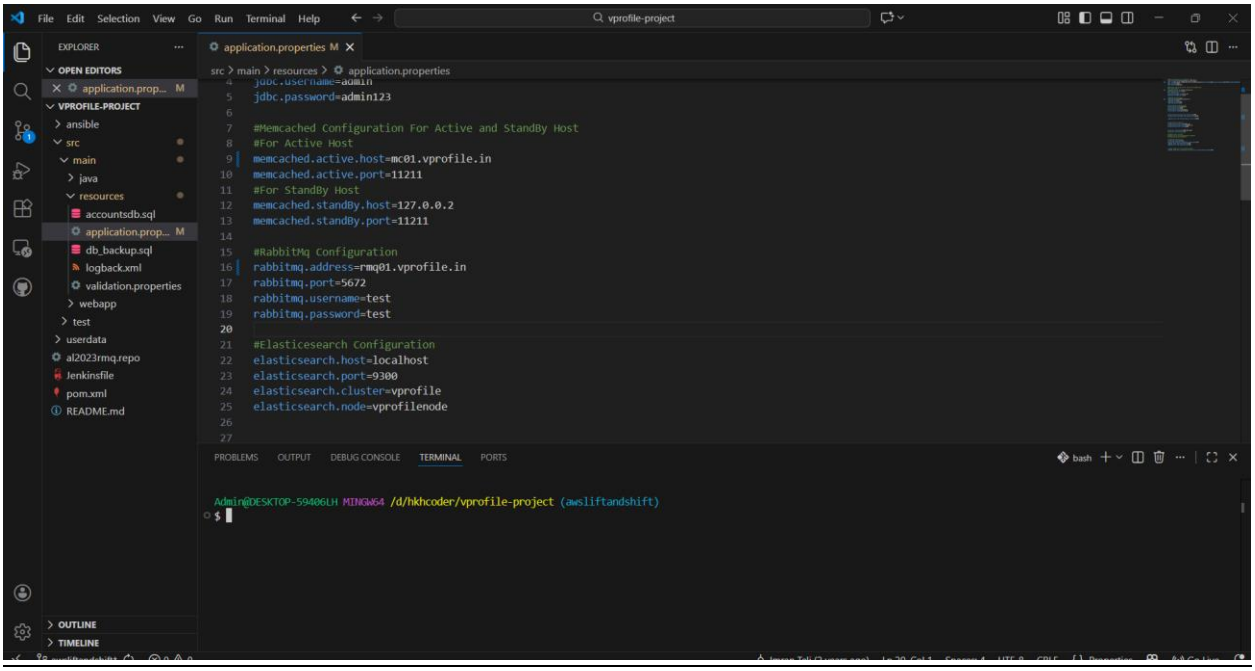
Create access key with the new user :



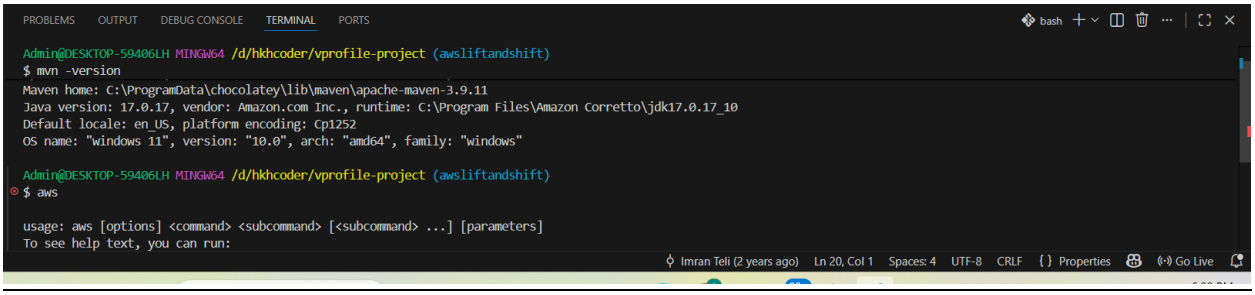
Create a Role with s3 full access permission:



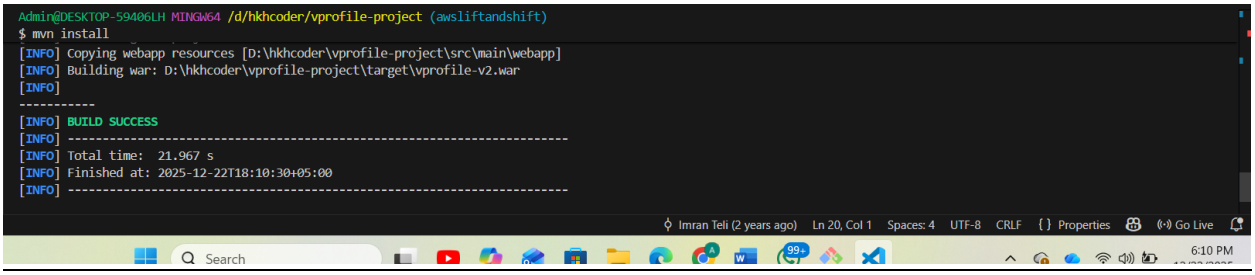
Opening a terminal:



Verifying maven and AWS :



Running a mvn install Command to build Artifact:



(note: after building configure aws using aws configure)

Pushing Artifact to s3 bucket:

```
Admin@DESKTOP-59406LH MINGW64 /d/hkhcoder/vprofile-project (awsliftandshift)
$ aws s3 cp target/vprofile-v2.war s3://vprofile-las-artifact0193/
upload: target\vprofile-v2.war to s3://vprofile-las-artifact0193/vprofile-v2.war

Admin@DESKTOP-59406LH MINGW64 /d/hkhcoder/vprofile-project (awsliftandshift)
$
```

```
Admin@DESKTOP-59406LH MINGW64 /d/hkhcoder/vprofile-project (awsliftandshift)
$ aws s3 ls  s3://vprofile-las-artifact0193/
2025-12-22 18:29:56   83278595 vprofile-v2.war

Admin@DESKTOP-59406LH MINGW64 /d/hkhcoder/vprofile-project (awsliftandshift)
$
```

Imran teli (1 year ago) Ln 28, Col 45 Spaces: 4 UTF-8 CRLF { } Properties Go Live

(Note: ssh app instance)

Install AWS CLI:

```
root@ip-172-31-21-16: ~
root@ip-172-31-21-16:~# snap install aws-cli --classic
aws-cli (v2/stable) 2.32.21 from Amazon Web Services (aws✓) installed
root@ip-172-31-21-16:~#
```

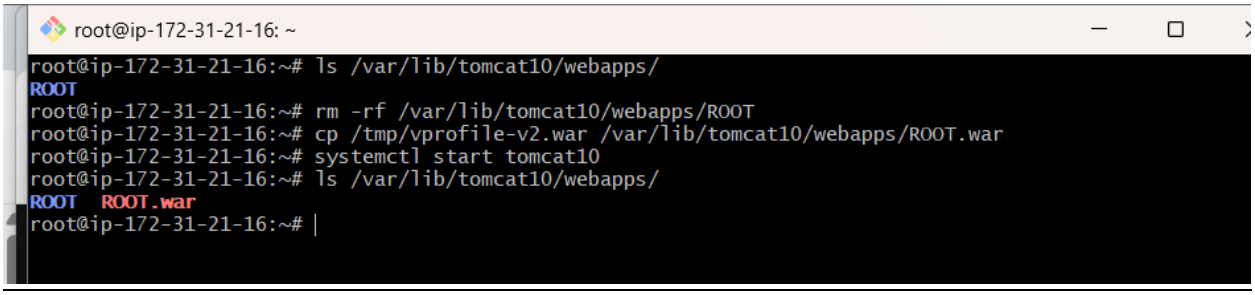
Copying Artifact from S3 bucket:

```
root@ip-172-31-21-16:~# aws s3 cp s3://vprofile-las-artifact0193/vprofile-v2.war /tmp/
download: s3://vprofile-las-artifact0193/vprofile-v2.war to ../tmp/vprofile-v2.war
root@ip-172-31-21-16:~#
```

