

Abad Naseer

AI Infrastructure & Platform Engineer | GPU · Kubernetes · vLLM Inference · Cloud · IaC · Observability
+92 340-5059429 | abad.naseerfast@gmail.com | linkedin.com/in/abadnaseer | github.com/AbadNaseer | abad.falcoflow.com

SUMMARY

Infrastructure engineer who owns the path from GPU cloud to served token: GPU worker provisioning, Kubernetes scheduling, vLLM serving, gateway routing, and the automation that makes any of it replaceable. Five years across cloud platforms, arriving at inference from the cost side: roughly 30% off one AWS bill at 1M+ users, and ~60% off infrastructure spend on a migration off AWS with zero customer-facing downtime. I also build what runs on top, which is uncommon in infrastructure engineers.

EXPERIENCE

Platform Engineer, AI Infrastructure & Inference

Jun. 2026 – Present

Ccript Agency

Islamabad, Pakistan

- **LLM Inference Platform (GPU + Kubernetes):** Own the full inference chain: GPU provider → GPU VM → Kubernetes worker → NVIDIA container runtime → GPU scheduling → vLLM → KV-cache → gateway, turning raw GPU capacity into a reliable, observable inference service that product teams consume as an endpoint
- **Inference Serving & Traffic Layer:** Deployed and operate vLLM as the serving runtime with LMCache for distributed KV-cache reuse and an AIBrix gateway fronting Kubernetes Services for per-model routing; validated readiness by actual inference rather than by pods reporting **Running**
- **GPU Worker Lifecycle Automation:** Automated the complete worker lifecycle with Ansible: provisioning, K3s cluster join, NVIDIA runtime and GPU verification, firewall synchronisation, dynamic address handling, teardown and rebuild. A GPU node is a replaceable unit, never a hand-maintained server
- **Failure Engineering:** Repeatedly destroy and recreate GPU workers to prove the platform recovers unattended; surfaced and fixed stale firewall state, node-local storage assumptions, a GPU device-plugin race, dynamic worker addressing, model/image recovery time, and verification logic competing with the workload for the GPU
- **Kubernetes Platform Engineering for AI Workloads:** K3s control plane and GPU worker pool: node lifecycle, scheduling, resource requests, Services, Deployments, ingress, storage behaviour, and Kubernetes failure and recovery scenarios specific to GPU-backed workloads
- **Inference Observability & Benchmarking (in progress):** Building Prometheus and Grafana across GPU, Kubernetes, vLLM and gateway metrics, plus a benchmark harness measuring time-to-first-token, inter-token latency, throughput, GPU utilisation, GPU memory and KV-cache capacity, so model, workload and configuration are matched on measurement rather than on published numbers
- **Cloud Data Platform (parallel workstream):** Reverse-engineered and migrated an Explo-powered dashboard serving 150+ outlets across the United States onto a native presentation layer, preserving platform architecture while removing third-party cost; repaired ETL/ELT pipelines, KPI accuracy, filtering, location mappings and BigQuery reconciliation, then built data-quality and monitoring workflows that ended recurring client-facing reporting errors
- **Production Support:** Investigate incidents across GCP, BigQuery, pipelines and deployments; perform impact analysis before deployment, validate fixes in non-production, and maintain monitoring documentation, operational runbooks and SLA resources

DevOps Engineer

Oct. 2025 – Apr. 2026

Mediatiz Foundation

Islamabad, Pakistan

- Managed AWS infrastructure (VPC, EC2, ECS, RDS, ALB, CloudFront, CloudWatch, SES, Route 53) for Mediatiz LMS and Khudi App serving 1M+ users with 20–30K concurrent requests at peak load; sustained 99.9% uptime
- Reduced AWS cloud costs by 30% through CloudWatch-driven auto-scaling, ECS task-definition tuning, and workload-aware right-sizing across staging and production clusters
- **Self-hosted model serving on Linux:** Architected and deployed PyBot, a Django + Ollama (Gemma 2B) LLM tutor running on self-managed Ubuntu infrastructure instead of a vendor API, removing an external service from the platform's failure domain and its cost line while keeping student PII inside our own boundary. Owned it end to end in production: provisioning, runtime, and availability. First system where I owned the model serving layer rather than calling someone else's
- Built GitHub Actions CI/CD pipelines for Android (APK/AAB), QA automated testing, Laravel LMS, Django, and Next.js admin dashboard with zero-downtime ECS rollouts, Slack deployment alerts, and Google Play Console automation; eliminated manual release-engineering overhead

