BookTable Application – AWS Deployment Report

Project Summary

This report outlines the deployment process of the BookTable MERN stack application on AWS using EC2 instances and an Application Load Balancer for high availability and scalability.

Deployment Steps and Implementation

1. Initial Setup

- We began with a fully functional EC2 instance (`nextgen`) where our backend server was deployed and tested successfully.
- This EC2 instance was running in the `us-east-1d` availability zone, as part of our VPC setup.

2. Creating Machine Images (AMIs)

- To ensure scalability and high availability, we created **two Amazon Machine Images (AMIs)** from the working EC2 instance (`nextgen`).
- These AMIs were then used to launch two new EC2 instances:
    - `nextgen` (original)
    - `next-Gen2` (duplicate from AMI)

3. Launching EC2 Instances

- Both instances were launched in different subnets under the same VPC for load balancing.
- Application servers were deployed and started on both instances, running on **port 80 (HTTP)**.

4. Load Balancer Setup

- An **Application Load Balancer (ALB)** named `booktableLb` was created.
- Scheme: `Internet-facing`
- Listener: **HTTP on port 80**
- It was configured to:

    - Forward traffic to a **target group** that includes both EC2 instances (`nextgen`, `next-Gen2`)

    - Automatically distribute requests to healthy instances based on round-robin routing.

5. Health Checks

- The load balancer performs regular health checks on both instances to ensure only healthy targets serve traffic.
- Port 80 was verified to be open in the security groups for HTTP traffic.

6. DNS and Access

A DNS name was generated by AWS:
`booktablelb-1368877047.us-east-1.elb.amazonaws.com`

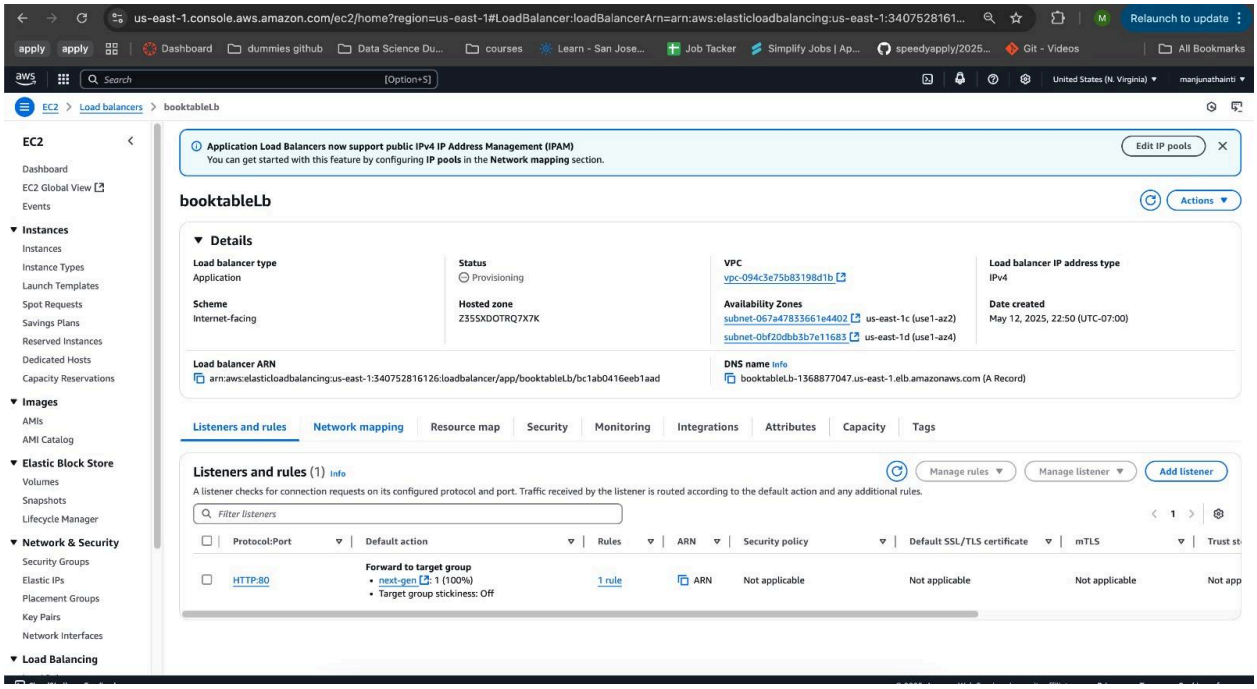
- This endpoint is used to access the frontend and backend of the application through the load balancer.