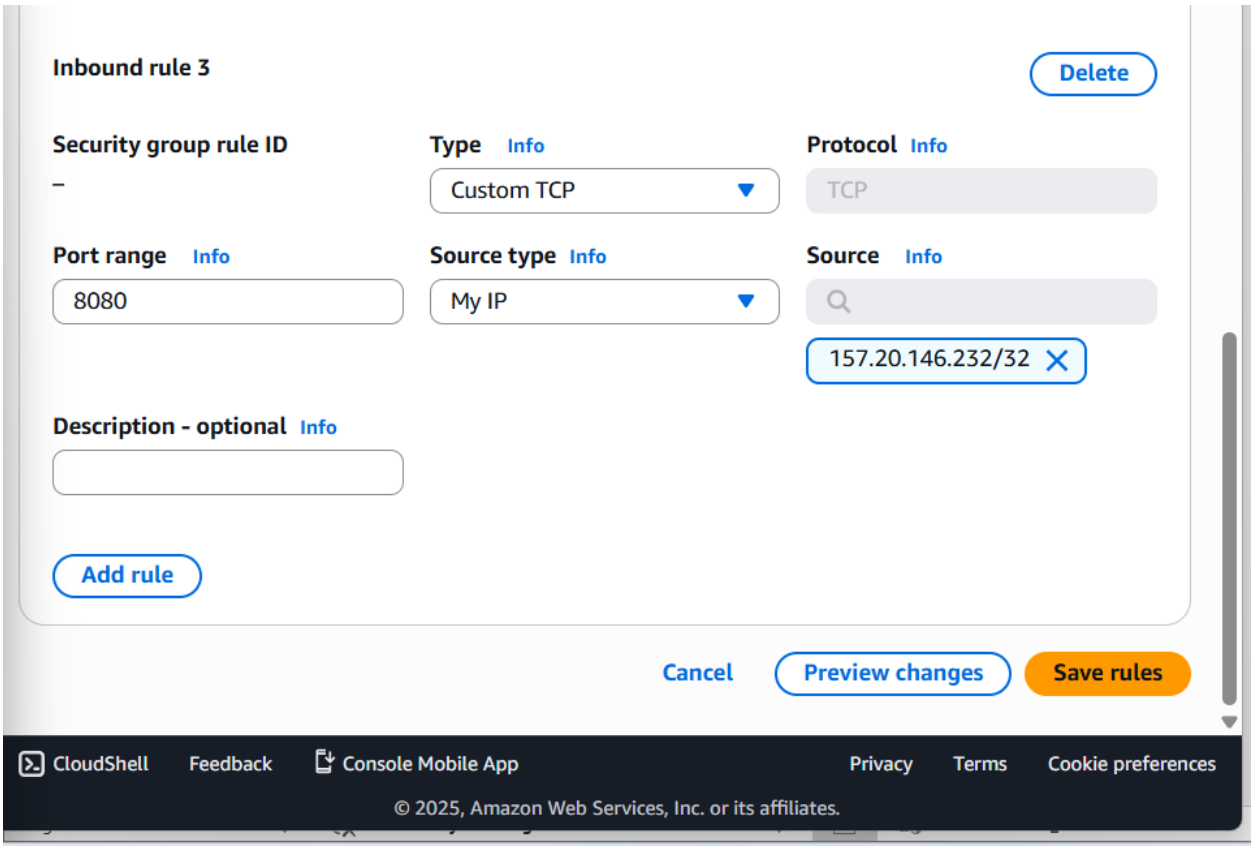
Stopping Tomcat service

```
root@ip-172-31-21-16:~# systemctl stop tomcat10
Warning: The unit file, source configuration file or drop-ins of tomcat10.service changed on disk. Run
'systemctl daemon-reload' to reload units.
root@ip-172-31-21-16:~# systemctl daemom-reload
Unknown command verb 'daemom-reload', did you mean 'daemon-reload'?
root@ip-172-31-21-16:~# systemctl daemon-reload
root@ip-172-31-21-16:~#
```

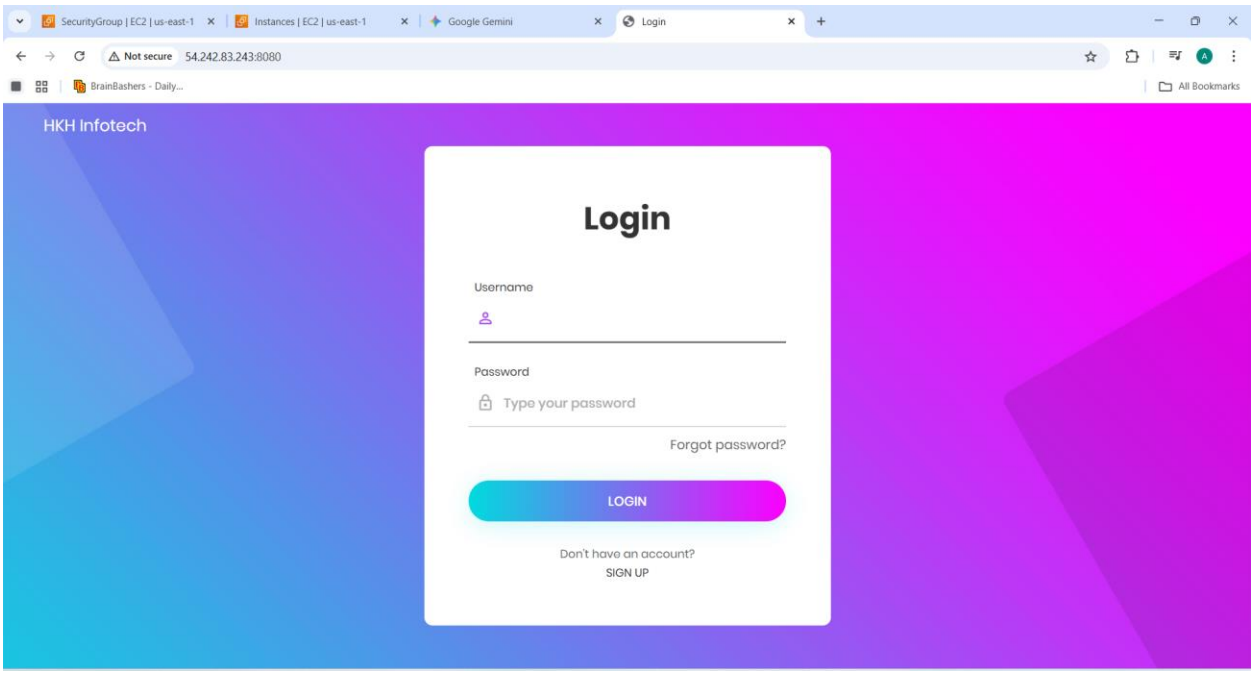
Manual Deployment and Hot-Swapping of vProfile Artifact to Tomcat Application Server:



