Muhammad Ibtisam Iqbal

DevOps & Cloud Engineer

contact@ibtisam-iq.com | Islamabad, Pakistan | linkedin.com/in/ibtisam-iq | github.com/ibtisam-iq | ibtisam-iq.com

General CV. A version written for a specific role is available on request.

PROFESSIONAL SUMMARY

CKA and CKAD certified DevOps & Cloud Engineer focused on Kubernetes, AWS, and GitOps delivery, with DevSecOps CI pipelines on Jenkins and GitHub Actions. Audited and modernized application codebases across Java, Python, and Node.js (dependency upgrades, config externalization, health endpoints, architectural refactoring) before containerizing them and deploying across EC2, ECS, kubeadm, and EKS. Built and documented 8 infrastructure projects end-to-end, from Terraform-provisioned microservices on EKS to a self-hosted CI/CD platform on microVMs, each with source code and a runbook. Moved into DevOps in 2024 after two years of client-facing e-commerce operations.

CERTIFICATIONS

Certified Kubernetes Administrator (CKA)
Cloud Native Computing Foundation (CNCF) · Nov 2025 · Valid until Nov 2027

Certified Kubernetes Application Developer (CKAD)
Cloud Native Computing Foundation (CNCF) · Nov 2025 · Valid until Nov 2027

Certified Kubernetes Security Specialist (CKS)
In Progress · Cloud Native Computing Foundation (CNCF)

TECHNICAL SKILLS

Cloud (AWS): EKS, ECS, EC2, S3, IAM, KMS, ACM, SNS, SQS, VPC, Auto Scaling, VPC Peering, NAT Gateway, Route 53, RDS, DynamoDB, CloudFront, CloudFormation, Lambda, CloudTrail

Containers & Orchestration: Kubernetes, kubeadm, Docker, Helm, Helmfile, Docker Buildx, Kustomize, ArgoCD, Docker Compose, ArgoCD Image Updater

CI/CD & DevSecOps: Git, Jenkins, Nexus, Trivy, Hadolint, Bandit, Maven, Make, GitHub Actions, pip-audit, SonarQube

Infrastructure as Code: Terraform, Ansible, Bash, eksctl, AWS CLI

Linux & Databases: Ubuntu, Alpine Linux, systemd, PostgreSQL, MySQL, Redis, RabbitMQ

Practices: GitOps, DevSecOps, Cloud Native, Application Modernization, Microservices Architecture, Load Balancing, Infrastructure Automation, Security & Compliance, Virtualization, OS Hardening, Blue/Green & Canary Deployments, Disaster Recovery, Troubleshooting, 12-Factor App

Networking: Gateway API, Ingress, Nginx, DNS, TLS/SSL, OpenSSL, AWS Load Balancer Controller, ExternalDNS, Cloudflare Tunnel

Observability: Prometheus, Grafana, AlertManager, Kibana, Filebeat, CloudWatch, Elasticsearch, Fluent Bit

Languages & Frameworks: Java, Spring Boot, Python, Flask, Gunicorn, Node.js, Express, React

EXPERIENCE

DevOps Engineer · *Independent Infrastructure Lab* — Apr 2024 – Present

- Delivered 8 projects across Java, Python, and Node.js on EC2, ECS, kubeadm, and EKS, each with source code and a runbook.
- Audited and modernized 3 codebases (upgrades, externalized config, health endpoints); containerized with multi-stage builds.
- Built 14 to 21 stage DevSecOps CI pipelines on Jenkins and GitHub Actions with Trivy vulnerability gates and SonarQube.
- Implemented GitOps with ArgoCD and Image Updater on EKS, provisioned with Terraform and routed through Gateway API.
- Built and ran projects on iximiuz Labs microVMs, KodeKloud labs, and AWS; published 2 tutorials and 5 playground images.
- Earned CKA and CKAD (2-hour exams in live clusters) and published certification prep at cert-vault.ibtisam-iq.com.

E-Commerce Operations Specialist · *Independent Clients (Amazon FBA) · Remote Contract* — Apr 2022 – Apr 2024

- Managed Amazon FBA private label and wholesale for 5+ international clients: research, sourcing, listings, PPC, and logistics.
- Owned each account from inception to live operation; one reached CAD $21K+ in revenue across 888 units sold in its first year.
- Self-funded 100% of the transition into DevOps engineering, including lab platforms and CNCF certification exams, from this income.