- Built serverless Slack-to-Google Sheets automation on AWS Lambda, API Gateway, EventBridge, and S3 for daily cross-departmental reporting across 8 departments; replaced a manual workflow
- Ran in-house security testing on the LMS staging environment (brute-force, SQL injection, CSRF) using Kali Linux, SQLMap, and SonarQube; validated WAF rules and produced remediation reports adopted by the development team
- Authored the internal LMS DevOps Automation Roadmap and built CloudWatch dashboards covering ECS cluster health, RDS performance, and cost tracking

DevOps Engineer

Dec. 2023 – Oct. 2025

Poshmaal Technologies

Islamabad, Pakistan

- **Firefly.online, AWS to On-Premises Migration:** Led end-to-end migration of an IoT SaaS platform (Node.js, React, PostgreSQL) from AWS to self-managed bare-metal infrastructure using a Blue-Green deployment strategy; replaced EC2, ECS, RDS, ALB, S3, CloudFront, Lambda, and Redis with fully self-hosted equivalents
- Executed full data migration of PostgreSQL databases, Redis persistence, and object storage with zero customer-facing downtime; reduced infrastructure spend by ~60% while improving operational control and deployment flexibility
- **Platform Rebuild & Production Infrastructure:** Designed and provisioned Development, Staging, and Production environments during the migration from Node.js to Java Spring Boot; built and secured Ubuntu infrastructure with Java, Maven, Docker, PostgreSQL, Redis, Apache Kafka, Node.js, and NGINX
- Built GitHub Actions + SonarQube CI/CD pipelines automating code-quality checks, Docker image builds, and environment-specific deployments; configured Nginx upstream load balancing for multiple backend instances
- Deployed full-stack observability with Grafana, Prometheus, Loki, and Node Exporter for real-time monitoring, centralized logging, and alerting; hardened all environments with SSL/TLS (Let's Encrypt), UFW, Fail2ban, and SSH key restrictions
- **ERP Platform, Kubernetes GitOps Infrastructure (TransIP):** Architected and deployed a production-ready K3s cluster on TransIP; provisioned servers, networking, and cluster resources for ERP workloads spanning C2B, B2C, and C2C business domains
- Deployed and maintained 10+ microservices on K3s with environment isolation, ingress routing, scaling policies, and service-to-service networking; implemented Argo CD GitOps for automated, version-controlled deployments and consistent release management
- **GCP Kubernetes Deployment:** Deployed and configured Kubernetes workloads on Google Cloud Platform, supporting cluster setup, application deployment, ingress configuration, service exposure, and environment validation for containerized applications

DevOps Consultant (Freelance)

Dec. 2021 – Nov. 2023

Fiverr & Upwork

Remote

- Delivered 20+ containerization, CI/CD, and server infrastructure projects for small-business clients; Docker-based deployments, GitHub Actions pipelines, Nginx configuration, and SSL automation with Certbot, using Ansible for repeatable provisioning
- Deployed self-hosted Kubernetes for several clients with Ingress, HPA, and RBAC; wrote Bash tooling for routine server maintenance, patching, and backup scheduling

CERTIFICATIONS & ACHIEVEMENTS

- **AWS Certified Solutions Architect** – In Progress
- **Certified Kubernetes Administrator (CKA)** – In Progress
- **RocketDevs (Vetted Talent):** Selected after technical assessments and interviews

PROJECTS

Conversational Commerce Platform | *FastAPI, pgvector, Gemini, Next.js, AWS* | smartzees.com 2026 – Present

- Three production AI assistants on one shared retrieval architecture (grocery ordering, home services, community documents), with the backend owning all state, search and money so the model only classifies intent and phrases the reply
- Replaced per-request database scans with an in-memory vector index: catalog search fell from 6,456 ms to 8.2 ms and database reads per message from 25,701 to 2, with results bit-identical to the old path across every query tested
- Moved voice off the browser speech API onto server-side speech-to-text and synthesis, taking the product out of Chrome-only; added an LRU cache on repeated phrases that cut a repeated voice turn from 10.0 s to 6.9 s

LogSense | *Python, FastAPI, MLflow, Airflow, DVC, HuggingFace, Slack, PagerDuty*

