**Description**

**Introduction:**

The TravelMemory application has been developed using the MERN stack. Your challenge is to deploy this application on an Amazon EC2 instance. This will provide you with hands-on experience in deploying full-stack applications, working with cloud platforms, and ensuring scalable architecture.

**Project Repository:**

Access the complete codebase of the TravelMemory application from the provided GitHub link: <https://github.com/UnpredictablePrashant/TravelMemory>

**Objective:**

- Set up the backend running on Node.js.

- Configure the front end designed with React.

- Ensure efficient communication between the front end and back end.

- Deploy the full application on an EC2 instance.

- Facilitate load balancing by creating multiple instances of the application.

- Connect a custom domain through Cloudflare.

**Tasks:**

**1. Backend Configuration:**

- Clone the repository and navigate to the backend directory.

- The backend runs on port 3000. Set up a reverse proxy using nginx to ensure smooth deployment on EC2.

- Update the *.env* file to incorporate database connection details and port information.

**2. Frontend and Backend Connection:**

- Navigate to the `urls.js` in the frontend directory.

- Update the file to ensure the front end communicates effectively with the backend.

**3. Scaling the Application:**

- Create multiple instances of both the frontend and backend servers.

- Add these instances to a load balancer to ensure efficient distribution of incoming traffic.

**4. Domain Setup with Cloudflare:**

- Connect your custom domain to the application using Cloudflare.

- Create a CNAME record pointing to the load balancer endpoint.

- Set up an A record with the IP address of the EC2 instance hosting the front end.

**5. Documentation:**

- Prepare comprehensive documentation detailing each step of the deployment process. Include relevant screenshots to make the process clear and reproducible.

- Design a deployment architecture diagram using [draw.io](https://www.draw.io/) to visualize the flow and connections.

**Expected Deliverables:**

1. A fully functional TravelMemory application hosted on an EC2 instance, accessible via a custom domain.

2. Detailed documentation of the deployment process with appropriate screenshots.

3. A deployment architecture diagram.

**Evaluation Criteria:**

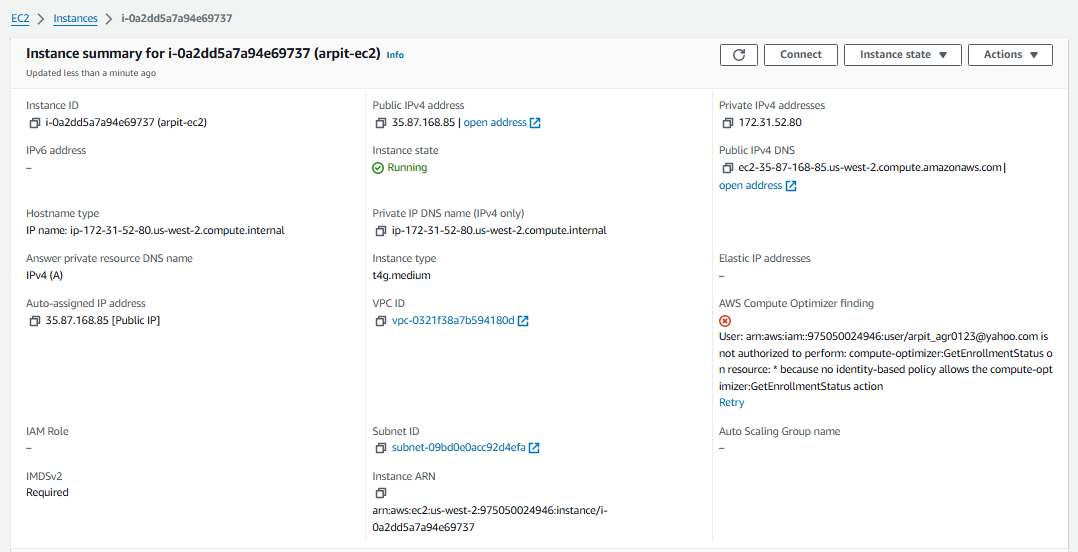
- Accuracy and effectiveness of the deployment.