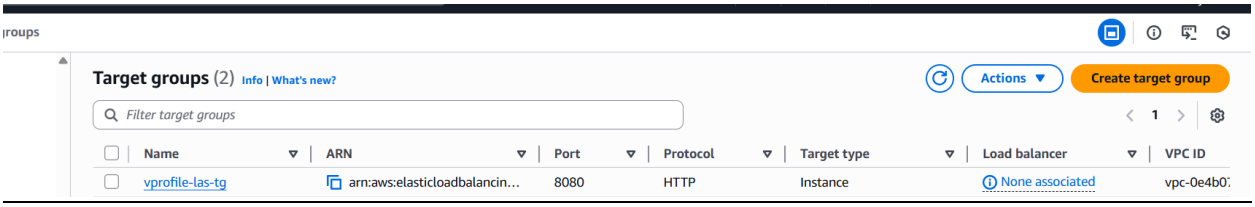
Adding rule to check app is running:



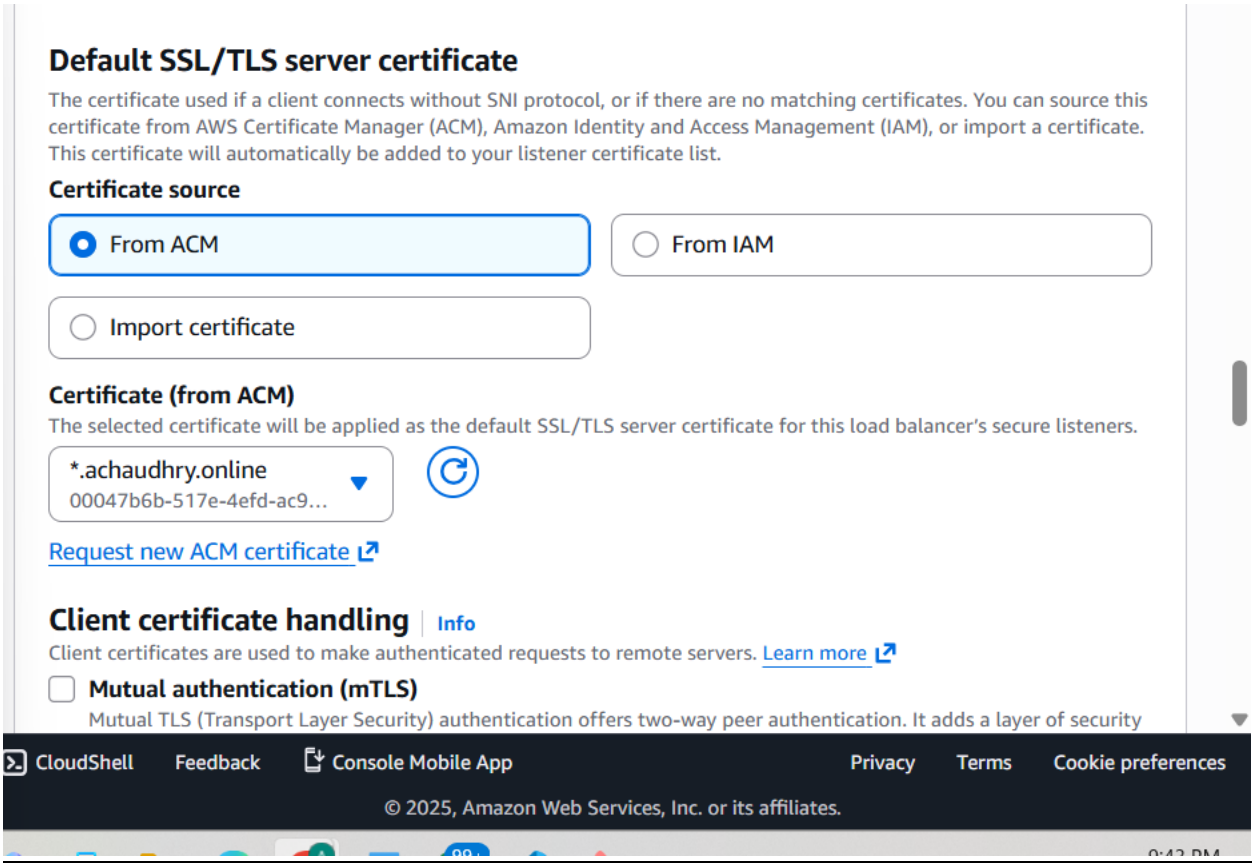
Accesing App from public app for verification:

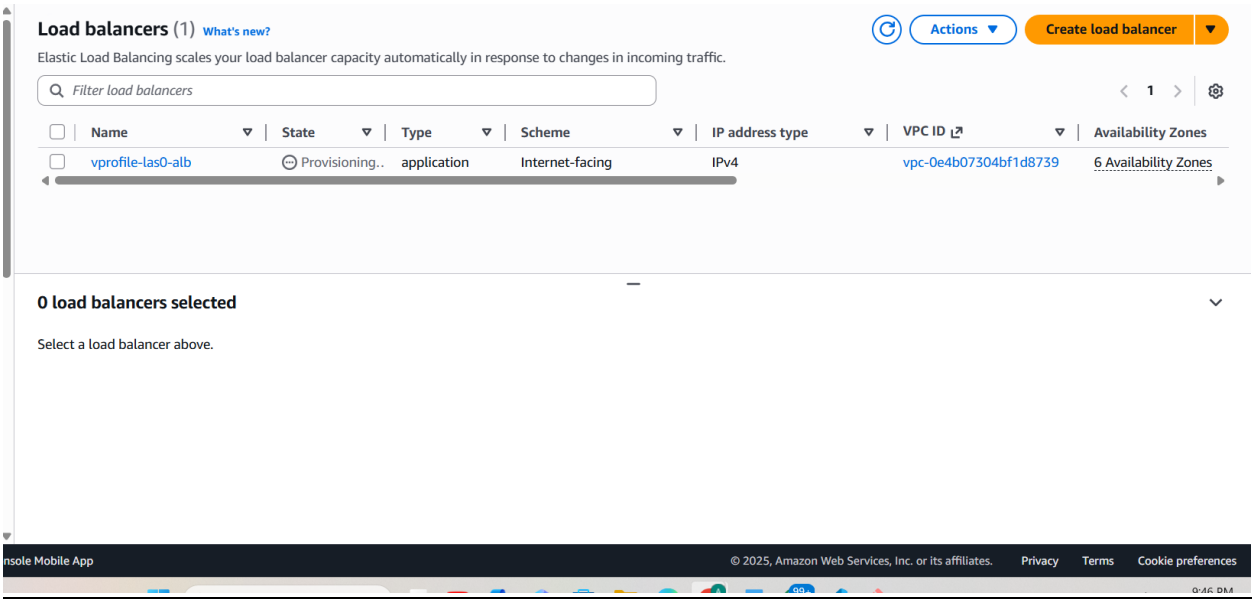


Creating Target Group:



Creating Load Balancer And adding Acm certificate:



