**Muhammad Uzair**

**29414  
MS Data Science**

**DevOps Semester Project**

**Key Links:**

Dockerized App  
  
<https://app.codelessops.site/>

Metabase App  
<https://bi.codelessops.site/>

Github Repo  
<https://github.com/muhammaduzair99/nodeapp_uzair>

DockerHub Repo

<https://hub.docker.com/repository/docker/uzair99/nodeapp-project/general>

Video Link:  
<https://www.loom.com/share/a80efbe3184643518c73a8cd911beb32?sid=eac72e1e-a494-4a06-a506-0f931c4eae51>

**Project Report: Secure Analytics Infrastructure with Dockerized Metabase on AWS**

**1. Project Overview**

This project involved designing and deploying a scalable, secure, and production-grade analytics infrastructure on AWS. The core components included:

* Dockerized **Metabase** as a BI tool
* **RDS MySQL and PostgreSQL** for analytics datasets
* **EC2 Auto Scaling Group** for application backend
* **Application Load Balancer (ALB)** with HTTPS
* **Nginx reverse proxy** and **Certbot SSL**
* Infrastructure-as-Code using **Terraform**

**2. Objectives**

* Build a cost-effective, containerized analytics infrastructure
* Ensure data security using HTTPS and restricted database access
* Enable scalable deployments with Auto Scaling EC2 and ALB
* Connect and visualize data from both MySQL and PostgreSQL in Metabase

**3. Technologies Used**

| **Component** | **Technology** |
| --- | --- |
| Cloud Provider | AWS (EC2, RDS, ALB, Route 53, IAM) |
| Containerization | Docker, Docker Compose |
| Web Server & SSL | NGINX, Certbot (Let's Encrypt) |
| Infrastructure Code | Terraform (~> v1.5.0) |
| Backend App | Node.js (Dockerized, exposed on port 3000) |
| Database Engines | Amazon RDS (MySQL & PostgreSQL) |
| Analytics | Metabase (Dockerized) |

**4. Architecture Diagram**

Users

|

HTTPS via Route 53 DNS & ACM SSL

|

Amazon ALB (HTTP 80 + HTTPS 443)

|

----------------------------------------------

| |

EC2 Auto Scaling Group EC2 (Metabase)

(Dockerized Node.js App) (Dockerized Metabase + NGINX + SSL)

|

RDS MySQL & PostgreSQL (Private Subnet)

**5. Implementation Steps**

**A. RDS Setup**

* Created **PostgreSQL** and **MySQL** instances via terraform.
* Configured database users, security groups, and external access.

**B. Dockerized Node.js Application**

* Dockerized an Express.js app with endpoints /, /mysql, and /pgsql.
* Configured the app to connect securely to both databases using .env.

**C. Auto Scaling & Load Balancer**

* Configured a Launch Template and Auto Scaling Group for Node.js backend.
* ALB forwards HTTPS traffic to EC2s running the app via port 3000.
* SSL enabled using AWS ACM and listener redirect from HTTP to HTTPS.

**D. Metabase EC2 Deployment (Ubuntu)**

* Created a dedicated Ubuntu EC2 instance using Terraform.
* Installed Docker, NGINX, and Certbot via user\_data script.
* Used Docker Compose to run Metabase on port 3001.
* Configured NGINX reverse proxy and issued SSL via Let's Encrypt.

**6. SSH & Tunneling for DBeaver Access**

* Created SSH tunnels from local machine to:
  + Metabase server for admin access (HTTPS)
  + PostgreSQL and MySQL RDS via bastion or public access
* Verified database connection via **DBeaver** GUI tool.

**7. Connecting Metabase to RDS**

* Logged into https://bi.codelessops.site
* Connected to both:
  + MySQL: mysql-db.c3wk6qw2orem.us-east-2.rds.amazonaws.com
  + PostgreSQL: postgres-db.c3wk6qw2orem.us-east-2.rds.amazonaws.com
* Used muzair12 as DB user, verified credentials.
* Visualized tables using Metabase's GUI (sample queries, charts).