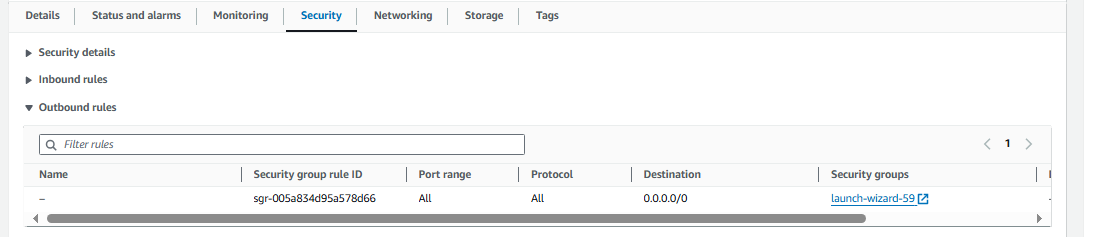
- Clarity and comprehensiveness of the documentation.

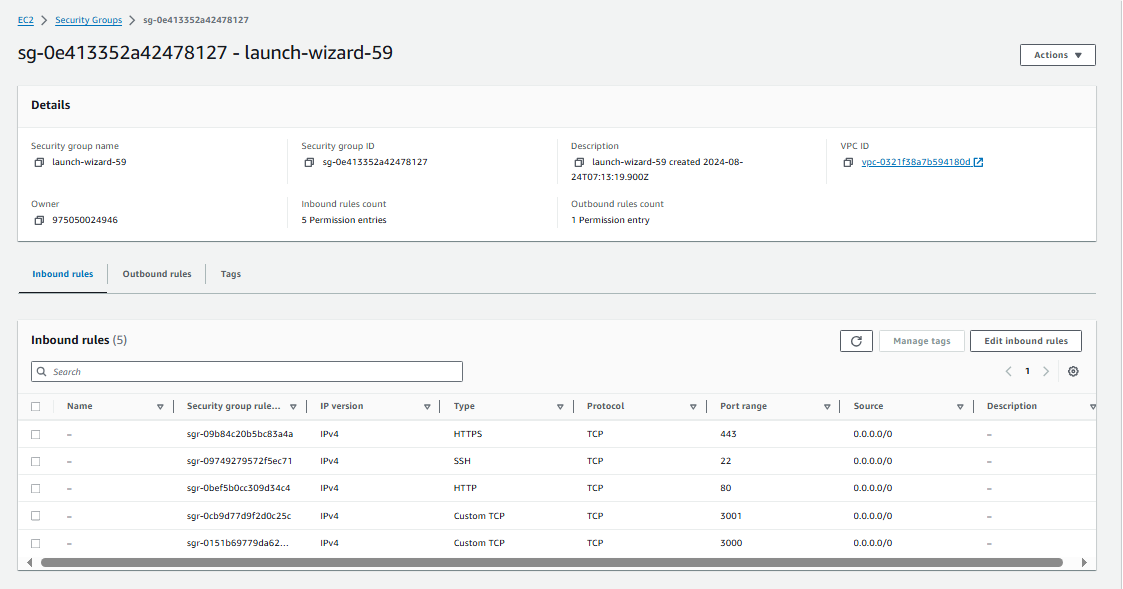
- Adherence to best practices in terms of security, scalability, and resilience.