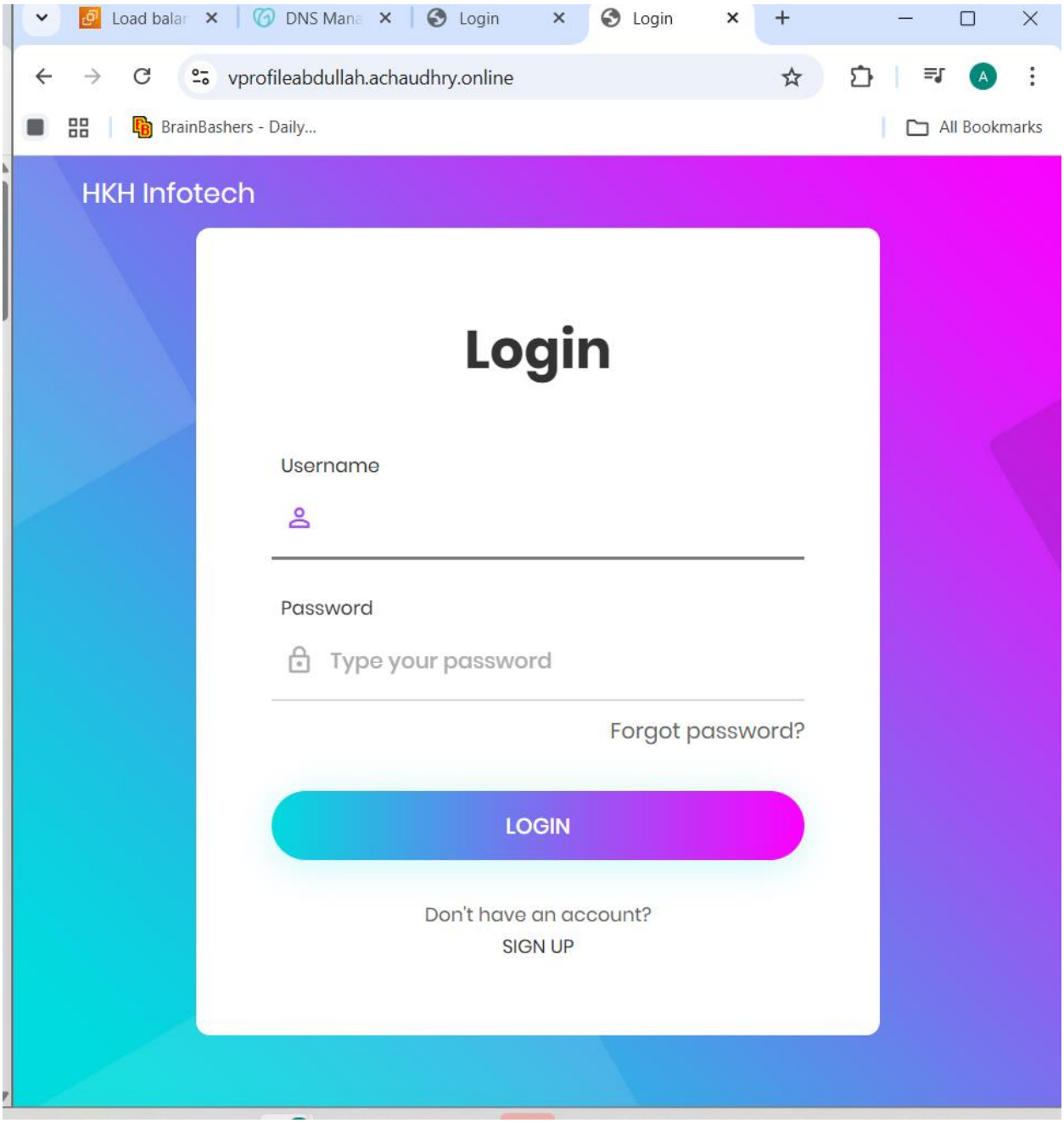


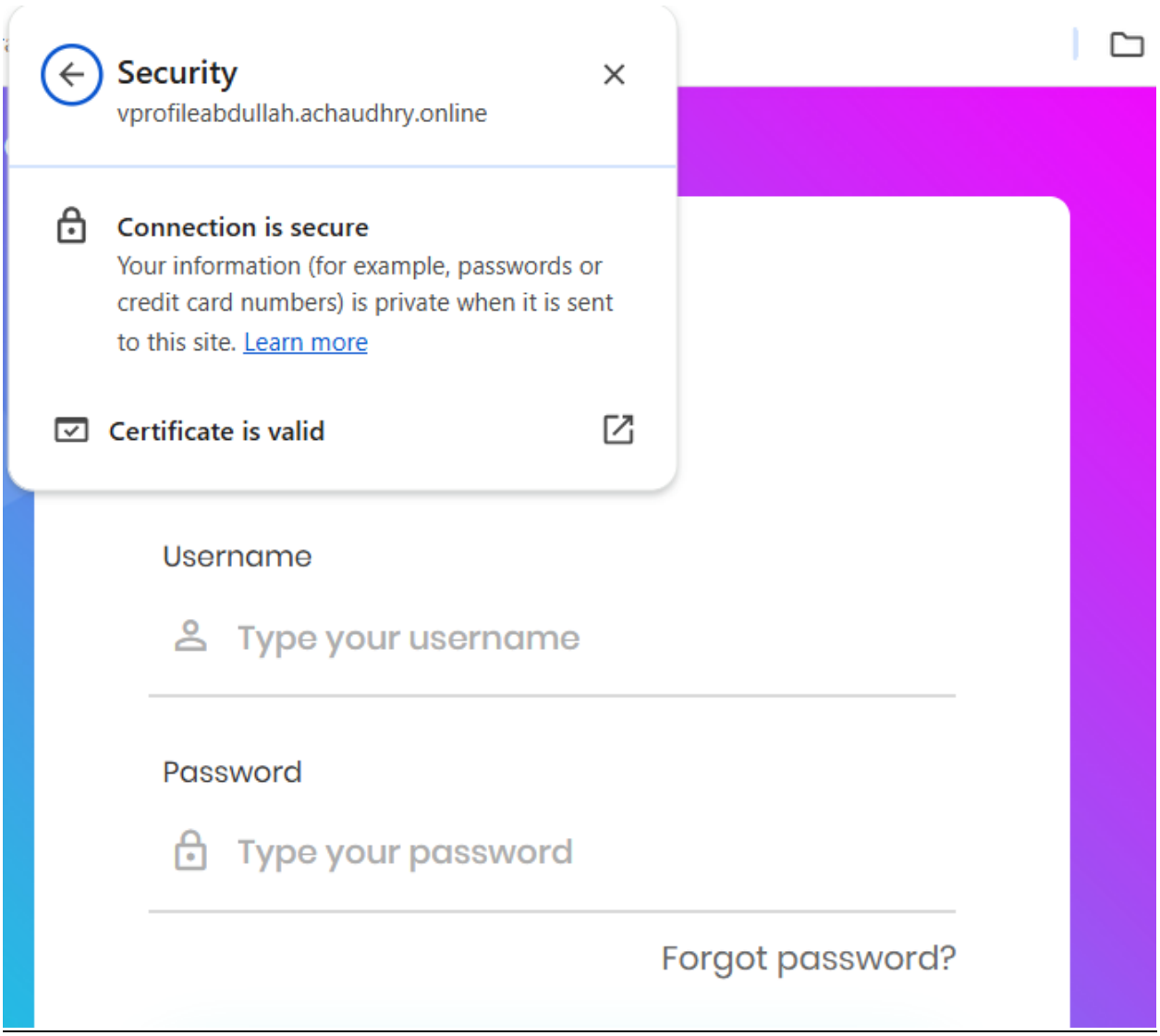
Adding Load Balancers DNS name in our domain Record:

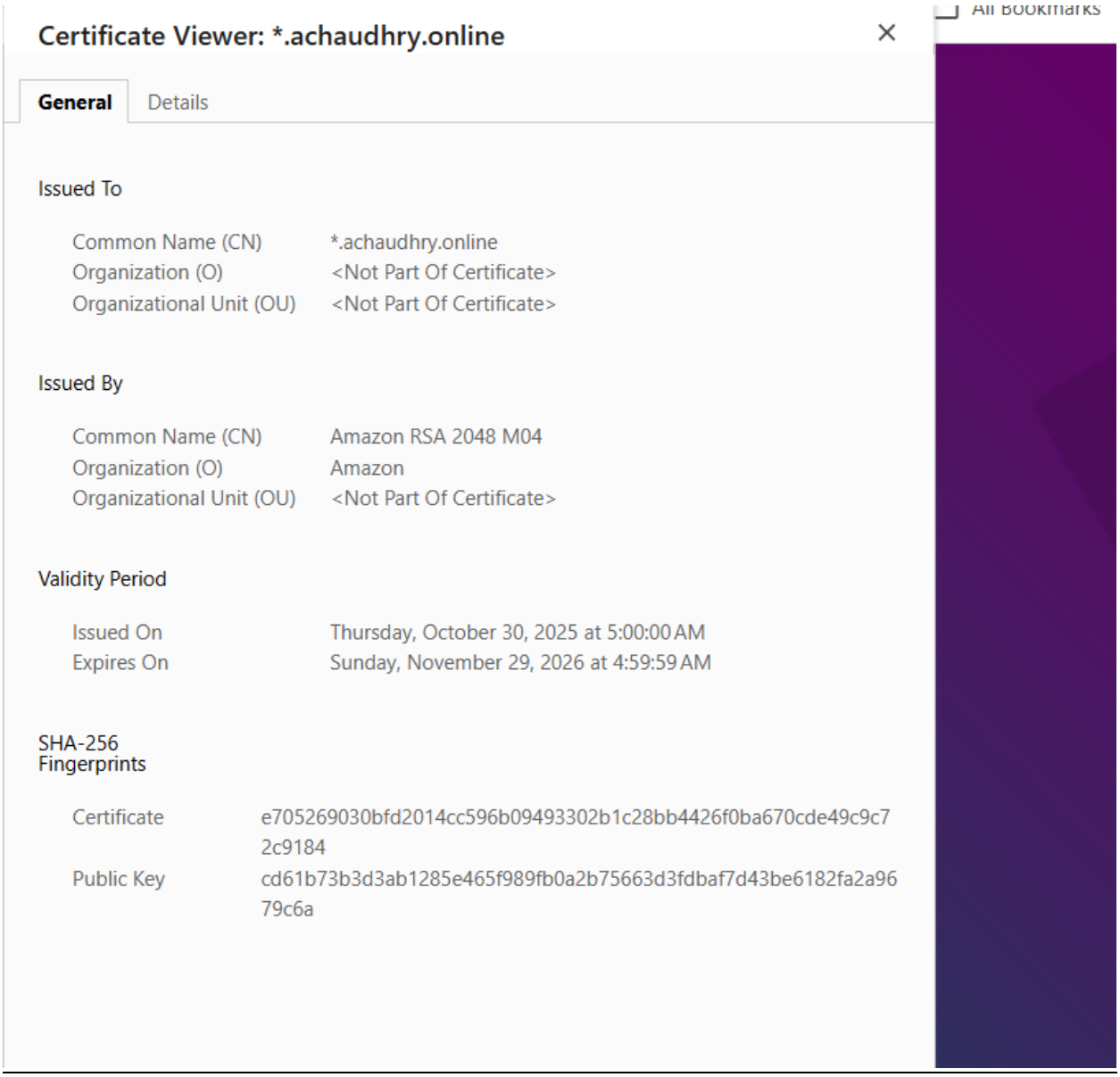
(Note: Iam using GoDaddy)

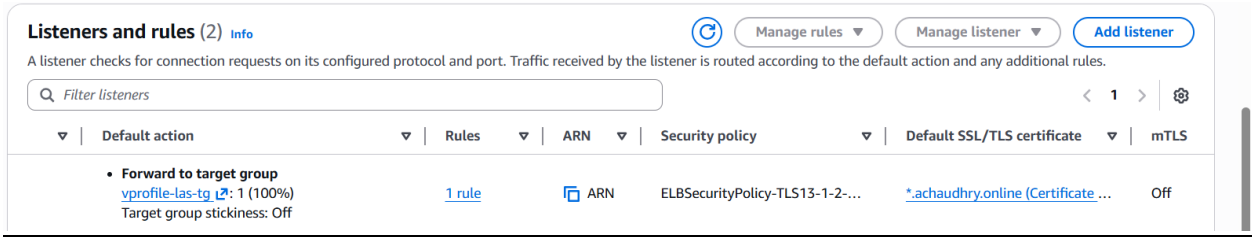
☑	CNAME	vprofileabdullah	vprofile-las0-alb-1574271398.us-east-1.elb.amazonaws.com.	1 Hour	🗑	✎

Accesing app with our own url:

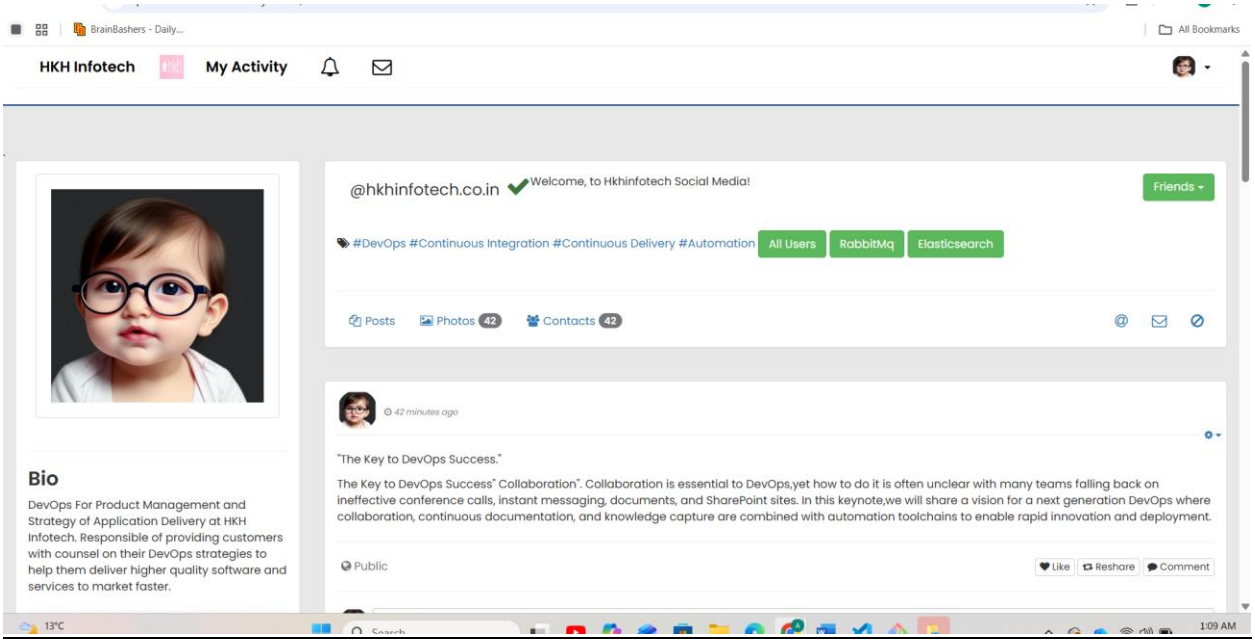



Security
vprofileabdullah.achaudhry.online
Connection is secure
Your information (for example, passwords or credit card numbers) is private when it is sent to this site. Learn more
Certificate is valid
Username
Type your username
Password
Type your password
Forgot password?

