DevOps Assignment Project Documentation

📌 Project Overview

This project demonstrates a full-stack web application deployed using modern DevOps practices. The stack includes a **FastAPI** backend and a **Next.js** frontend, both containerized with Docker, tested automatically, and deployed on **AWS ECS (Fargate)** using **Terraform** and **GitHub Actions CI/CD pipelines**.

📁 Project Structure

```
DevOps-Assignment/
├── backend/                # FastAPI backend service
│   ├── app/                # Main application code
│   │   └── main.py         # API endpoints
│   ├── tests/              # Unit tests
│   │   └── test_main.py
│   ├── Dockerfile          # Multi-stage Docker build
│   └── requirements.txt    # Python dependencies
│
├── frontend/               # Next.js frontend application
│   ├── pages/              # UI pages
│   │   └── index.js
│   ├── __tests__/          # E2E tests
│   │   └── index.test.js
│   ├── Dockerfile          # Frontend Docker build
│   ├── package.json        # Node.js dependencies
│   └── .env.local          # Environment config
│
├── terraform/              # Infrastructure as Code
│   ├── main.tf             # Main Terraform config
│   ├── variables.tf        # Input variables
│   ├── outputs.tf          # Output values
│   ├── terraform.tfvars    # Actual values for variables
│   └── provider.tf         # AWS provider config
│
└── .github/
    └── workflows/          # CI/CD pipelines
        ├── ci.yml          # CI: tests & build on develop
        └── cd.yml          # CD: deploy on main
```

⚙️ Technologies Used

- **Frontend:** Next.js (React), JavaScript
- **Backend:** FastAPI, Python
- **Containerization:** Docker
- **CI/CD:** GitHub Actions
- **Cloud:** AWS ECS (Fargate), ECR, ALB, CloudWatch
- **IaC:** Terraform
- **Monitoring:** CloudWatch Container Insights

🧪 Testing

- **Backend:** Unit tests with `pytest`
- **Frontend:** E2E tests using `@testing-library/react`
- All tests run on each push via GitHub Actions

🔄 CI/CD Pipelines

- **CI (on `develop`):**
  - Checkout code
  - Run backend & frontend tests
  - Build Docker images
  - Push to AWS ECR with `GIT_SHA` and `latest` tags

- **CD (on `main`):**
    - Deploy new Docker images to ECS services

🏗️ Infrastructure (Terraform)

- **Network:**
    - VPC with public subnets
    - Internet gateway and route tables
    - Security groups for ALB and ECS tasks
- **Compute:**
    - ECS Cluster (Fargate launch type)
    - Services for frontend and backend
- **Load Balancing:**
    - Application Load Balancer with path-based routing
    - `/api/*` to backend, `/` to frontend
- **IAM Roles:**
    - Least privilege task execution roles

🖥️ Monitoring & Alerting

- Enabled **CloudWatch Container Insights**
- Dashboard shows CPU, memory, request metrics
- Alarm: Send email if CPU > 70% for 5 mins

🛡️ Security Practices

- IAM roles with minimum required policies
- Security groups to allow inbound HTTP only on necessary ports
- Secrets stored in `.env.local` (local dev only)

🔁 Failover Demonstration

- Deployed 2+ tasks for frontend and backend
- Manually stopped 1 backend task
- ALB automatically rerouted traffic to healthy task
- Verified app remained functional

🌐 Live URLs

- **Frontend:**http://devops-alb-1451775341.us-east-1.elb.amazonaws.com
- **Backend:**http://devops-alb-1451775341.us-east-1.elb.amazonaws.com/api/message