**8. Sample Visualizations**

* Created dashboards in Metabase showing:
  + Server time from both DBs
  + Comparative row counts
  + Sample table previews
  + Time series (if applicable data present)

**9. Key Highlights**

* ✅ Full HTTPS deployment with free SSL using Certbot
* ✅ Seamless ALB routing to Dockerized app on EC2
* ✅ Reusable Terraform modules and scripts
* ✅ Docker Compose used for clean container orchestration
* ✅ Metabase running stably on Ubuntu EC2 with minimal config

**10. Challenges and Fixes**

| **Issue** | **Solution** |
| --- | --- |
| Certbot not supported on Amazon Linux 2 | Switched Metabase instance to **Ubuntu 24.04** |
| Instance IP changed on every apply | Used Route 53 DNS + ACM certificate |
| Connection issues from Metabase to RDS | Allowed port access via **Security Group** |
| Docker container restart behavior | Added restart: always in Docker Compose |
| User Data debugging | Used cat /var/log/cloud-init-output.log |

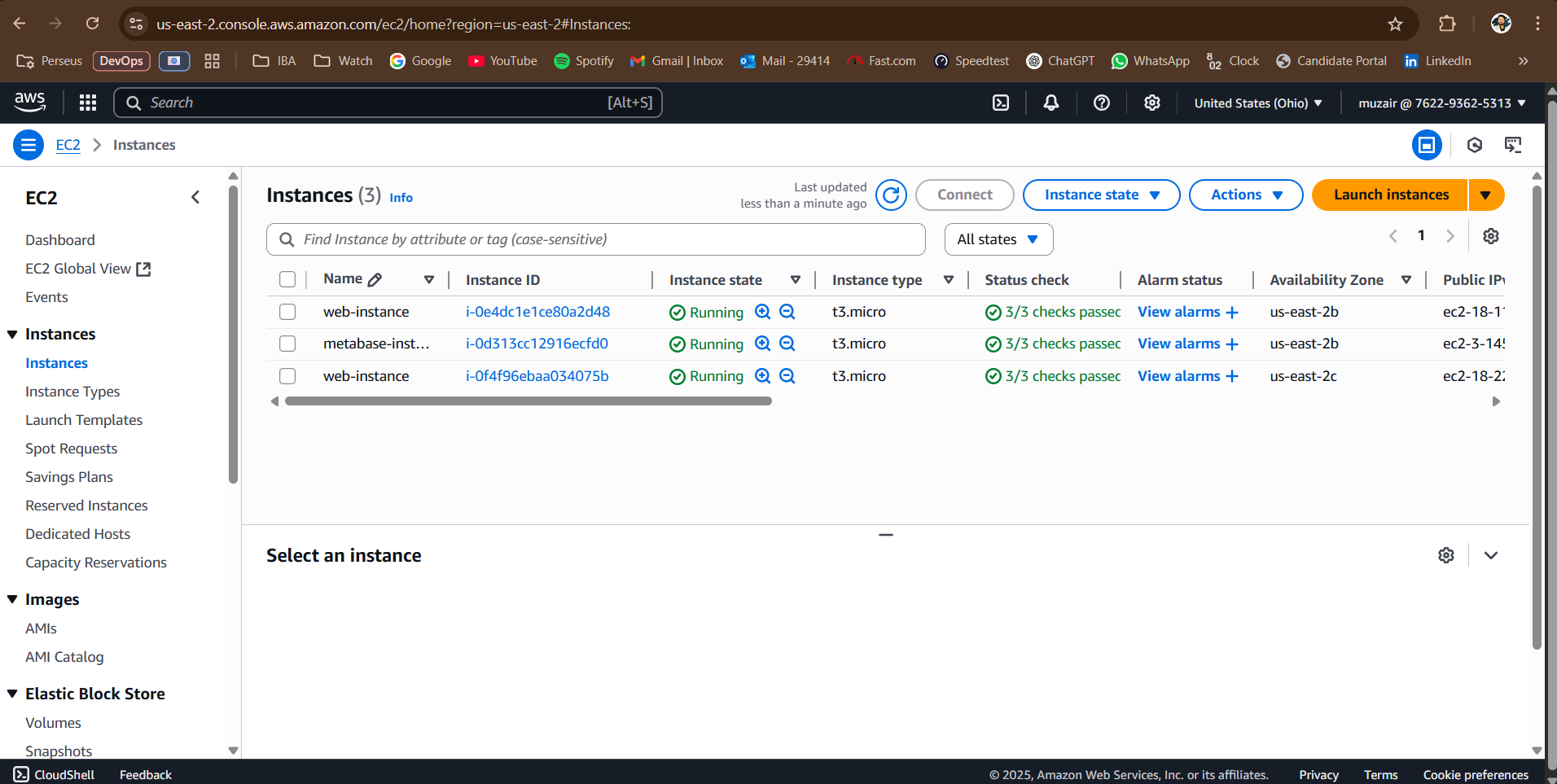
**11. Future Improvements**

* Add S3 backup for Metabase and RDS snapshots
* Auto-renew SSL certificates via system cron job
* Replace hardcoded credentials with AWS Secrets Manager
* Implement monitoring (e.g., CloudWatch or Grafana)

**12. Conclusion**

This project successfully demonstrates how to deploy a secure, scalable, and containerized analytics platform on AWS using Metabase. The full infrastructure was provisioned via Terraform, ensuring reproducibility and modularity. The final outcome provides a robust base for data visualization and business intelligence use cases in real-time or scheduled data analysis environments.

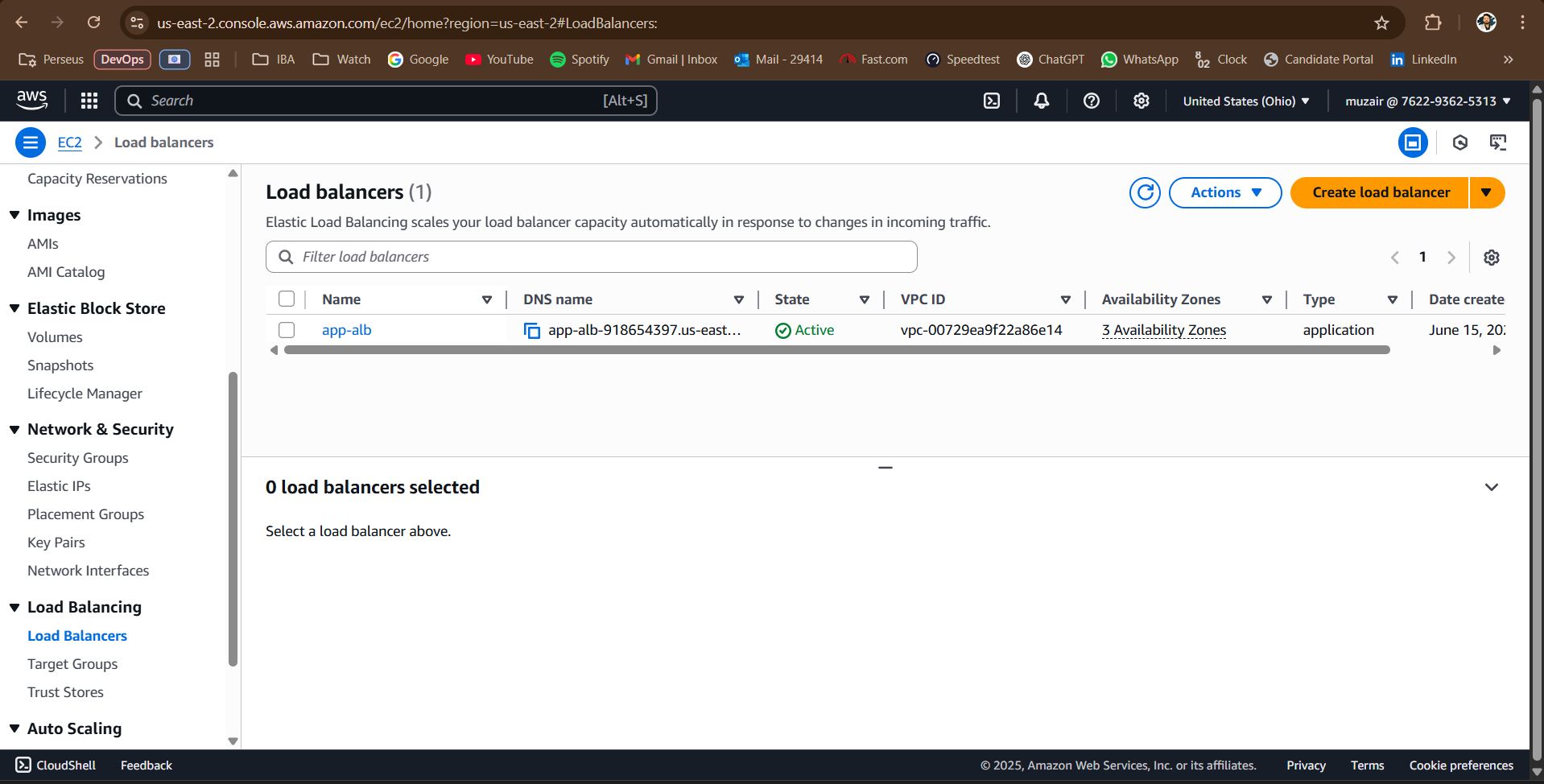
A screenshot of a computer

AI-generated content may be incorrect.  
A computer screen shot of a computer

AI-generated content may be incorrect.A computer screen shot of a chat

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A computer screen with a white background

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.  
A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.