DevOps & Cloud Engineer - Project Portfolio

1. High Availability Healthcare Cloud Infrastructure (HIPAA-Compliant)

I led the design and deployment of a high-availability, HIPAA-compliant cloud infrastructure for a healthcare provider. The goal was to securely host patient data and applications while ensuring compliance and minimizing downtime.

Architecture:

- Multi-AZ VPC with public/private subnets
- EC2 in Auto Scaling Groups behind Application Load Balancers
- RDS PostgreSQL in Multi-AZ mode with cross-region backups
- S3 with lifecycle policies and encryption
- AWS WAF and Shield Advanced for threat protection

Tools:

- Terraform for IaC
- GitHub Actions for CI/CD
- AWS Secrets Manager for credential management
- CloudWatch + GuardDuty for monitoring and threat detection

Outcome:

- Achieved 99.99% uptime
- Passed third-party HIPAA compliance audit
- Reduced recovery time (RTO) to under 5 minutes using DR failover

2. E-Commerce Microservices Platform on Kubernetes (EKS)

I migrated a monolithic e-commerce platform to microservices architecture using EKS to improve scalability and developer velocity.

Architecture:

- Microservices deployed in isolated namespaces on Amazon EKS
- Ingress controlled by NGINX Ingress Controller
- Backend services connected to Amazon RDS, frontend served via S3 + CloudFront

- ElastiCache (Redis) for cart/session management
- CI/CD with GitLab CI and Helm charts

Key Features:

- Canary deployments with ArgoCD
- Horizontal pod autoscaling based on CPU and latency
- Secrets managed using Kubernetes Secrets with IRSA

Outcome:

- Reduced deployment time from 1 hour to 10 minutes
- Scaled to handle 10x traffic during holiday season with zero downtime
- Onboarded 5 development teams with self-service GitOps workflows

3. Centralized Logging and Monitoring Platform

I implemented a logging and observability stack across 60+ microservices in production to enhance SRE visibility and reduce MTTR.

Architecture:

- Fluent Bit agents on EC2 and EKS nodes
- Logs sent to Elasticsearch via Logstash
- Metrics collected via Prometheus, visualized in Grafana
- Alerting configured with Alertmanager and PagerDuty

Tools:

- IAM roles for service access control
- KMS for encrypting logs at rest
- Integration with AWS CloudTrail and GuardDuty

Outcome:

- Reduced incident detection time by 70%
- Enabled proactive monitoring with dynamic SLO dashboards

- Decreased mean time to recovery from 45 minutes to 10 minutes

4. Serverless Data Processing Pipeline

I designed a fully serverless data pipeline for a marketing analytics platform to ingest, transform, and analyze clickstream data.

Pipeline:

- Amazon Kinesis streams for ingestion
- Lambda functions for real-time data processing
- Data stored in Amazon S3 and transformed using Athena
- Reports generated via QuickSight dashboards

Automation:

- SAM framework for infrastructure deployment
- CloudWatch alarms to monitor Lambda duration and errors

Outcome:

- Cut infrastructure cost by 60% compared to EC2-based system
- Processed 10 million events/day with sub-second latency
- Enabled real-time campaign insights for marketing team

5. Multi-Region Disaster Recovery Setup for Fintech App

I designed a multi-region DR architecture to meet a strict RTO of 15 minutes and RPO of 5 minutes for a payment gateway system.

Infrastructure:

- Primary and standby stacks in us-east-1 and us-west-2
- Active-active S3 replication, Aurora Global Databases
- EC2/ALB in both regions with health-based routing using Route 53
- Terraform modules abstracted for dual-region deployment

Testing:

- Regular failover simulations using AWS Elastic Disaster Recovery
- Cross-region snapshots and config sync via Lambda

Outcome:

- Achieved near-zero data loss during DR drill
- Met compliance for business continuity plan
- Infrastructure recovery fully automated via Terraform

6. DevOps Automation Framework for a SaaS Product

I built a DevOps automation framework to standardize CI/CD, testing, security, and IaC across 10+ microservices maintained by different teams.

Tools Used:

- Jenkins shared libraries for pipeline templating
- Terraform modules for provisioning AWS infrastructure
- SonarQube, Trivy, and Checkov for scanning
- Slack + GitHub integration for deployment feedback loops

Innovations:

- Dynamic staging environments spun up per pull request
- Auditable deployments with version tagging and changelogs
- Role-based access controls via IAM and GitHub teams

Outcome:

- 5x faster onboarding for new services
- Eliminated manual deployments entirely
- Security issues surfaced during build, not post-deploy

DevOps & Cloud Engineer - Project Portfolio

7. Cost Optimization and Forecasting Platform

I implemented automated cost tracking and forecasting for a multi-account AWS organization to bring cloud spend under control.

Solutions:

- Deployed Athena queries on the Cost and Usage Report (CUR)
- Built QuickSight dashboards with cost breakdowns by tag/team
- Enforced budgets and alerts for high-spend services (EC2, RDS)
- Automated detection of idle resources via Lambda + EventBridge

Outcome:

- Identified $20K/month in idle resources
- Cost visibility improved decision-making for engineering leads
- Cost forecasting enabled better quarterly planning