MINI-PROJECT-3 :

Web Application Deployment on AWS using EC2, Target Groups, and Load Balancer

Objective :

To deploy a simple web application on AWS and make it accessible on the internet using EC2 instances, target groups, and an Application Load Balancer. The goal is to ensure high availability, scalability, and automated setup using user data.

Services Used :

* Amazon EC2 – For hosting the application.

* Elastic Load Balancer (ALB) – For distributing incoming traffic across instances.

* Target Groups – For managing and monitoring registered instances.

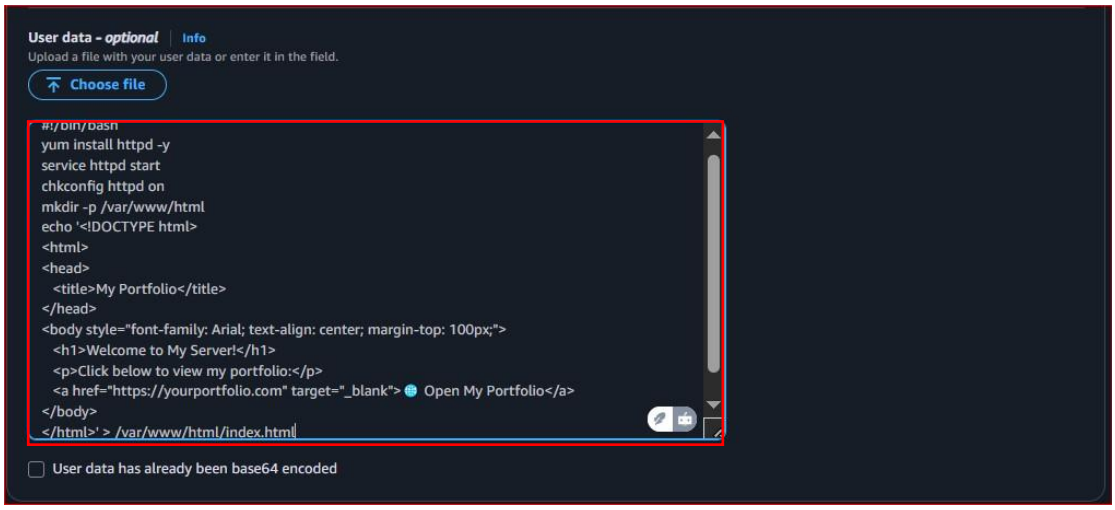
* Security Groups – For controlling inbound and outbound traffic.

* User Data (Linux commands) – For automated software installation and web server configuration.

Steps :

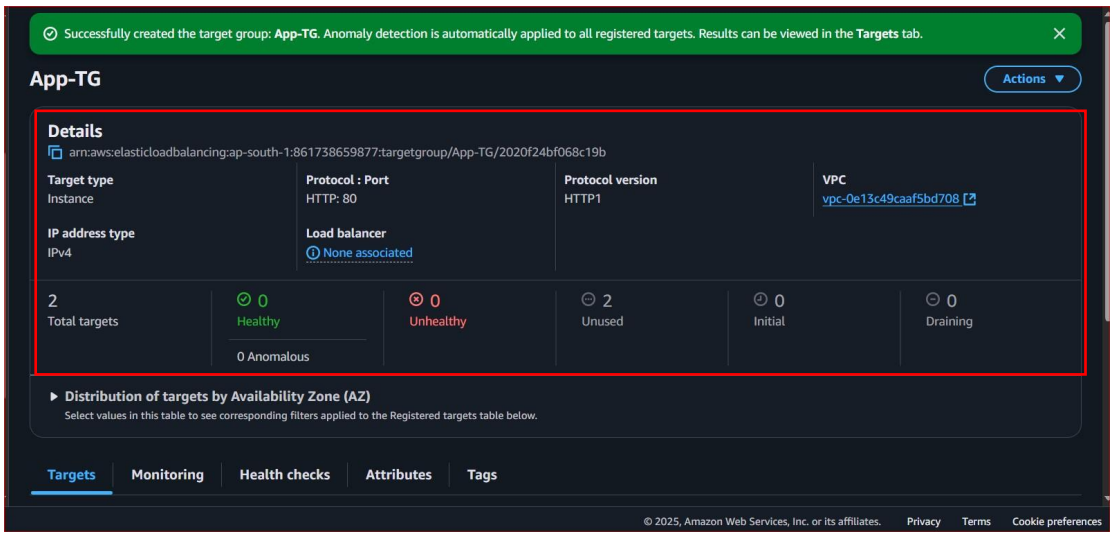
1.Launched an EC2 Instance :

-Selected Amazon Linux AMI.
-Configured instance details and added user data script for automatic Apache/HTTPD installation.
-Opened HTTP (port 80) in the security group.