---

📌 Final Notes

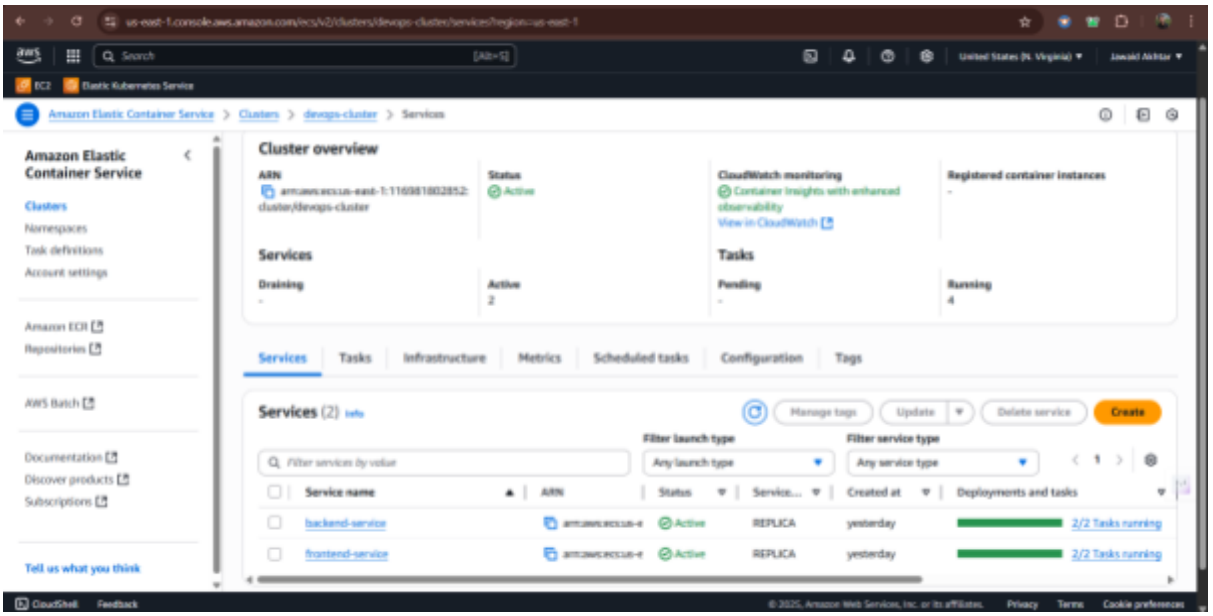
This project showcases end-to-end DevOps skills, including containerization, cloud infrastructure automation, CI/CD, health monitoring, and failover. Ready for production-grade enhancements like HTTPS, auto-scaling, and domain management.