2024

- Built NLP models for anomaly detection, issue classification, and automated root-cause analysis on infrastructure logs; reduced MTTR by ~45%
- Designed an end-to-end MLOps pipeline: MLflow (experiment tracking + model registry), DVC (data/model versioning), Airflow (orchestration); containerized FastAPI inference microservice
- Wired webhook-driven alert routing to Slack, Microsoft Teams, and PagerDuty with severity-based escalation; grouped correlated anomalies before notifying on-call to cut alert noise
- Minimized reliance on external LLM APIs by serving custom NLP models in-house for privacy and cost control

Kubernetes GitOps Platform | *K8s, ArgoCD, Helm, GitHub Actions, Prometheus, Traefik* 2025

- Deployed a 10-service polyglot microservices app (Go, Java, Node.js, Python) on a self-hosted bare-metal Kubernetes cluster via Helm; managed all cluster state in Git with ArgoCD auto-sync and self-healing on drift
- Built GitHub Actions CI/CD for all 10 services. Each pipeline builds and tags Docker images with the commit SHA, pushes to Docker Hub, and commits per-service tag overrides back to the GitOps repo for ArgoCD to roll out independently
- Fixed broken upstream Dockerfiles: replaced the deleted openjdk:8-jdk image with a maven:3.8-eclipse-temurin-8 multi-stage build on amazoncorretto:8, and upgraded the Go dispatch service to Go 1.22+ to resolve a pprof dependency break
- Deployed kube-prometheus-stack via ArgoCD (multi-source Application, tuned for an 8 GB VPS); wrote ServiceMonitors and instrumented cart/payment services with prom-client verified end-to-end in Grafana via PromQL

NextLab Platform Deployment | *AWS, ECS, RDS, ALB, CloudFront, Route 53, GitHub Actions* 2025

- Designed and deployed end-to-end AWS infrastructure for a multi-tenant laboratory SaaS, covering networking, compute, databases, security, CI/CD, monitoring, and production operations

EDUCATION

National University of Computer and Emerging Sciences (FAST-NUCES)	Islamabad, Pakistan
<i>Bachelor of Science in Software Engineering</i>	<i>2020 – 2025</i>

TECHNICAL SKILLS

AI Infrastructure & Inference: vLLM, LMCache (distributed KV-cache), AIBrix gateway, GPU scheduling (NVIDIA container runtime, device plugin), GPU node lifecycle, model serving & autoscaling, inference benchmarking (TTFT, inter-token latency, throughput, GPU utilisation, KV-cache capacity), Ollama, HuggingFace, PyTorch

AI Applications & MLOps: RAG (pgvector, FAISS, MiniLM), LangChain, MLflow, Airflow, DVC, FastAPI inference services, speech-to-text / text-to-speech pipelines

Cloud & Infrastructure: GCP (GKE, Compute Engine, BigQuery, Cloud Storage, Cloud SQL, IAM, Cloud Logging & Monitoring, Pub/Sub, Cloud Functions), AWS (EC2, ECS/ECR, RDS, ALB, S3, CloudFront, WAF, Lambda, API Gateway, EventBridge, CloudWatch, SES, Route 53, IAM, VPC), bare metal, VPS, Linux (Ubuntu/Debian)

Kubernetes, CI/CD & IaC: Kubernetes (K3s, EKS, GKE), Helm, ArgoCD, GitOps, Docker (multi-stage), Terraform, Ansible, GitHub Actions, Jenkins, Nginx, Traefik, blue-green & zero-downtime deployments, self-hosted runners, Bash

Observability & Reliability: Prometheus (ServiceMonitor, PromQL, prom-client), Grafana, Loki, EFK Stack, Jaeger, OpenTelemetry, eBPF (groundcover), CloudWatch dashboards, incident response, root-cause analysis, failure & chaos testing

DevSecOps & Security: AWS WAF, SSL/TLS (Let's Encrypt), OS & SSH hardening, UFW/iptables, Vault (secrets), least-privilege IAM & RBAC, OWASP testing (SQLi, CSRF, brute-force), Kali Linux, SQLMap, SonarQube

Databases, Messaging & Languages: PostgreSQL, pgvector, MySQL, MariaDB Galera + ProxySQL, Redis, Apache Kafka, SQLite, Supabase | Python, Bash, Java/Spring Boot, PHP/Laravel, TypeScript, Kotlin, Next.js, React, Django, Node.js