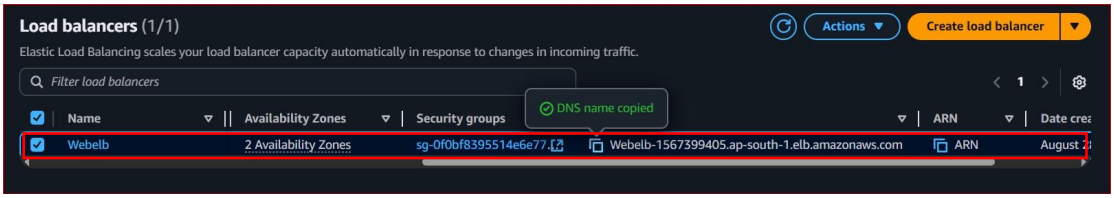
2. **Created a Target Group :**

-Defined the target group with protocol HTTP.
-Registered the EC2 instance to the target group.



3.Created an Application Load Balancer :

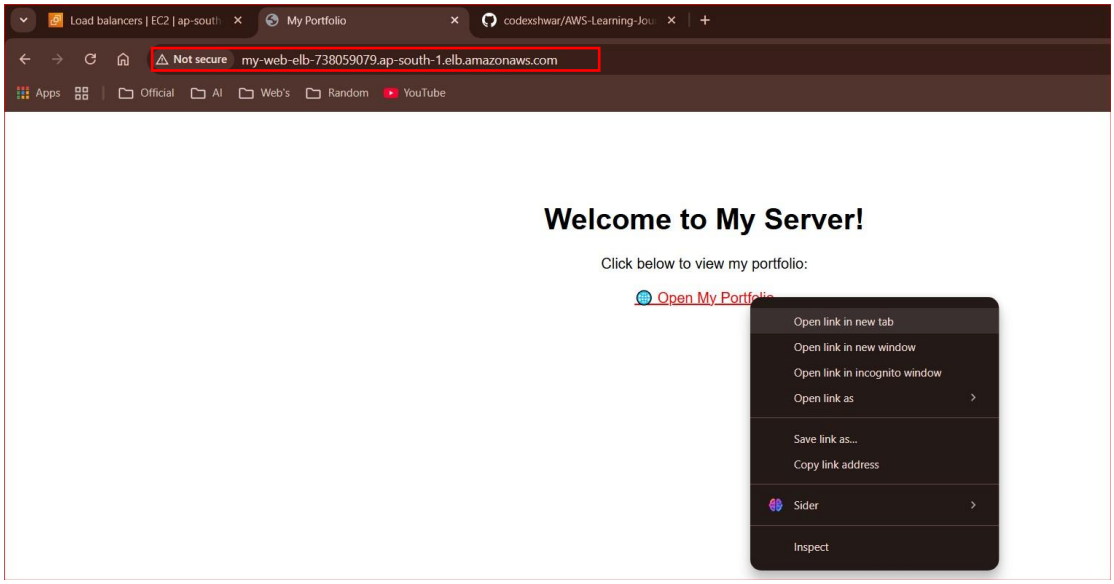
-Configured listeners on port 80.
-Attached the previously created target group.
-Security group allowed inbound traffic (HTTP).



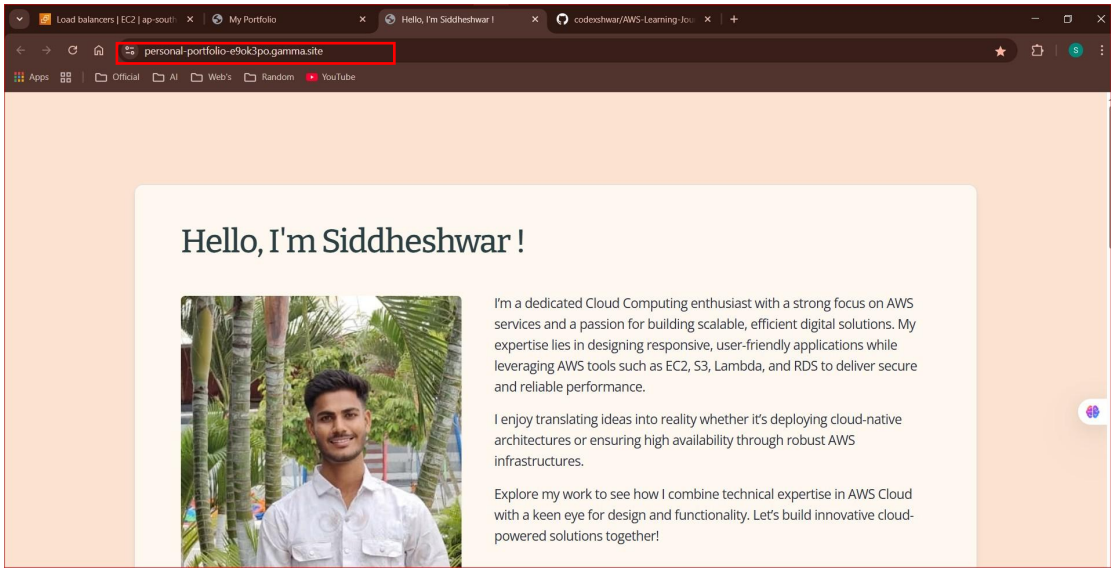
4.Testing the Deployment :