PROJECTS

Microservices GitOps on Amazon EKS — *Terraform, EKS, ArgoCD, Gateway API, HTTPRoutes, ExternalDNS, HPA, ACM, EBS CSI, Prometheus, Elastic Stack*

- Deployed 10 microservices on Amazon EKS using modular Terraform (VPC, bastion, self-managed ASGs) and 3 GitHub Actions CI pipelines with dynamic matrix dispatch; built monorepo CI with per-service change detection so only affected services rebuild and one failure never blocks the rest.
- Automated GitOps delivery: CI pushes images to GHCR, ArgoCD Image Updater detects new SHA-256 digests and triggers EKS rollouts with zero manual intervention, using Kustomize overlays to inject custom Gateway API manifests while keeping the upstream Helm chart unmodified.
- Configured Gateway API routing with a shared ALB across 5 HTTPRoutes, ExternalDNS for automatic Route 53 record creation, and HPA for CPU-based autoscaling via Metrics Server.
- Provisioned an ACM wildcard certificate for TLS and configured the EBS CSI driver with gp3 StorageClass for Elasticsearch PVCs; ran observability independently of the app via kube-prometheus-stack (Slack alerts through AlertManager) and Elastic Stack across a 4-node cluster.

SilverStack: Self-Hosted CI/CD Platform — *systemd, Docker, Jenkins, SonarQube, Nexus, Cloudflare Tunnel, Nginx, GHCR*

- Built a self-hosted CI/CD platform (Jenkins, SonarQube, Nexus) on microVMs with 5 custom OCI rootfs images, each using systemd as PID 1 for runtime provisioning at boot and independently launchable as a public playground on iximiuz Labs.
- Configured Nginx as a reverse proxy with a build-time parameterized upstream port, /health readiness endpoint, and hardened security headers (X-Frame-Options, X-Content-Type-Options, Referrer-Policy); automated image CI on GitHub Actions to parse configs and verify systemd wiring before every push to GHCR.
- Configured outbound-only Cloudflare Tunnels to expose services on custom domains without public IPs, bypassing Carrier-Grade NAT (CGNAT); enforced security via strict sudoers profiles, daemon user isolation, pinned binary versions, and SHA256 checksum verification on all downloads.

DevSecOps CI Pipelines *Jenkins, GitHub Actions, Trivy, SonarQube, GHCR, Docker Hub, Nexus, JaCoCo, Cobertura*

- Validated each application's build and test cycle locally (Maven, Webpack, pytest against real databases) before automating; built one CI pipeline contract for 3 codebases at 14, 16, and 21 stages (Java, Python, Node.js) and implemented it twice, on Jenkins and GitHub Actions, to prove tool-agnostic design.
- Integrated language-specific scanning (Bandit and pip-audit for Python, npm audit and ESLint for Node.js, Trivy filesystem scan for Java) with 3-pass Trivy gating: library CVEs hard-fail on CRITICAL, OS CVEs run advisory-only to avoid false blocks from noise.
- Tagged every image with semantic version, 7-character Git SHA, and CI build number (no floating 'latest' tags), publishing to GHCR, Docker Hub, and Nexus simultaneously; ran CI tests against in-memory databases (SQLite for Python) while still generating JaCoCo and Cobertura coverage reports for SonarQube quality gates.
- Enforced strict CI/CD separation: pipelines commit image SHAs to the CD repository. No pipeline has kubectl access to any cluster.

Multi-Environment Deployment: Bare-Metal to Amazon EKS *Helmfile, kubeadm, EKS, IRSA, DynamoDB, SQS, Lambda, SNS, ElastiCache, CloudFormation*

- Deployed 5 microservices across bare-metal Kubernetes (kubeadm) and Amazon EKS, keeping upstream Helm charts unmodified and swapping DynamoDB, SQS, and ElastiCache for containerized RabbitMQ and Redis on bare metal; wrote 3 Helmfile configurations with dependency sequencing so databases and brokers provision before application services.
- Created self-managed CloudFormation worker nodes to bypass KodeKloud lab SCP restrictions on EKS managed node groups. Configured IRSA to bind services to DynamoDB and a serverless event pipeline (SQS, Lambda, SNS).
- Deployed Fluent Bit to stream logs to CloudWatch Container Insights; set up an ALB Ingress Group to share a single load balancer across the UI, Prometheus, and Grafana.

