

AWS & DevOps Mastery Project

Project Overview

You will build a scalable, highly available, and fully automated cloud-based web application with a CI/CD pipeline. The project will include:

- Frontend: React.js (hosted on S3 with CloudFront)
- Backend: FastAPI (deployed on AWS Lambda with API Gateway)
- Database: Amazon DynamoDB (NoSQL) or RDS (PostgreSQL/MySQL)
- Storage: Amazon S3 (for static assets)
- Authentication: AWS Cognito
- Monitoring & Logging: AWS CloudWatch & AWS X-Ray
- Infrastructure as Code: Terraform
- CI/CD Pipeline: GitHub Actions + AWS CodePipeline
- Containerization: Docker + ECS (Fargate) / EKS (if free tier allows)
- Networking: VPC, Subnets, Security Groups
- IAM Roles & Policies: AWS IAM for access control

Implementation Steps

1. Setup AWS & Terraform

Create an AWS free-tier account. Install Terraform and configure AWS CLI.

2. Deploy Frontend (React)

Develop a React application, host it on S3 with CloudFront, and automate deployment using Terraform.

3. Deploy Backend (FastAPI)

Create a FastAPI service, deploy it on AWS Lambda with API Gateway, or ECS Fargate, and automate using Terraform.

4. Setup Database & Storage

Choose DynamoDB (NoSQL) or RDS (PostgreSQL/MySQL). Store files in S3. Automate using Terraform.

5. Authentication with Cognito

Implement user authentication using AWS Cognito. Restrict access based on IAM roles.

6. CI/CD Pipeline

Configure GitHub Actions, AWS CodePipeline & CodeBuild for automated deployment using Terraform.

7. Monitoring & Logging

Use CloudWatch for logs and AWS X-Ray for tracing. Automate using Terraform.

8. Networking & Security

Configure VPC, Subnets, Security Groups, IAM roles, and WAF for API protection.

Why This Project?

- Covers AWS Cloud Practitioner Topics: Compute, Storage, Networking, Security, Monitoring, and Billing.
- Master DevOps Concepts: CI/CD Pipelines, Infrastructure as Code, Containerization, and Serverless Deployment.
- AWS Free-Tier Friendly: Utilizes free-tier services such as S3, DynamoDB, Lambda, API Gateway, CloudWatch, and Cognito.
- Practical Hands-on Learning: Gain real-world experience by implementing a fully functional project.

Next Steps

1. Start setting up Terraform and AWS CLI.
2. Begin coding the React & FastAPI app.
3. Deploy the frontend and backend incrementally while automating infrastructure.
4. Let me know if you need Terraform code snippets for automation.