-Accessed the Load Balancer DNS name in a browser.
-Verified that the website was loading successfully.
- After Creation of loadbalancer Copy the DNS name and paste it in your browser and check it is working or not.

5.After opening it agin open it in new tap :



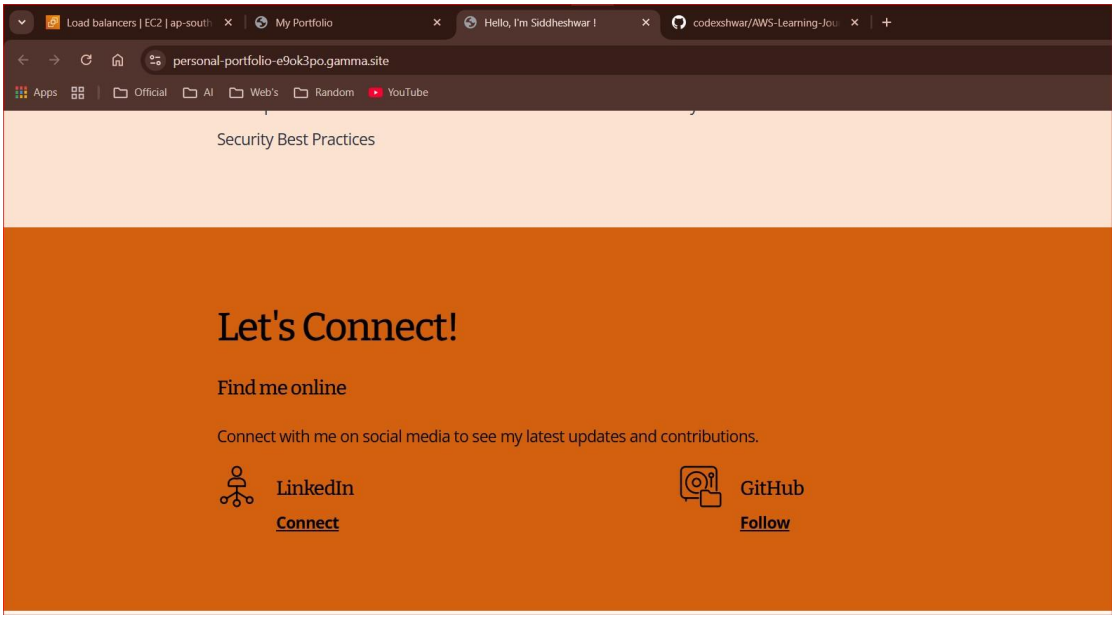
6.Successfully Deployed :



2)


Load balancers | EC2 | ap-south
My Portfolio
Hello, I'm Siddheshwar !
codexshwar/AWS-Learning-Jou
personal-portfolio-e9ok3po.gamma.site
Apps
Official
AI
Web's
Random
YouTube
My Toolkit: Languages & Technologies
Cloud Services
Amazon EC2
RDS (Managed database)
Lambda (Serverless computing)
S3 (object storage & static website hosting)
Tools & Platforms
Git, VS Code, AWS CLI
Principles
Cloud-Native Design
Scalability & High Availability
Infrastructure as Code (IaC)
Cost Optimization
Security Best Practices
Programming Languages
C
C++
Java
Python

3)


Load balancers | EC2 | ap-south
My Portfolio
Hello, I'm Siddheshwar !
codexshwar/AWS-Learning-Jou
personal-portfolio-e9ok3po.gamma.site
Apps
Official
AI
Web's
Random
YouTube
Security Best Practices
Let's Connect!
Find me online
Connect with me on social media to see my latest updates and contributions.
LinkedIn
Connect
GitHub
Follow

Result :

- Successfully deployed a working website accessible from the internet.
-Traffic is routed via the Application Load Balancer ensuring availability.
- Instances are automatically managed through the target group.
-Web server installation and content setup were automated using user data.

Key Learnings :

-How to deploy a website on AWS EC2.
-Setting up and configuring target groups and load balancer.
-Automating deployments using Linux commands in user data.
-Managing security groups for controlled access.

Stay Connected :

https://www.linkedin.com/in/siddheshwar-shinde-90b73a353

https://github.com/codexshwar

Thank you.