Polyglot Application Modernization Across 4 Compute Models *Java, Spring Boot, Python, Flask, Gunicorn, Node.js, Express, React, Docker, Kustomize, kubeadm, EKS, ECS, EC2*

- Audited and modernized 3 existing codebases (Java Spring Boot 3.4, Python Flask, Node.js Express): upgraded frameworks, pinned dependencies, externalized all config to environment variables, and injected container health endpoints.
- Refactored the Node.js app from 2-tier to 3-tier architecture (Nginx serving React, proxying API to Express). Built multi-stage Dockerfiles for each stack; validated every build on bare metal before containerizing.
- Debugged environment-specific edge cases across all 3 stacks (python-dotenv URL truncation, PostgreSQL 15+ schema-level grants, JDBC URL quoting in shell scripts); kept the Flask app as a single-container Gunicorn deployment instead of forcing a microservice split, matching the architecture to the application, not the trend.
- Deployed all 3 applications across EC2, ECS Fargate, bare-metal Kubernetes (kubeadm), and Amazon EKS. Wrote Kustomize overlays sharing 80% of base manifests between environments.

BankApp: Same Artifact Across EC2, ECS, and EKS *EC2, ECS Fargate, EKS, RDS, Gateway API, Kustomize, Docker*

- Deployed the same Java artifact (Spring Boot 3.4, Java 21) across EC2 Auto Scaling, ECS Fargate, and Amazon EKS to compare each compute model firsthand. All three share one VPC and RDS instance.
- Decoupled the embedded database to Amazon RDS, reduced container image by 60% with multi-stage builds, and aligned JVM memory parameters with Kubernetes resource limits to prevent OOM kills.

DebugBox: Open-Source Kubernetes Diagnostics Toolkit *Alpine Linux, Docker Buildx, Trivy, Hadolint, GitHub Actions, MkDocs, kubectx, kubens*

- Built an open-source Kubernetes debugging toolkit with 3 Alpine-based variants (15MB to 91MB); the smallest is 93% smaller than netshoot. Separated Dockerfiles (declarative) from installation scripts (procedural), with package lists in '.packages' files and complex installs using dedicated scripts with architecture detection and SHA256 verification.
- Pre-loaded kubectx/kubens and custom bash functions (json(), yaml(), sniff-dns(), cert-check()) for faster debugging sessions; automated multi-arch builds (amd64, arm64) with Trivy gating on HIGH/CRITICAL findings, Hadolint linting, in-container smoke tests, and dual-registry publishing (GHCR + Docker Hub).
- Published documentation via MkDocs and GitHub Pages. Wrote CONTRIBUTING.md, CODE_OF_CONDUCT.md, and SECURITY.md for open-source governance.

AWS Static Hosting with CloudFront OAC and Cross-Region Replication *S3, CloudFront, KMS, Route 53, CloudTrail, IAM, Glacier, OAC*

- Built a zero-compute hosting architecture on AWS using CloudFront with Origin Access Control (OAC/SigV4) against a fully private S3 origin. Configured Custom Error Responses to handle SPA client-side routing.
- Configured Cross-Region Replication (us-east-1 to us-west-2) with separate KMS Customer Managed Keys per region and S3 Bucket Keys to reduce KMS API costs; set up CloudTrail auditing across both regions and S3 Server Access Logs for HTTP telemetry, with Glacier lifecycle policies for versioned content.
- Deployed on a personal AWS account and kept live for 7 days. Total cloud cost: $1.45 across CloudFront, S3 (3 buckets), CloudTrail, and KMS, covered by AWS Free Tier credits.

EDUCATION & TRAINING

AWS Solutions Architect Training (100 hrs) · *Technical Guftgu* — Dec 2025

Linux Administration (60 hrs) · *Technical Guftgu* — Nov 2025

DevSecOps & Cloud DevOps BootCamp · *DevOps Shack* — Jan 2025

Hands-on Lab Platforms · *KodeKloud (CKA/CKAD Prep, AWS Labs) + iximiuz Labs (2 Published Tutorials, 5 Playground Images)* — 2024 – Present

M.Sc. Horticulture · *University of Agriculture, Faisalabad* (Silver Medal, Mar 2024) — 2022

PORTFOLIO & DOCUMENTATION

Projects: projects.ibtisam-iq.com

Cert Vault: cert-vault.ibtisam-iq.com

Blog: blog.ibtisam-iq.com

SilverStack: github.com/ibtisam-iq/silver-stack

Runbook: runbook.ibtisam-iq.com

Knowledge Base: nectar.ibtisam-iq.com

DebugBox: debugbox.ibtisam-iq.com

iximiuz Labs: labs.iximiuz.com/a/ibtisam-iq