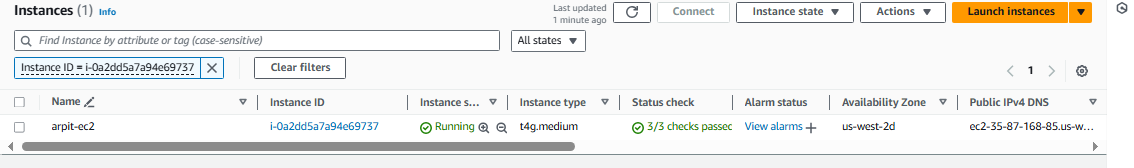
**Hints:**

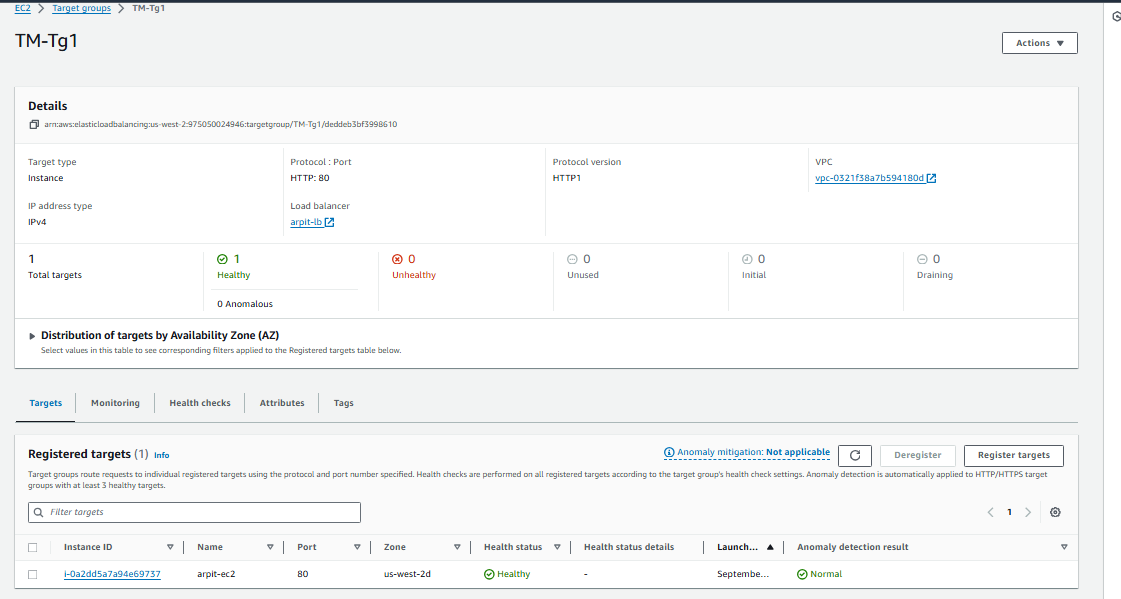
While setting up with Cloudflare, remember that a CNAME record is essential for linking the load balancer endpoint. An A record, on the other hand, connects the EC2 instance via its IP address.

