Task 3

Objective: High Availability + Auto Scaling Architecture Deployment

Steps Performed:

1. VPC & Networking Setup

- Created a custom VPC for secure networking.
- Configured **two public subnets** across different Availability Zones.
- Configured **two private subnets** to host backend EC2 instances.
- Attached an **Internet Gateway** to enable public subnet access.
- Created route tables:
    - Public route table → IGW.
    - Private route table → NAT Gateway (for outbound access from private instances).

2. Application Load Balancer (ALB) Setup

- Created an **Internet-facing Application Load Balancer** across two public subnets.
- Configured a security group for ALB allowing:
    - HTTP (80) from anywhere.
    - HTTPS (443) (optional).
- Created a **Target Group** using Instance target type.
- Configured health checks using:
    - Protocol: HTTP
    - Path: /
- Attached the Target Group to the ALB Listener (HTTP:80).

3. EC2 Launch Template Configuration

- Created a Launch Template defining:
    - Amazon Linux 2023 AMI (Free Tier eligible)
    - Instance type: t2.micro
    - Security group allowing port 80 **only from ALB security group**
    - User data script to install and run a simple web server:
    - #!/bin/bash
    - dnf install -y httpd
    - systemctl start httpd
    - systemctl enable httpd
    - echo "Hello from ASG Instance" > /var/www/html/index.html
- Ensured **Auto-assign public IP was disabled** (private EC2 instances).

4. Auto Scaling Group (ASG) Deployment

- Created an ASG using the Launch Template.
- Selected **two private subnets** across different AZs for High Availability.
- Attached the ASG to the previously created Target Group.
- Configured scaling parameters:
    - Minimum capacity: 1
    - Desired capacity: 2
    - Maximum capacity: 3
- Enabled **ELB health checks** to ensure unhealthy instances are replaced automatically.

5. High Availability Validation

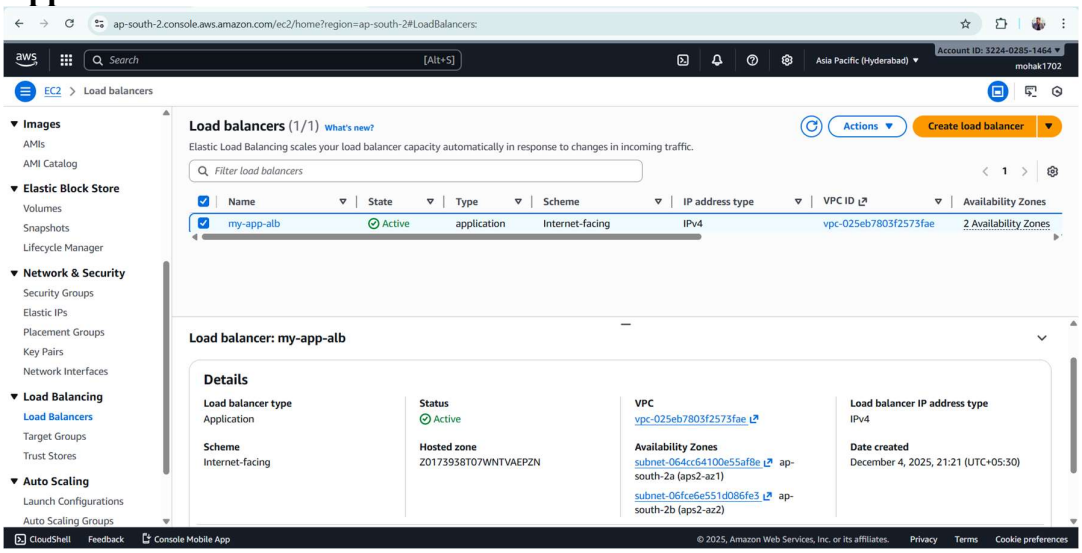
- EC2 instances launched automatically inside private subnets.
- ALB performed health checks and routed traffic only to **healthy instances**.
- Terminated one instance to validate:
    - ASG automatically launched a replacement instance.
    - ALB continued routing traffic without downtime.