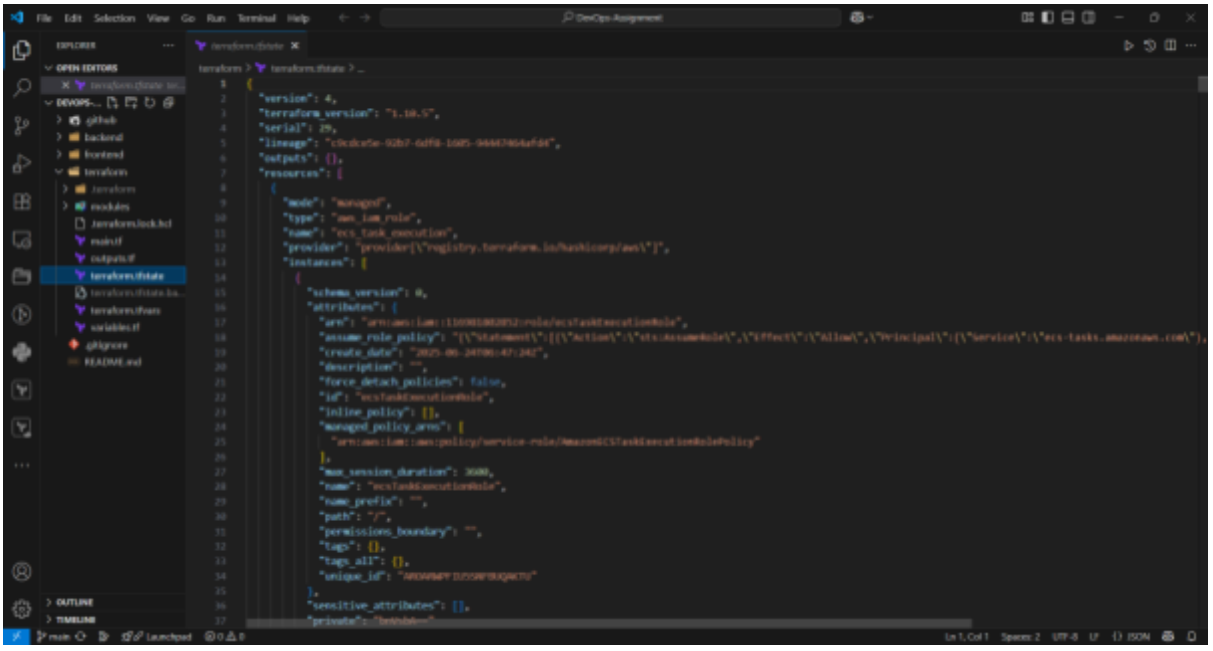
📽️ Demo Video

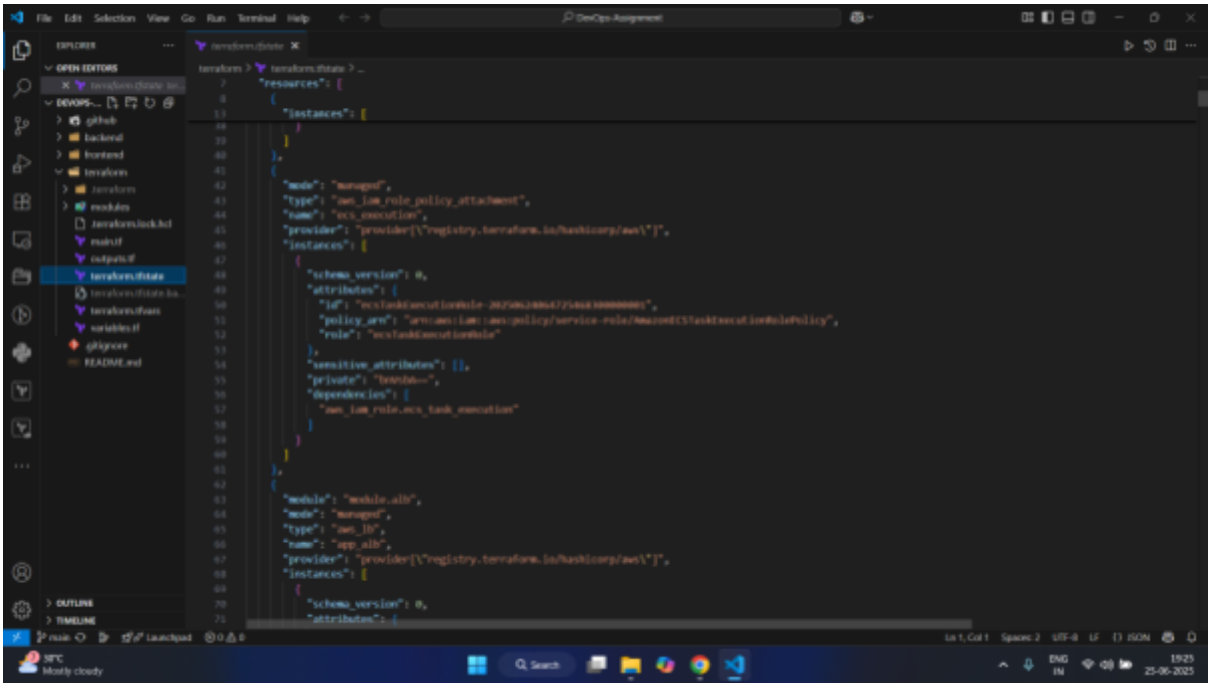
https://drive.google.com/file/d/1IfItb5vejOebxcZw6vrq2UhKX70SKiFN/view?usp=sharing

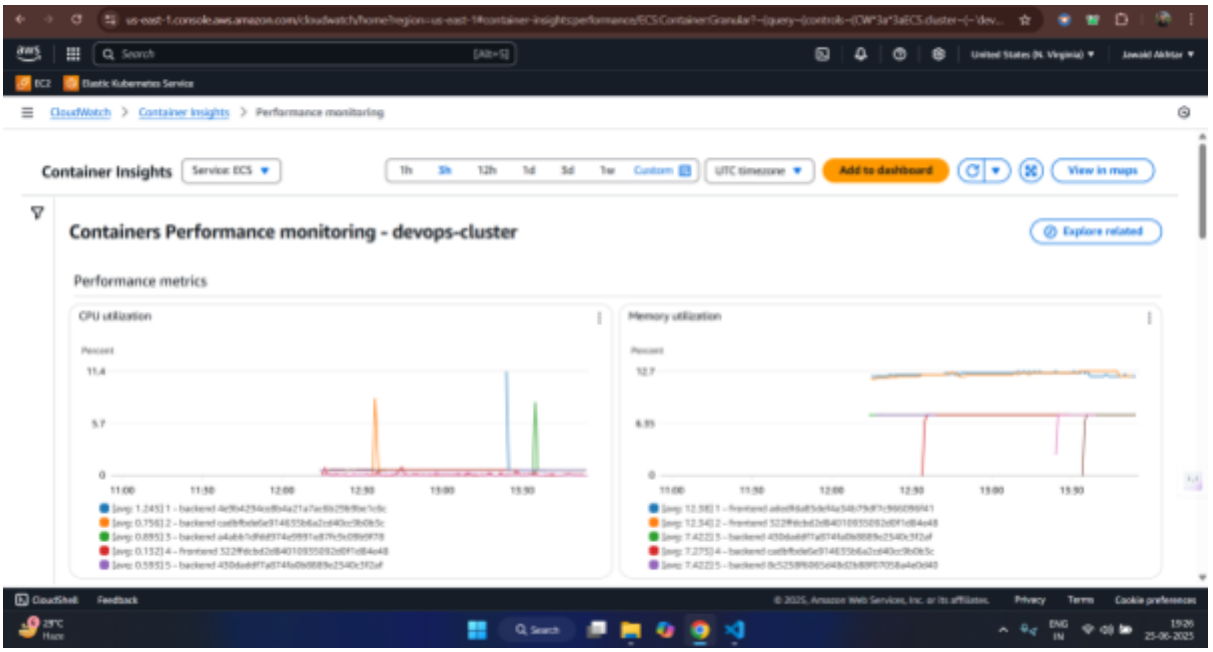
📎 Submitted by

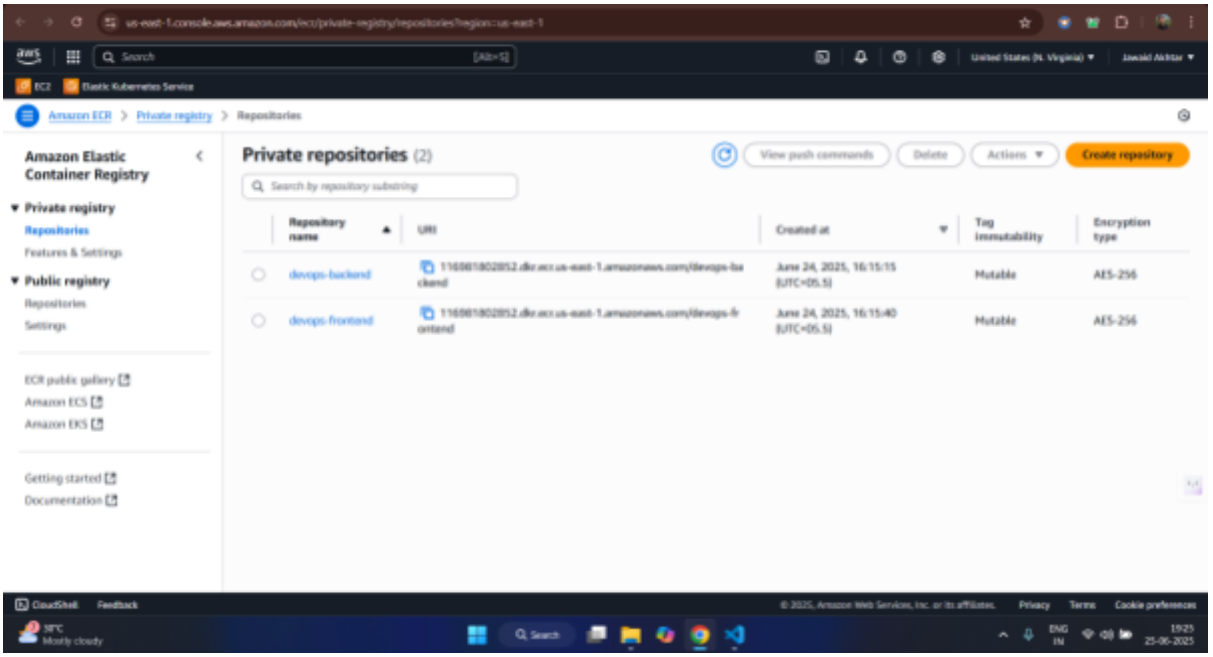
Jawaid Akhtar
DevOps Engineer

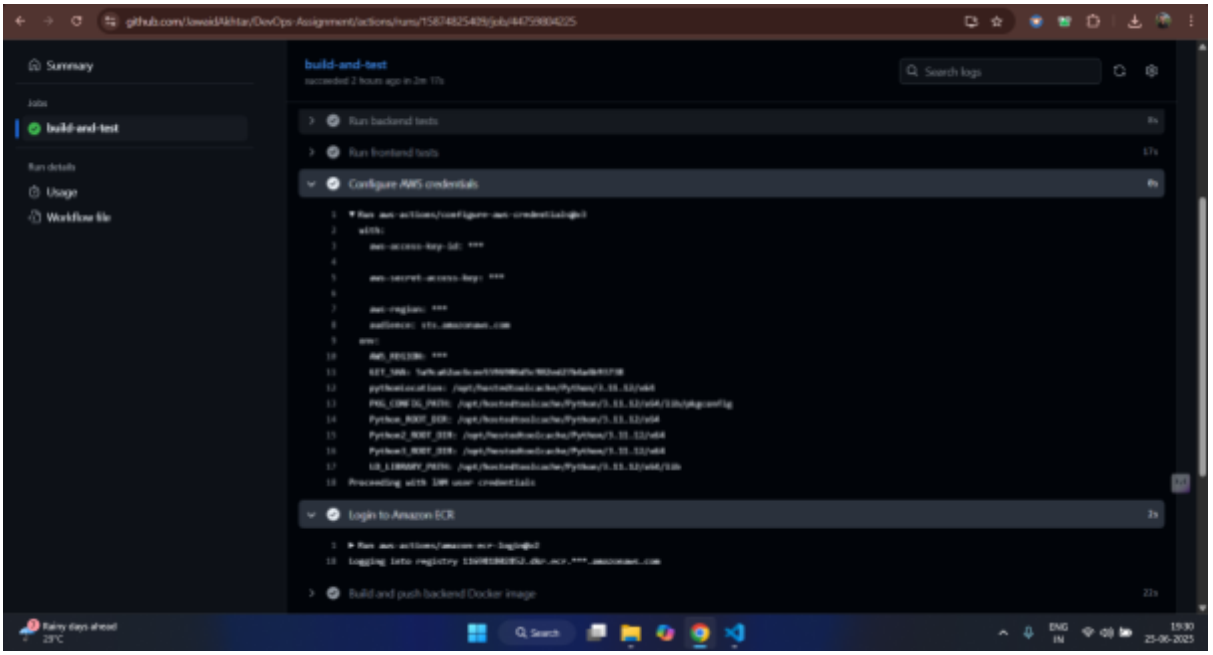





File Edit Selection View Go Run Terminal Help
DevOps-Assignment
EXPLORER
OPEN EDITORS
terraform.tfstate
DEVOPS-...
github
backend
frontend
terraform
modules
.terraform.lock.hcl
main.tf
outputs.tf
terraform.tfstate
terraform.tfvars
variables.tf
.gitignore
README.md
OUTLINE
TIMELINE
main
Launchpad
Ln 1, Col 1
Spaces: 2
UTF-8
LF
{} JSON
38°C
Mostly cloudy
Search
ENG
IN
19:25
25-06-2025


us-east-1.console.aws.amazon.com/cloudwatch/home?region=us-east-1#container-insights:performance/ECS:ContainerGranular?~(query~(controls~(CW*3a*3aECS.cluster~(~'dev...
Search
[Alt+S]
United States (N. Virginia)
Jawad Akhtar
EC2
Elastic Kubernetes Service
CloudWatch
Container Insights
Performance monitoring
Container Insights
Service: ECS
1h
3h
12h
1d
3d
1w
Custom
UTC timezone
Add to dashboard
View in maps
Containers Performance monitoring - devops-cluster
Explore related
Performance metrics
CPU utilization
Memory utilization
Percent
11.4
5.7
0
12.7
6.35
11:00
11:30
12:00
12:30
13:00
13:30
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
28°C
Haze
Search
ENG
IN
19:26
25-06-2025


Amazon Elastic Container Registry
Private registry
Repositories
Features & Settings
Public registry
Repositories
Settings
ECR public gallery
Amazon ECS
Amazon EKS
Getting started
Documentation
Private repositories (2)
View push commands
Delete
Actions
Create repository
Search by repository substring
Repository name
URI
Created at
Tag immutability
Encryption type
devops-backend
devops-frontend
June 24, 2025, 16:15:15 (UTC+05.5)
June 24, 2025, 16:15:40 (UTC+05.5)
Mutable
AES-256
CloudShell
Feedback


Summary
build-and-test
Usage
Workflow file
Run backend tests
Run frontend tests
Configure AWS credentials
Login to Amazon ECR
Build and push backend Docker image