7. UI Validation

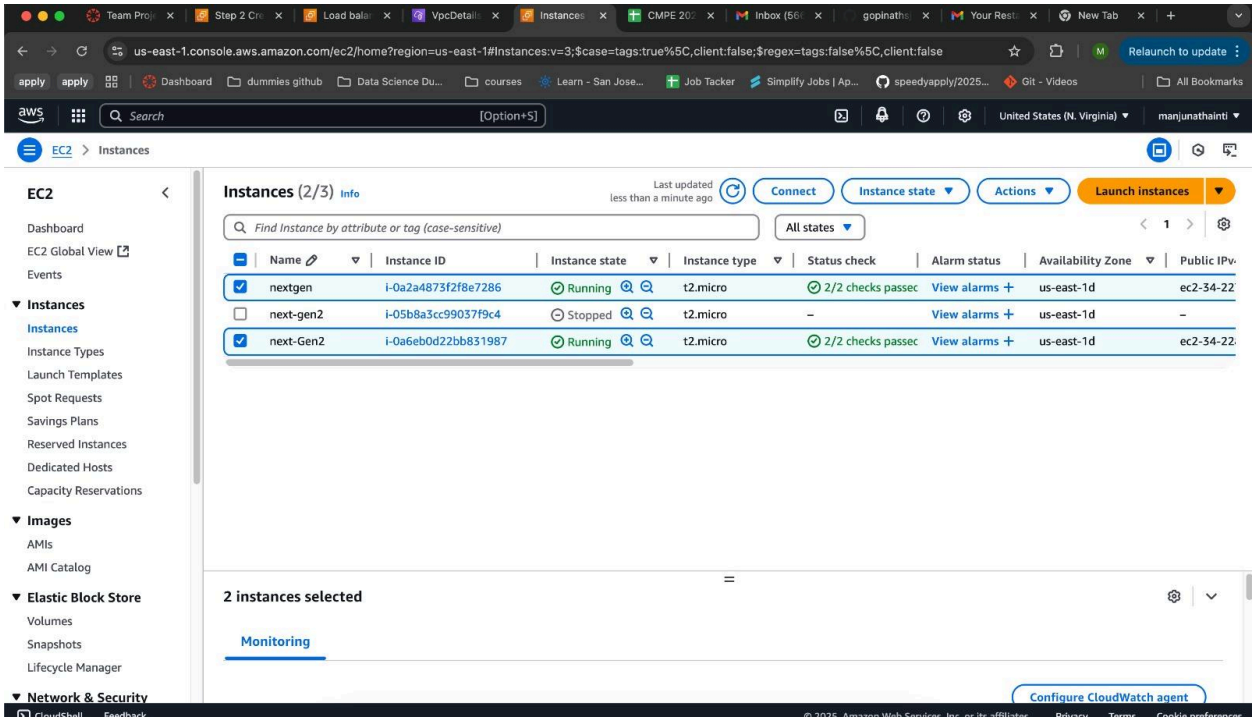
- Accessing the DNS confirms that the application loads successfully and routes to either instance behind the load balancer.
- The UI displays `My Bookings`, `Search`, and other features seamlessly through the load-balanced setup.

Attached Screenshots

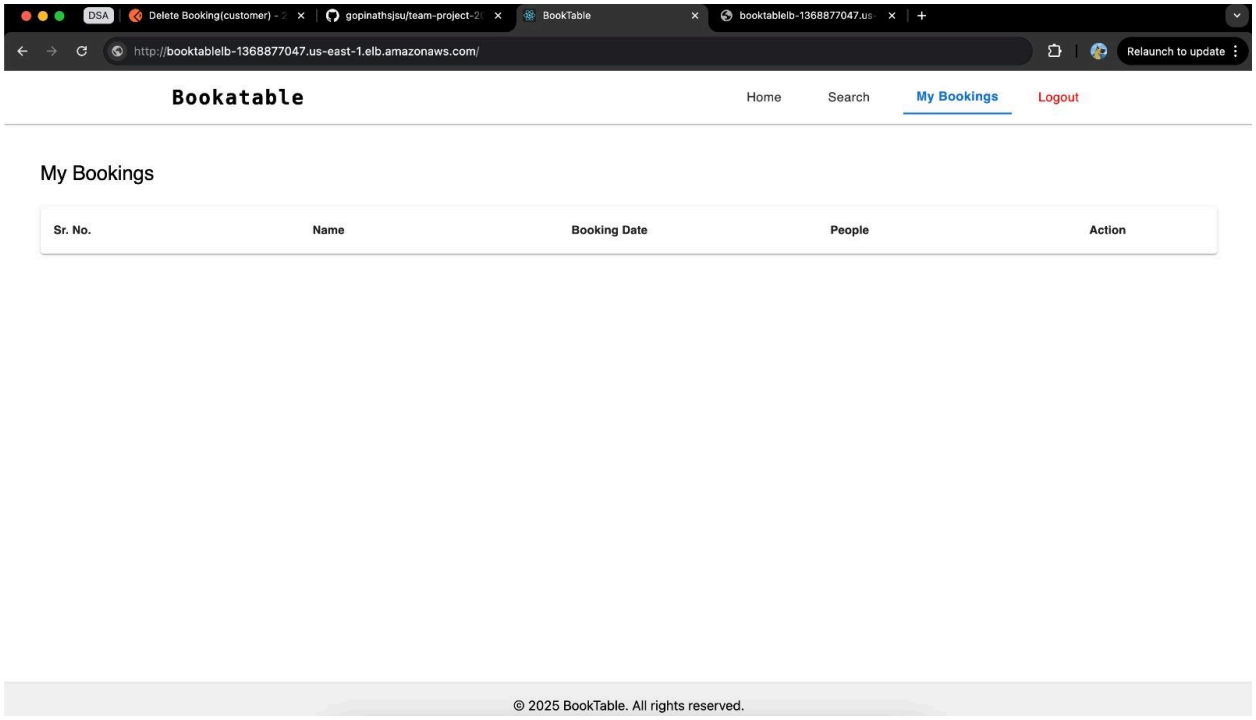
1. **Load Balancer Configuration** – Showing listener rules and target group mapping.
2. **EC2 Instance List** – Showing the original and duplicated EC2 instances running.
3. **Live Application UI** – Accessed via the Load Balancer DNS and displaying the frontend.



Load Balancer Configuration



EC2 Instance List



Live Application UI

Conclusion

We have successfully deployed a scalable version(using AWS Auto Scaling) of the BookTable app using:

- EC2 Instances from AMIs,
- Application Load Balancer,
- Auto DNS routing.

This setup ensures **fault tolerance**, **scalability**, and **easy maintenance** of the application.