# Travel Memory Application Deployment

The Travel Memory 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:**

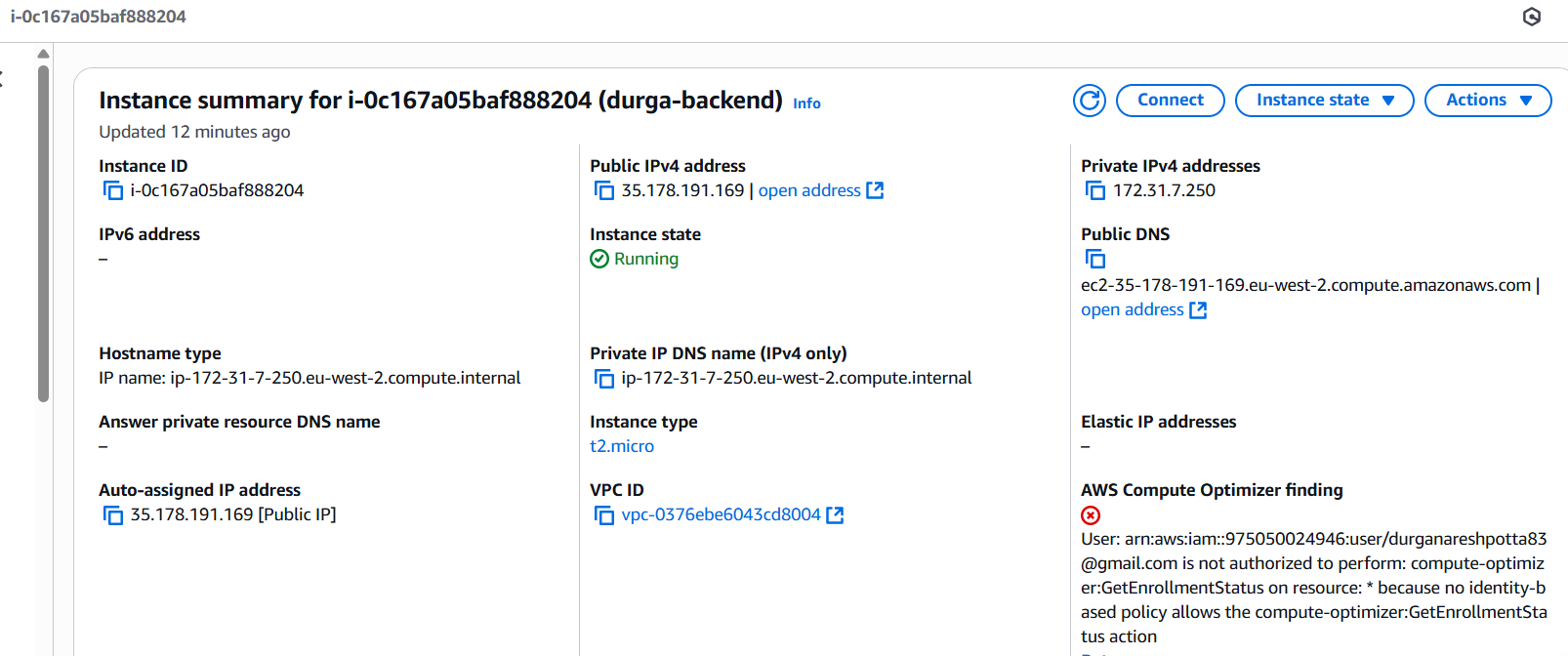
Access the complete codebase of the TravelMemory application from the provided GitHub link: [https://github.com/UnpredictablePrashant/TravelMemory](https://github.com/UnpredictablePrashant/TravelMemory" \o "https://github.com/UnpredictablePrashant/TravelMemory" \t "https://vlearnv.herovired.com/page-activity/357/assign/_blank)

Here are the step-by-step instructions to deploy the Travel Memory MERN application on AWS EC2, set up load balancing, and connect a custom domain via Cloudflare.

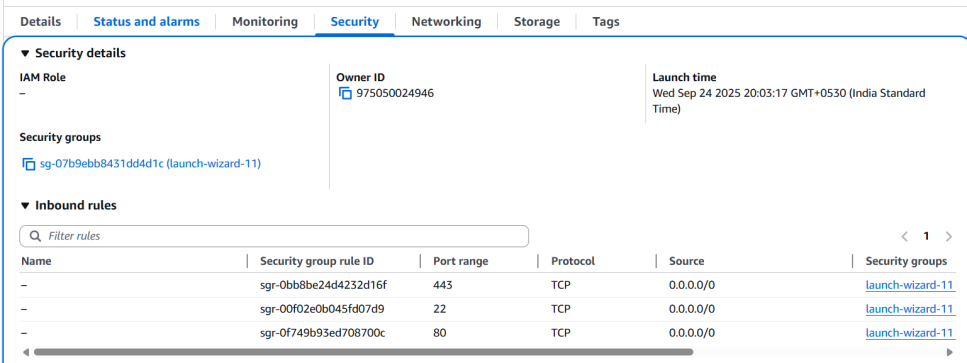
### Backend Configuration:

#### a. Launch EC2 Instance:

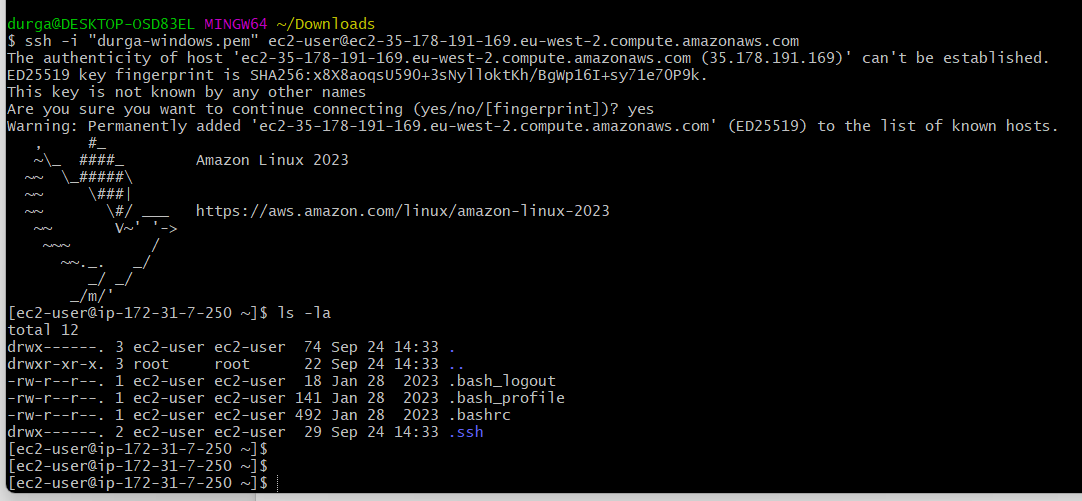
* Log in to your AWS Console and start a new EC2 instance
* Choose an instance type



* Configure security group: open ports 22 (SSH), 80 (HTTP), and 443 (HTTPS).



* Connect to the server via SSH.



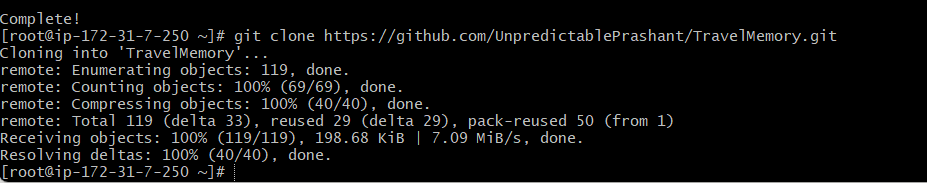
#### **b. System Setup:**

* Update the system: sudo yum update && sudo yum upgrade -y
* Install Node.js and npm:
  + dnf search nodejs - **Check available Node.js versions**
  + sudo dnf install -y nodejs18
  + Sudo yum install git

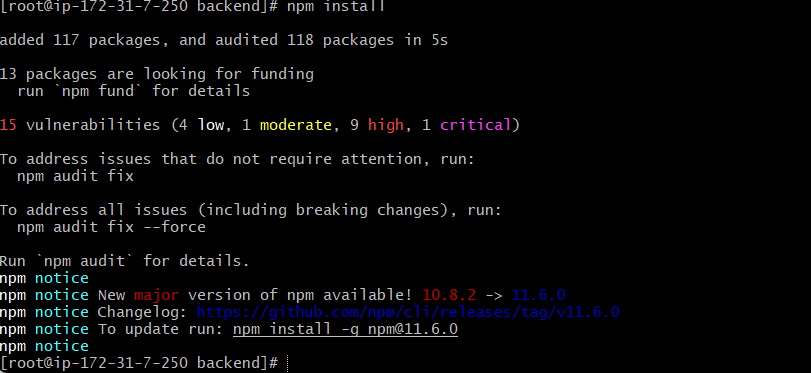
Once node js and git installations are done, clone the git repository.

#### **c.Clone and Configure Backend:**

* Clone the repository:
  + git clone <https://github.com/UnpredictablePrashant/TravelMemory.git>



* cd TravelMemory/backend
* Install dependencies: npm install



* Create and update .env with database URI and port info
  + MONGO\_URI=mongodb+srv://username:password@cluster.mongodb.net/dbname
  + PORT=3001

#### **d. Set Up NGINX Reverse Proxy:**

* Install NGINX
* Configure NGINX for reverse proxy
  + sudo nano /etc/nginx/conf.d/durga.conf
    - server {

listen 80;

server\_name \_;

location / {

proxy\_pass http://127.0.0.1:3000;

proxy\_http\_version 1.1;

proxy\_set\_header Upgrade $http\_upgrade;

proxy\_set\_header Connection 'upgrade';

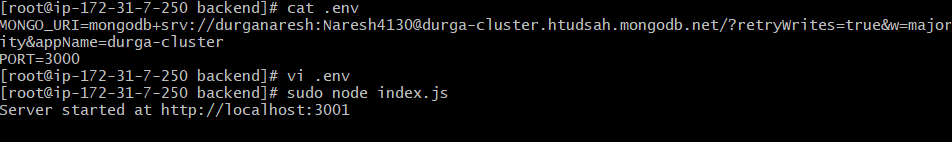
proxy\_set\_header Host $host;

proxy\_cache\_bypass $http\_upgrade;

}

}

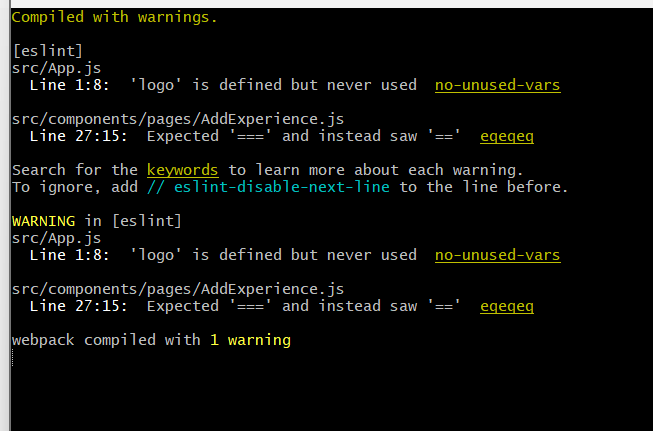
* + Once updated the configuration restart the nginx.
    - sudo systemctl restart nginx
* Start Backend process manager
  + npm install -g pm2
  + pm2 start index.js --name travel-memory-backend
  + pm2 startup
  + pm2 save



### Frontend Configuration:

1. Configure frontend:
   1. cd frontend
   2. npm install
   3. Update the url.js to specify the address of your backend to frontend
      1. nano url.js
      2. Replace the localhost with public ip of ec2 instance.
2. Build the Frontend:

* npm install
* npm run build
* sudo cp -r /home/ubuntu/TravelMemory/frontend/build/\* /var/www/html/

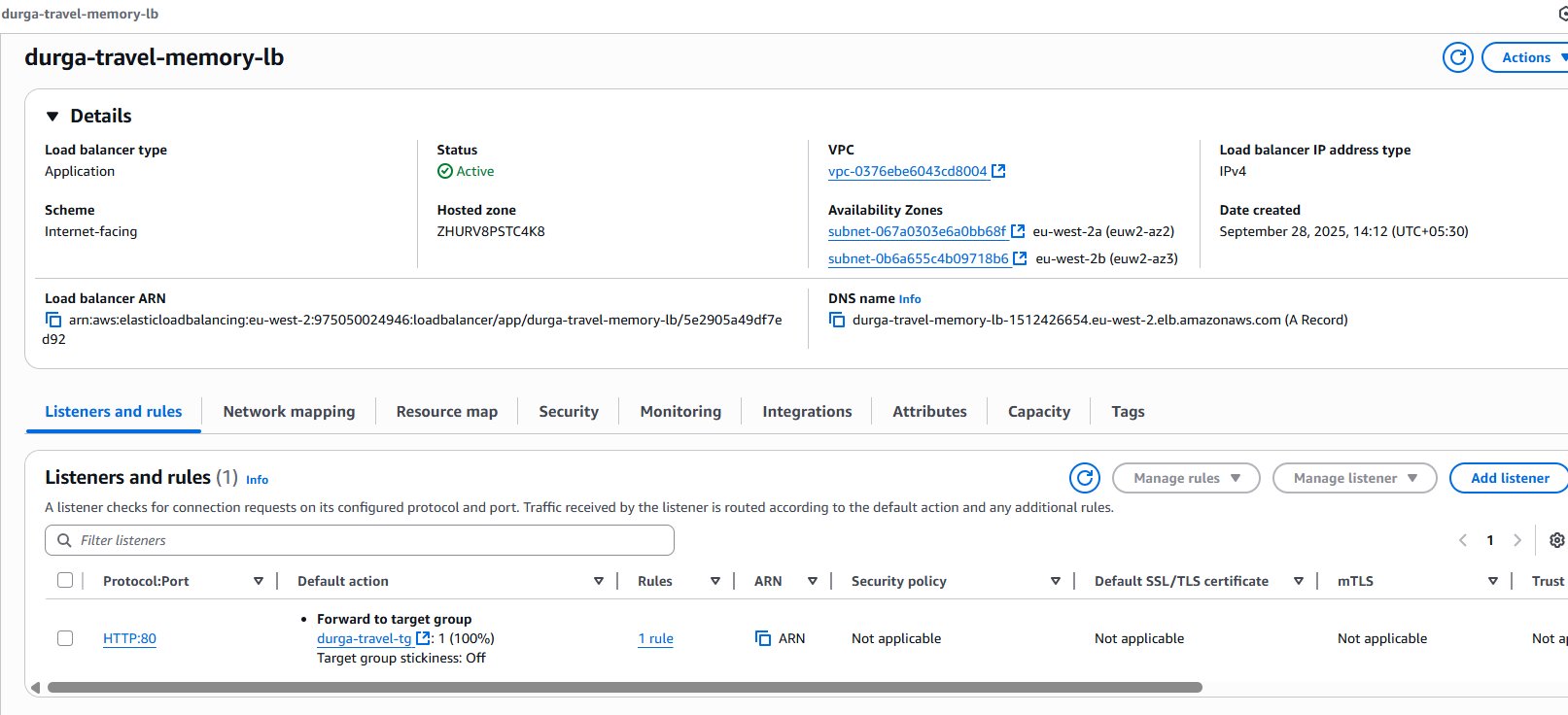


## Scaling the Application:

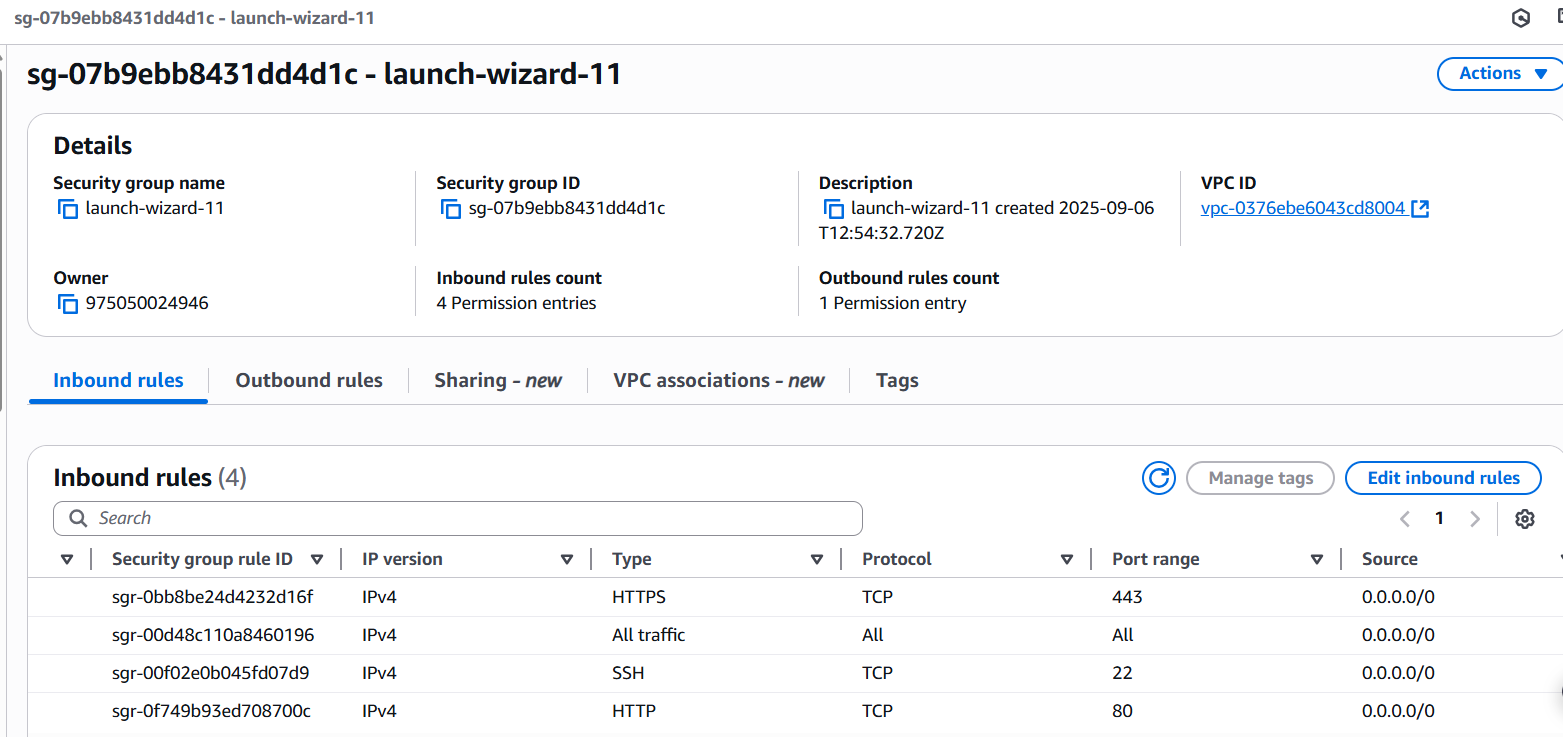
1. Launch Multiple EC2 Instances:

* Repeat the above steps to create identical backend/frontend servers on multiple EC2 instances.

1. Configure AWS Elastic Load Balancer (ELB):
   * 1. Go to EC2 console > Load Balancers > Create Load Balancer.
     2. Choose Application Load Balancer.
     3. Add your EC2 instances to the target group.
     4. Configure health checks (path: /api/health if available).
     5. Set listeners for HTTP (80)



1. Update Security Groups:
   1. Ensure all connected EC2 instances allow traffic from the ELB.



## Domain Setup with Cloudflare:

1. Register and Add Domain to Cloudflare:
   1. sign up at cloudflare.com and add your domain.
   2. Update your domain registrar to use Cloudflare’s nameservers

#### Step 1: Add Your Domain to Cloudflare

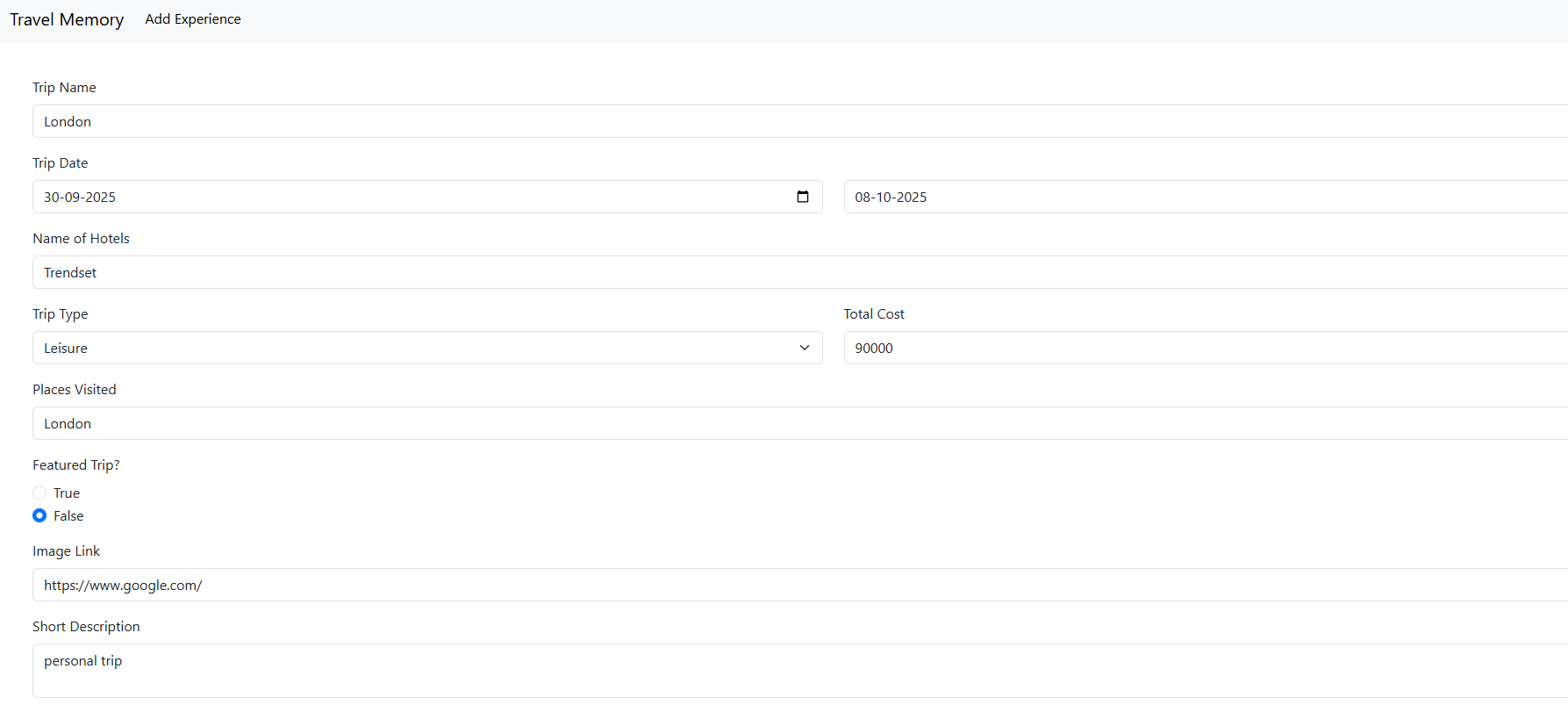
* + 1. Log in to Cloudflare Dashboard
    2. Click Add a Site.
    3. Enter your domain (e.g., mytravelmemory.com).
    4. Select the Free plan (or whichever you need).

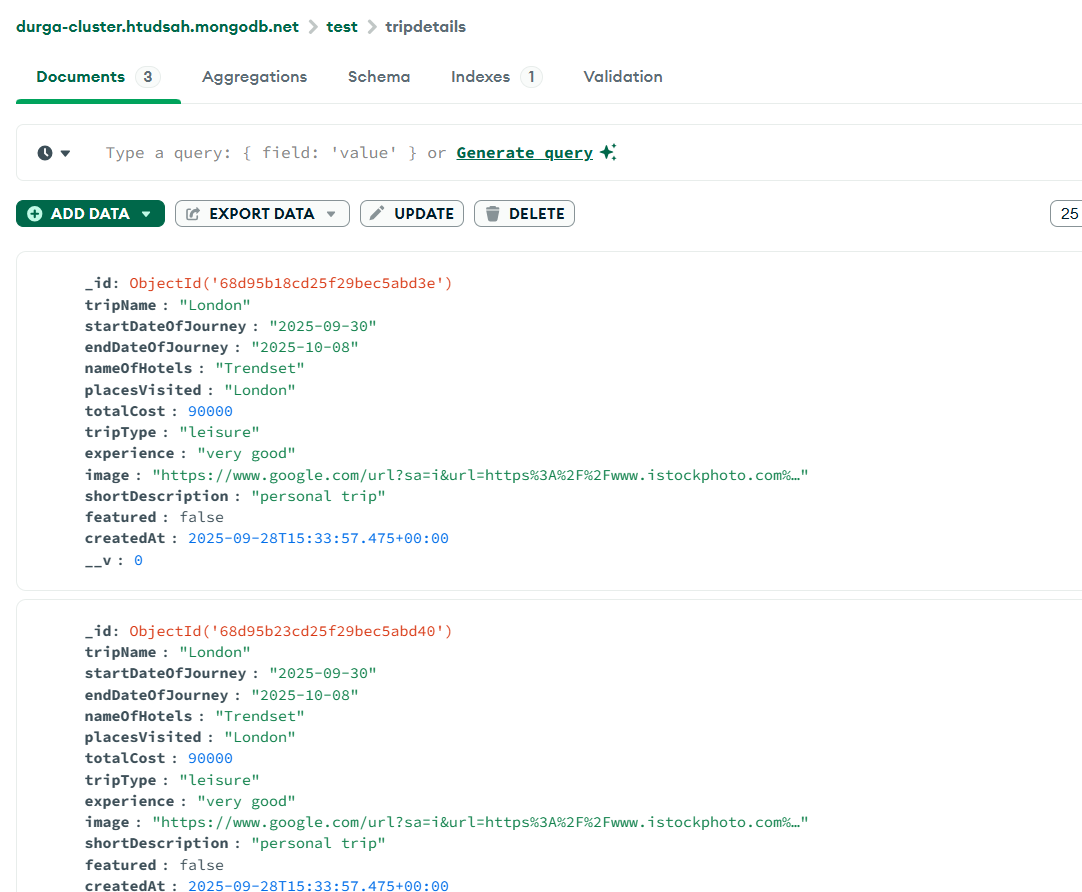
#### Step 2: Set Up DNS Records in Cloudflare

* + 1. Now add the required DNS records:
    2. ✅ Add CNAME for ALB
    3. Go to DNS → Records → Add record
    4. Type: CNAME
    5. Name: [www.durgatravelmemory.com](http://www.durgatravelmemory.com)
    6. Target: <your-alb-dns-name>.elb.amazonaws.com
    7. TTL: Auto
    8. Proxy status: DNS only (gray cloud) – safer for ALB to avoid proxy/TLS conflicts.

#### ✅ Add A Record for EC2 Front-End

* + 1. Add another record:
    2. Type: A
    3. Name: @ (root domain) OR frontend if you want it on a subdomain
    4. IPv4 address: <your-ec2-public-ip>
    5. TTL: Auto
    6. Proxy status:
       1. DNS only if you want Cloudflare to just resolve the IP.





# Architecture Diagram