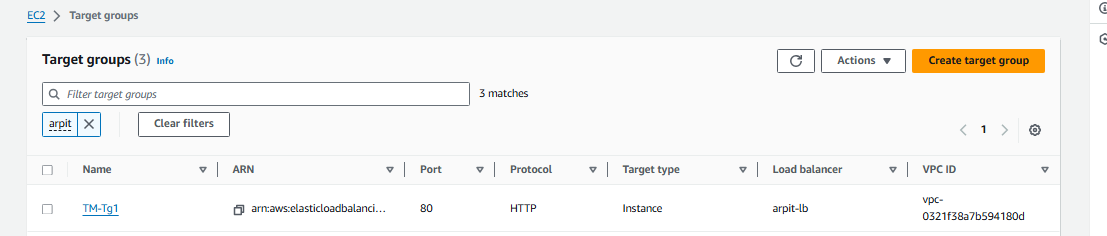


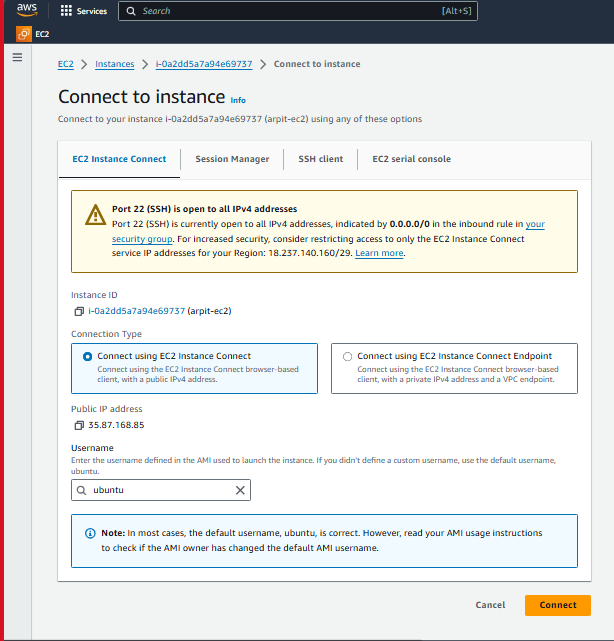


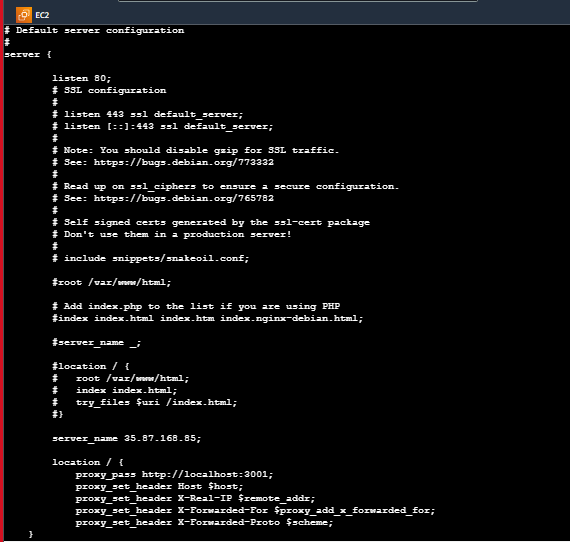




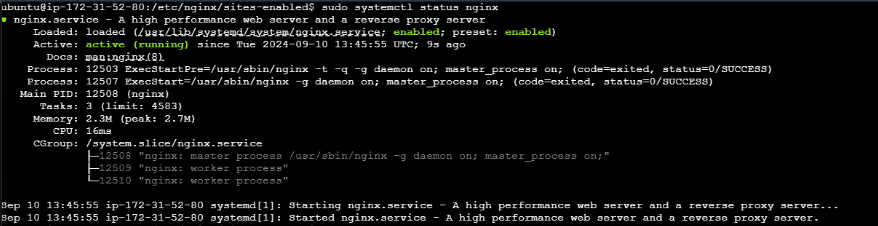






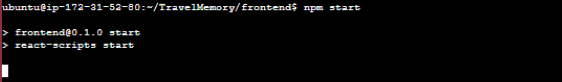


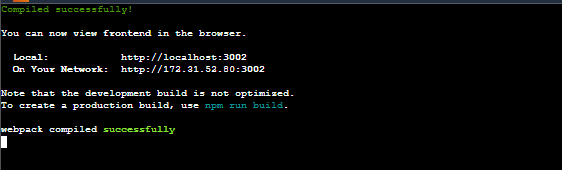


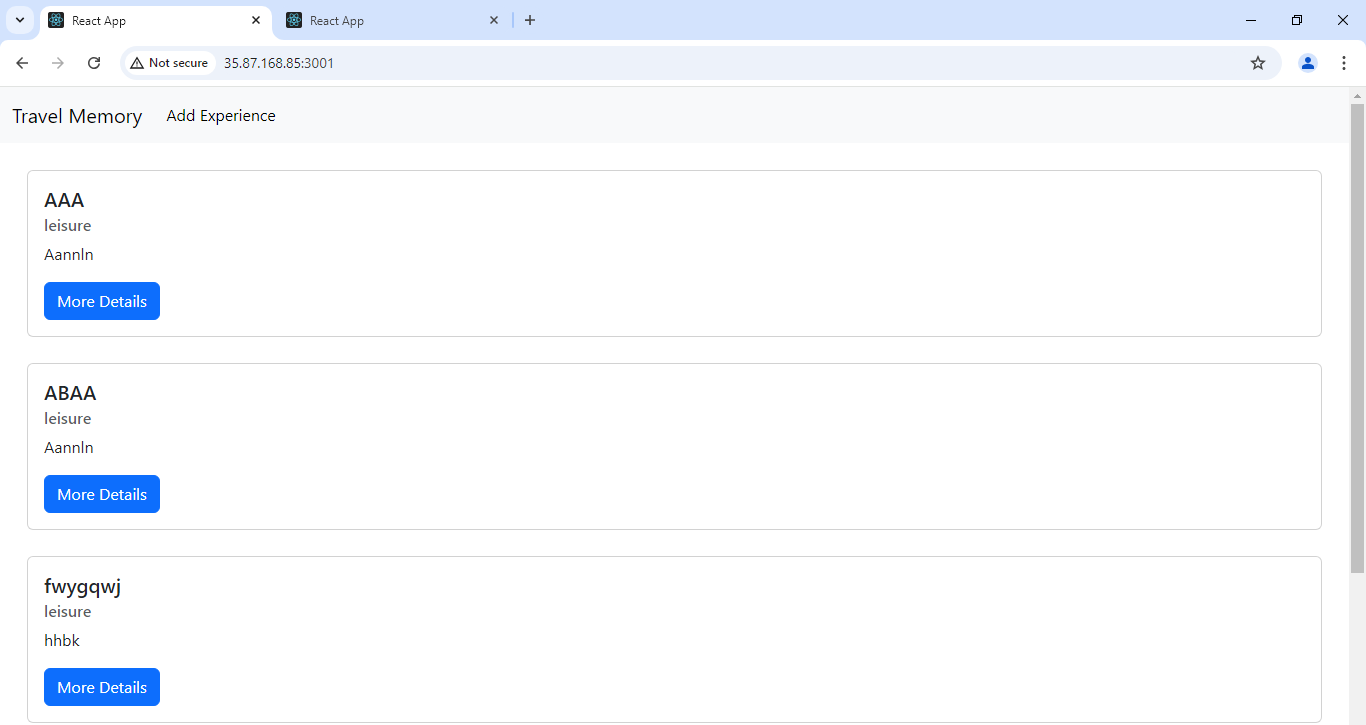


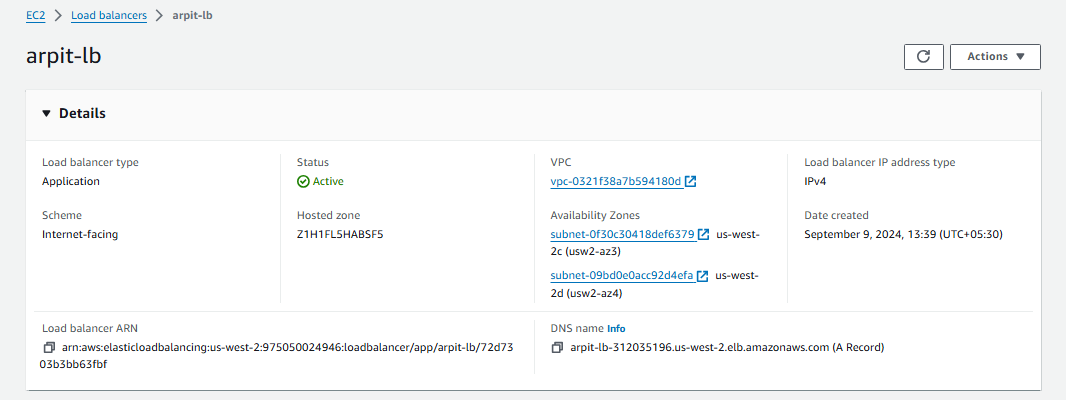


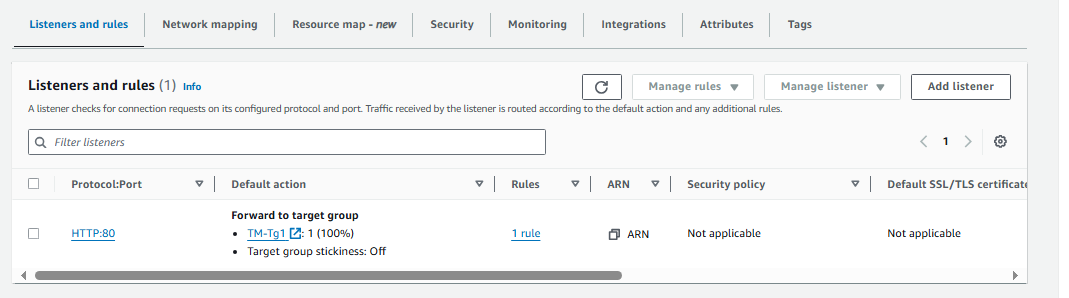


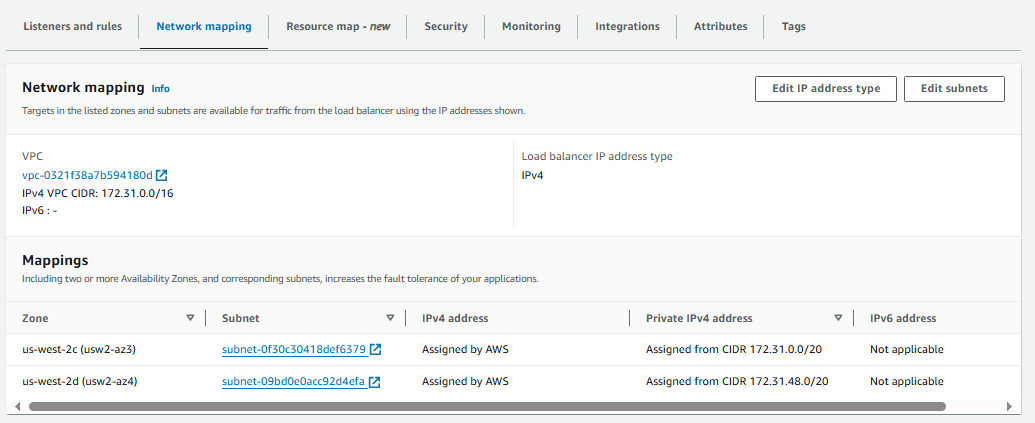


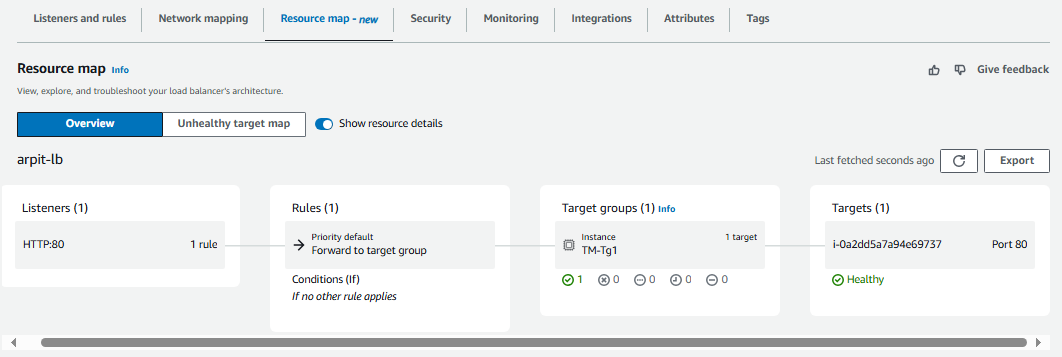


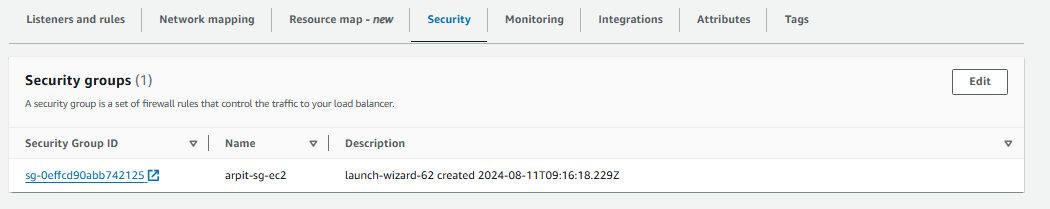












Purchased a domain name from GoDaddy:

DNS: ‘sudarshanrestaurant.live’

Configured the Nameserver