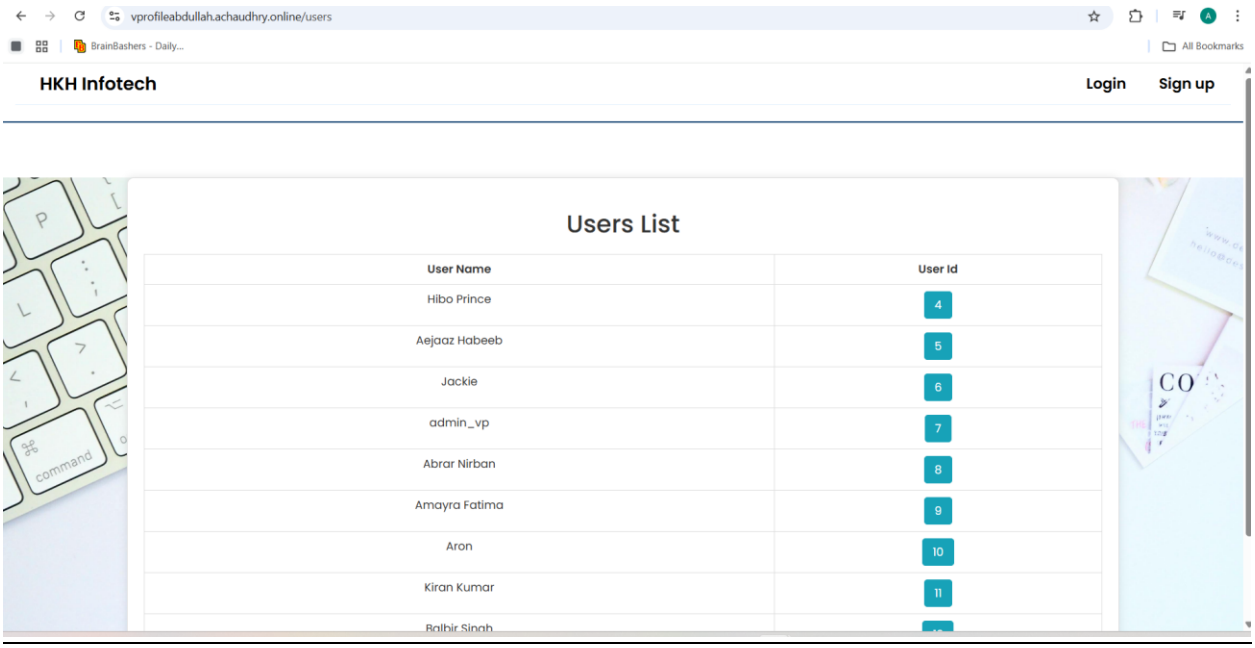


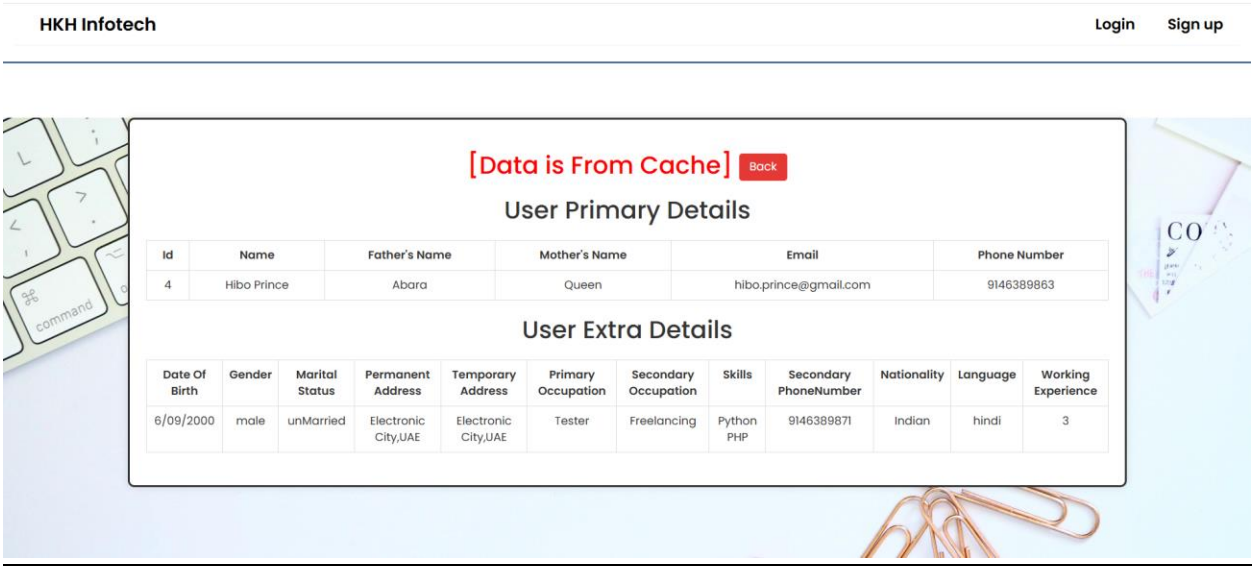


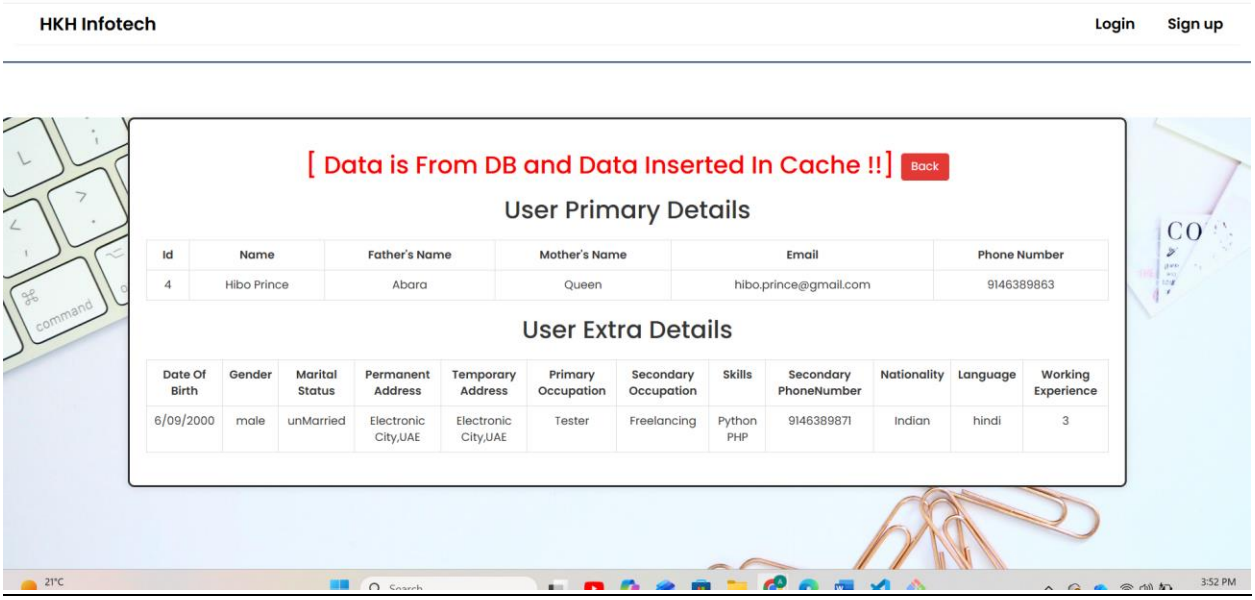
Logging to verify Database Connection:



Verifying Memcached Working :



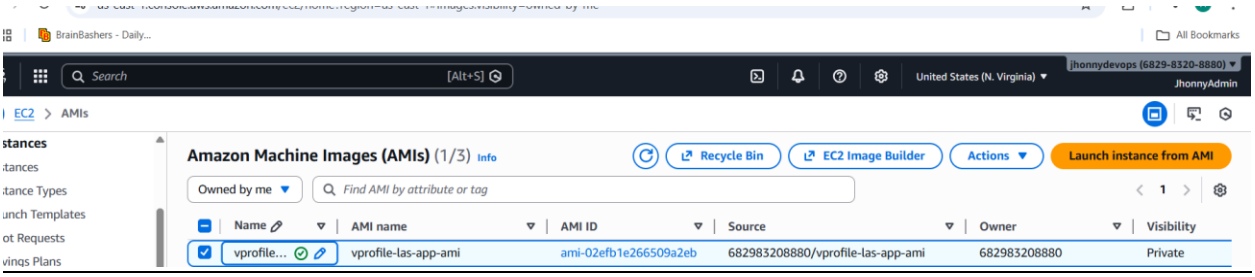




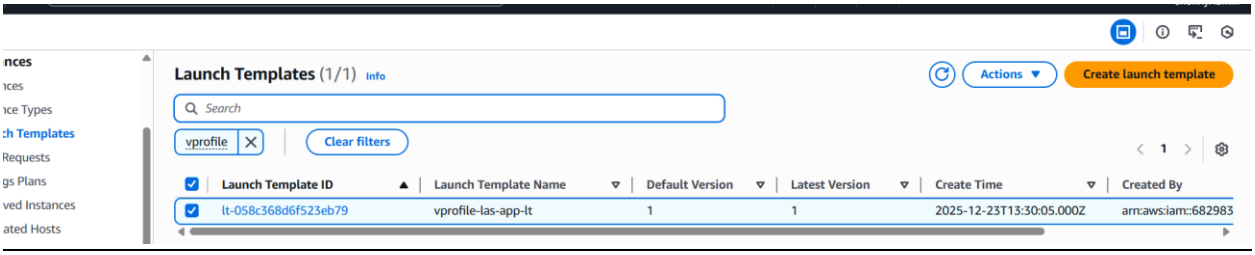
Process to configure autoscaling group:

1. First we need Ami of the instance
2. Launch Template
3. Auto scalling group

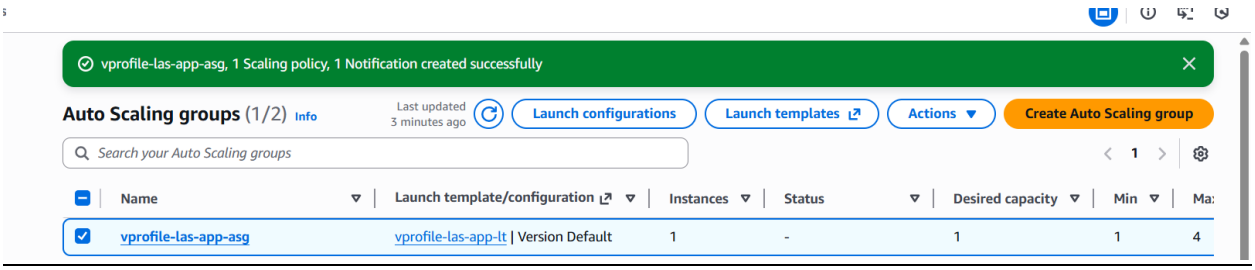
AMI created of app01 instance:



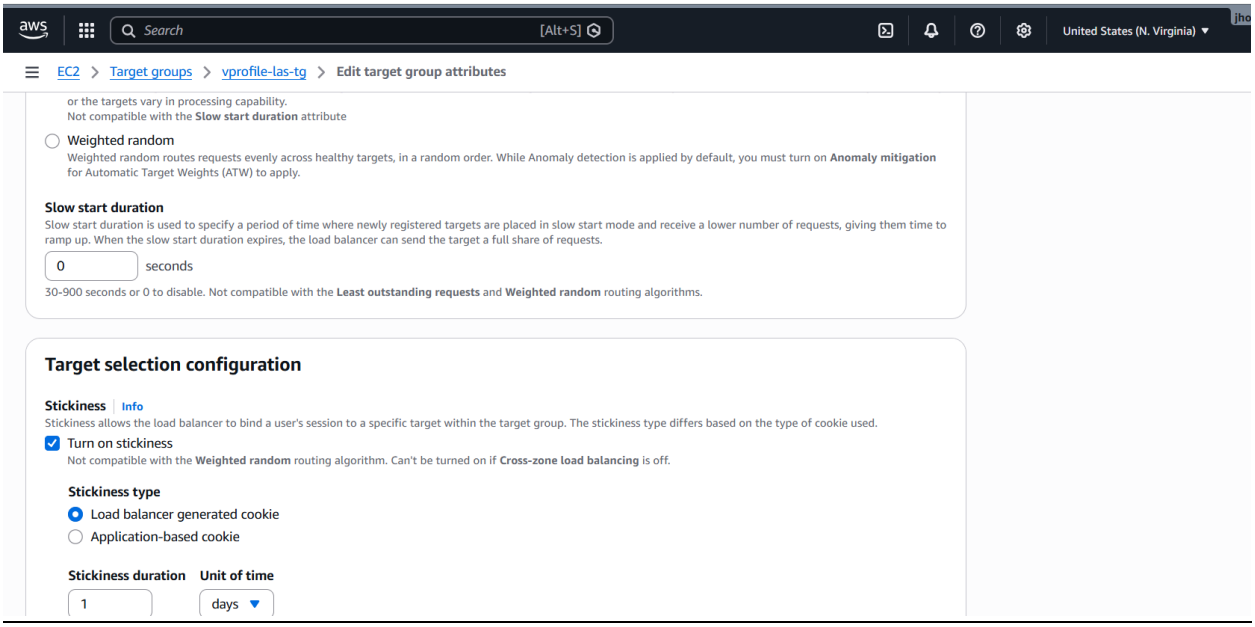
Create Launch Template:



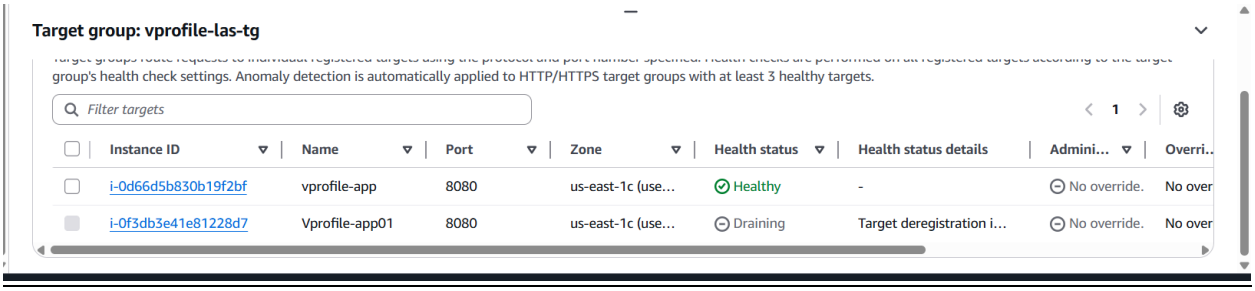
Created auto Scalling Group:

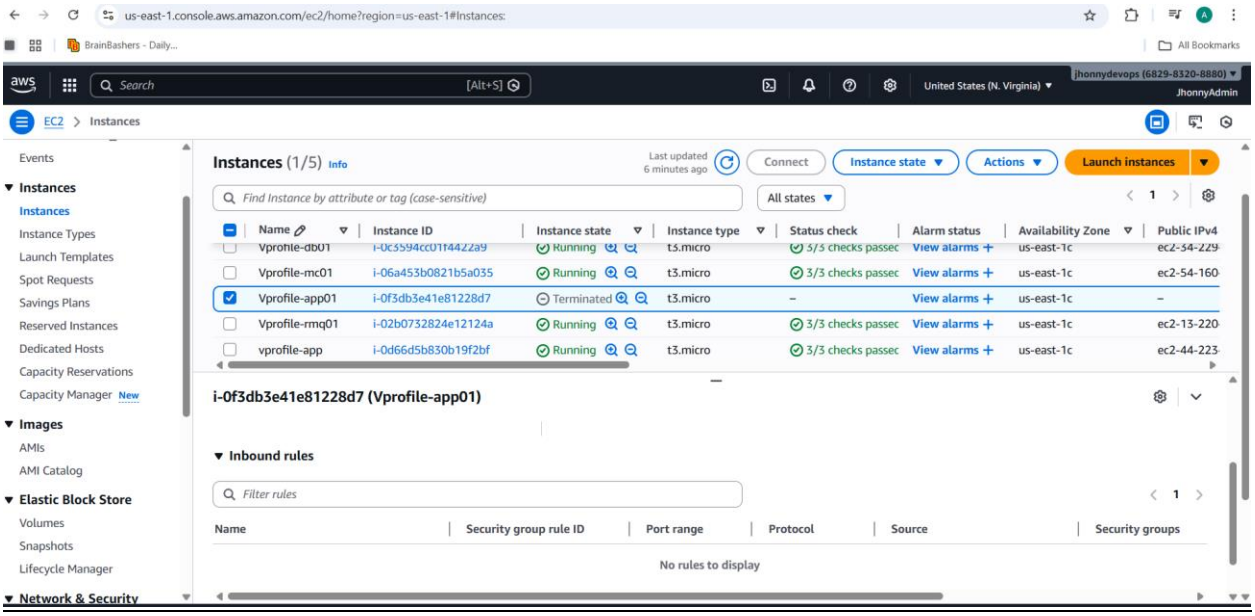


Enable stickiness on auto scalling Group:

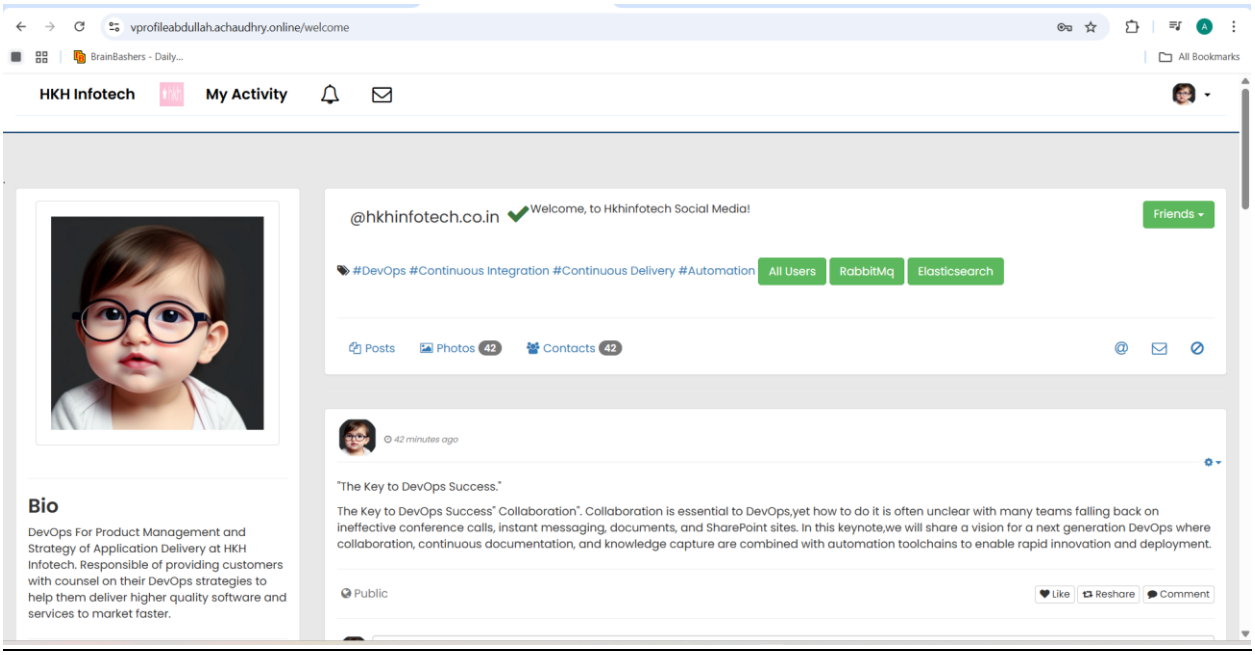


Deregister target App01 from Target Group and delete app01 instance to AutoScallingGroup to handle:





AutoScalling Group handling the application instance:


vprofileabdullah.achaudhry.online/welcome
BrainBashers - Daily...
All Bookmarks
HKH Infotech
My Activity
@hkhinfotech.co.in
Welcome, to Hkhinfotech Social Media!
Friends
#DevOps #Continuous Integration #Continuous Delivery #Automation
All Users
RabbitMq
Elasticsearch
Posts
Photos 42
Contacts 42
42 minutes ago
"The Key to DevOps Success."
The Key to DevOps Success" Collaboration". Collaboration is essential to DevOps,yet how to do it is often unclear with many teams falling back on ineffective conference calls, instant messaging, documents, and SharePoint sites. In this keynote,we will share a vision for a next generation DevOps where collaboration, continuous documentation, and knowledge capture are combined with automation toolchains to enable rapid innovation and deployment.
Public
Like
Reshare
Comment
Bio
DevOps For Product Management and Strategy of Application Delivery at HKH Infotech. Responsible of providing customers with counsel on their DevOps strategies to help them deliver higher quality software and services to market faster.