Outcome:

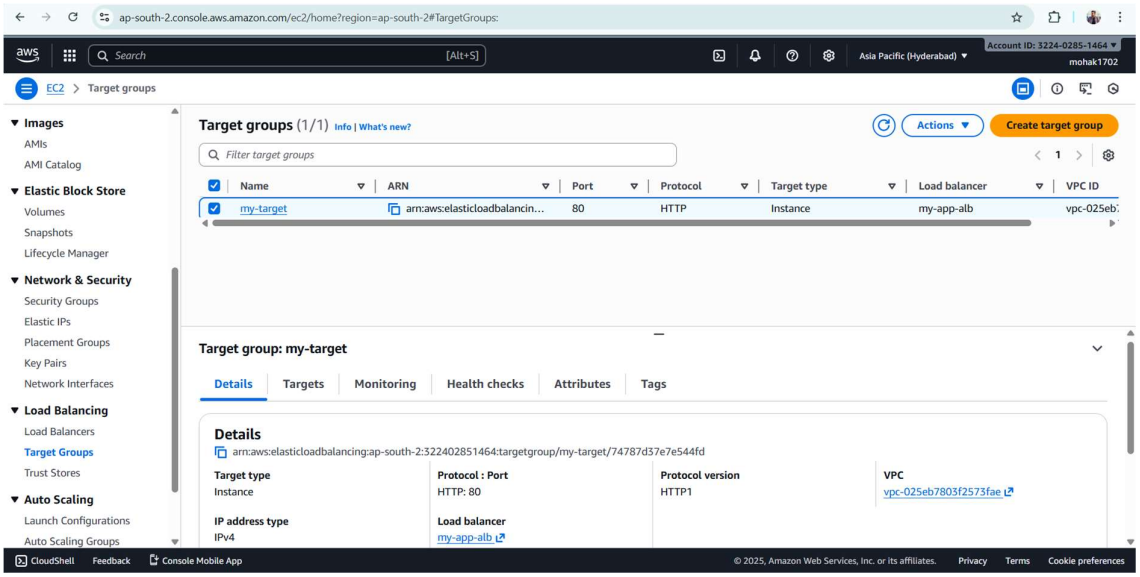
- A fully functional **Highly Available, Auto-Scaling architecture** is deployed.
- Traffic successfully flows:
  Internet → ALB → Target Group → Private EC2 Instances (ASG)
- The application remains accessible even if an AZ or instance fails.
- EC2 instances are secure in private subnets, reachable only through ALB.
- System is now scalable, fault-tolerant, and production-ready.

Screenshots

1. Application Load Balancer



2. Target Group


ap-south-2.console.aws.amazon.com/ec2/home?region=ap-south-2#TargetGroups:
Search
[Alt+S]
Asia Pacific (Hyderabad)
Account ID: 3224-0285-1464
mohak1702
EC2 > Target groups
Images
AMIs
AMI Catalog
Elastic Block Store
Volumes
Snapshots
Lifecycle Manager
Network & Security
Security Groups
Elastic IPs
Placement Groups
Key Pairs
Network Interfaces
Load Balancing
Load Balancers
Target Groups
Trust Stores
Auto Scaling
Launch Configurations
Auto Scaling Groups
Target groups (1/1) Info | What's new?
Actions
Create target group
Filter target groups
Name
ARN
Port
Protocol
Target type
Load balancer
VPC ID
my-target
arn:aws:elasticloadbalancin...
80
HTTP
Instance
my-app-alb
Target group: my-target
Details
Targets
Monitoring
Health checks
Attributes
Tags
Details
arn:aws:elasticloadbalancing:ap-south-2:322402851464:targetgroup/my-target/74787d37e7e544fd
Target type
Instance
Protocol : Port
HTTP: 80
Protocol version
HTTP1
VPC
vpc-025eb7803f2573fae
IP address type
IPv4
Load balancer
my-app-alb
CloudShell
Feedback
Console Mobile App
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences