Omobayonle Ogundele

Devops Engineer

ogundelebayo8@gmail.com
Lagos, Nigeria
linkedin.com/in/bayo123
github.com/BayoJohn

PROFESSIONAL SUMMARY

Junior DevOps Engineer with practical experience in designing and deploying scalable applications using cloud services and containerization tools. Proficient in Git-based workflows, CI/CD pipelines, and infrastructure provisioning. Demonstrated ability to troubleshoot and optimize deployment processes through hands-on project work. Eager to contribute to reliable and automated systems in a collaborative environment.

SKILLS

AWS (EC2, S3, IAM, VPC, CloudWatch, EKS, Lambda, RDS) · Oracle Cloud (Compute, VCN, Object Storage) · Docker · Docker Compose · Kubernetes · Helm · Harbor · Drone CI · GitHub Actions · Jenkins · ArgoCD · Gitea · WireGuard · Terraform · Ansible · Prometheus · Grafana · Node Exporter · cAdvisor · Python · Bash · Flask · SQL · Git · GitHub · Linux (Ubuntu) · Nginx · SQLite · PostgreSQL

PROJECTS

StackedByBayo — Self-Hosted Portfolio, Blog & CI/CD Infrastructure

- Deployed **Flask** portfolio site with custom admin dashboard, markdown blog editor, and **SQLAlchemy** ORM on k3s (3-replica autoscaling)
- Built end-to-end GitOps pipeline (**Gitea** → **Drone** CI → **Harbor** → **ArgoCD** → **k3s**) with declarative infrastructure-as-code; all deployments tracked in Git with automatic cluster sync via ArgoCDConfigured a **WireGuard VPN tunnel** between local homelab and Oracle Cloud server enabling secure private SSH access without exposing port 22 to the public internet
- Deployed kube-prometheus-stack (**Prometheus**, **Grafana**, Alertmanager) with Slack alerts for real-time visibility into cluster health and application metrics
- Implemented image scanning pipeline with Harbor + Trivy to block vulnerable container images before production deployment
- Debugged and resolved a critical **DHCP** IP change incident where the Harbor registry became unreachable — wrote a **shell automation script** that detects the current IP, reconfigures Harbor, restarts containers and updates the pipeline config automatically
- Configured **Kubernetes** PVCs and **Docker** volumes ensuring zero data loss across container restarts and blue-green deployments
- Automated infrastructure provisioning using **Ansible** playbooks for Docker and k3s deployment on Oracle Cloud instance, ensuring idempotent and reproducible cluster setup

NOVA Store — AWS Serverless E-Commerce Platform

- Architected fully serverless e-commerce application on AWS using **Lambda (Node.js), DynamoDB, API Gateway, S3 + CloudFront, SNS, SQS, SES, CloudWatch** — maximizing AWS service breadth within free-tier constraints
- Implemented Infrastructure as Code using **Terraform** with S3-backed state management and DynamoDB locking for repeatable, consistent deployments to us-east-1
- Designed dark-themed two-page frontend with **JavaScript** optimizing for global CDN distribution and minimal first contentful paint
- Integrated **CodePipeline and CodeBuild** for automated CI/CD deployment, enabling push-to-production workflows
- Configured **CloudWatch** for comprehensive application and infrastructure monitoring, with log insights for debugging and performance analysis
- Optimized for cost efficiency within **AWS free tier** limits while maximizing service breadth across Lambda, DynamoDB, API Gateway, S3, CloudFront, SNS, SQS, SES, and CloudWatch

Chaos Engineering & Resilience — Homelab Kubernetes

- Deployed Chaos Mesh on **k3s cluster** to design and execute fault injection experiments validating resilience patterns and auto-recovery mechanisms under adverse conditions
- Ran pod termination chaos experiments measuring Flask app recovery time (target <30s), validating horizontal pod autoscaling and service mesh retry logic
- Planned chaos experiments: service latency injection, pod termination, network partition simulation, and resource exhaustion to identify failure points and recovery strategies
- Implemented network fault injection (latency injection, packet loss, network partition simulation) combined with Prometheus metrics and Loki logs to identify failure modes and validate graceful degradation
- Integration with **Loki logs**, Prometheus metrics, and **Grafana** dashboards for comprehensive incident analysis and post-mortem documentation
- Validated observability gaps: identified missing alerts (KubeProxy false positives), adjusted thresholds, and improved Alertmanager routing to reduce alert fatigue

High-Availability Cloud Infrastructure & Monitoring

- Architected production-grade infrastructure on AWS EC2 using **Docker Compose** with 3+ application replicas, **Nginx** load balancer, and automatic failover eliminating single points of failure
- Built fully automated CI/CD pipeline with **GitHub Actions** (Code Push → Build → Test → Docker Hub → SSH Deploy → Health Verify) achieving zero-downtime rolling deployments
- Implemented self-healing mechanism using Docker health checks and autoheal container for automatic recovery from failures; configured **Prometheus** health check intervals and thresholds
- Deployed comprehensive monitoring stack (**Prometheus, Grafana, Node Exporter**) collecting system, container, and application metrics with real-time dashboards and custom alerts
- Configured **SSL/TLS automation** with zero-downtime certificate renewal and secure HTTPS traffic encryption for

CERTIFICATIONS & EDUCATION

Introduction to Linux — the Linux Foundation

AWS Certified Cloud Practitioner — Amazon

B.Sc. Geography and Planning — University of Lagos (UNILAG)