ZAIN RAISAN

Abu Dhabi, United Arab Emirates

zain.kml.raisan@gmail.com ◆ +971 56 606 5536 ◆ zainraisan.com ◆ linkedin.com/in/zain-kml-raisan ◆ github.com/zraisan

AI infrastructure engineer running self-hosted AI platforms for Abu Dhabi government on Azure Kubernetes Service. Serves LLMs on vLLM and SGLang, packs multiple models per GPU with Triton and HAMi GPU slicing, builds TensorRT engines, and templates 130+ Kubernetes services with Helm under Grafana, Prometheus, Loki, and Tempo observability. Runs rollouts and workshops with government stakeholders. Native Arabic, fluent English, UAE based and open to travel.

WORK EXPERIENCE

Department of Municipality and Transport — **Abu Dhabi, UAE**

Lead Platform Engineer — Jun 2025 – Present

- Turn business demand from DMT and its six affiliated entities, including the three regional municipalities, Abu Dhabi Mobility, and ADREC, into infrastructure, mapping each request onto the platform architecture and provisioning any new infrastructure it requires, such as nodes, GPUs, or storage, with Pulumi
- Ran the AKS observability stack on Grafana, with Prometheus for metrics, Loki for logs, and Tempo for traces across every containerized AI service
- Led development of a department-wide demand onboarding platform as lead developer, building a conversational AI planner that elicits requirements from business stakeholders, checks incoming demand against existing contracts to avoid duplicate work, and produces budget and timeline estimates before any build starts
- Initiated and led the department's API catalogue standard, enforcing a single OpenAPI specification across application teams with an automated conformance linter as a quality gate

AI Engineer — Jan 2025 – Jun 2025

- Containerized and deployed 130+ services with Docker on Azure Kubernetes Service (AKS), generating their Deployment, Service, and Ingress manifests from Helm templates
- Served LLMs on vLLM and SGLang to support more concurrent users per GPU, using PagedAttention to cut KV cache waste and RadixAttention to reuse shared prompt prefixes
- Sliced physical GPUs across Kubernetes workloads with HAMi GPU virtualization, so several inference services share one GPU instead of each holding a full card
- Hosted multiple models on a single GPU with NVIDIA Triton Inference Server, such as Qwen3 8B alongside a Qwen reranker, and built TensorRT engines optimized for the specific GPU each model runs on
- Built a bilingual legal RAG system over 600+ Abu Dhabi legal documents with self-hosted Qwen embedding and reranker models on Triton, keeping Arabic and English in separate vector stores and delivering sub-2-second response times
- Built a multi-agent AI dashboard with 13 specialized agents on Agno and FastAPI, pulling from two government databases and serving 10,000+ internal users with visual analytics
- Delivered Microsoft 365 Copilot enablement workshops and training materials across two government organizations, one for executive directors and one for staff, covering Work and web grounding, custom agent building, and Copilot Studio pipelines
- Architected a city-wide road defect detection system using a quantized Qwen3 vision-language model with an LLM verification layer, processing dashcam and CCTV footage from 1000+ sources
- Built the AI Champions Platform, an internal engineering-productivity platform for 300+ UAE government employees, with Next.js, FastAPI, PostgreSQL, and Entra SSO, and shipped API specs and deployment runbooks so other teams could operate it

Deloitte — **Dubai, UAE**

Technology Consultant Intern — Sep 2024 – Jan 2025

- Applied NLP with a T5 model to cluster laws and regulations into thematic groups and generate representative questions for each group
- Built a 3D OT network visualization tool with React.js, Three.js, and TypeScript for real-time inspection of complex network topologies
- Built a full-stack web application with Next.js, Django, and PostgreSQL with REST APIs for project data management

PROJECTS

Argus — Jan 2023 – Present

- Built a C++20 multi-stream object detection pipeline that decodes video with FFmpeg/libav, runs YOLO inference through TensorRT, and re-encodes on the GPU with NVENC
- Profiled it with Nsight Systems and NVTX and cut a 16-stream test on one 8 GB laptop GPU from 27 to 7 seconds (about 180 to 730 frames per second) through pinned host buffers, two overlapping CUDA streams, uint8 model input, FP16 with an FP32 NMS block, and one shared CUDA context for TensorRT and NVENC
- Sharing one CUDA context across encoder sessions cut setup from 3.67 to 0.57 seconds and GPU memory from 2,744 to 1,186 MiB, and keeping NMS in FP32 held FP16 accuracy at a median IoU of 0.991

Strategic Sourcing Intelligence Agent — Apr 2026 – Present

- Built a governed enterprise procurement AI platform solo, covering 8 agentic workflows including supplier onboarding and sanctions screening, purchase requisition intake, RFP and proposal comparison, approval routing, contract redline review, goods-receipt matching, and supplier scorecards
- Architected the system for customer-owned deployment, connecting to the customer's own identity provider, private model endpoint, and object storage so no data or model traffic leaves their environment
- Routed every LLM call through a Bifrost OpenAI-compatible gateway with a sensitive mode that blocks public providers outright, so the model provider can be swapped to a customer-approved private endpoint without application changes
- Built durable human-in-the-loop approval workflows on Temporal with role-gated actions and audit events, deliberately keeping the agent unable to approve spend, reject suppliers, or release payments on its own
- Wrote 376 backend tests and a 45-case offline agent evaluation suite covering grounding, out-of-scope handling, bias, and Arabic and English routing

- Cut container memory by 650 MiB and freed 5.3 GiB of host RAM by moving to gunicorn preload so forked workers share embedding model weights through copy-on-write, after diagnosing 2.4 GiB of per-worker weight duplication

AUS / EPRI-funded research Jun 2023 – Jul 2024
American University of Sharjah
- Built simulation benchmarks for cascading natural-disaster failures on electric power grids, modelling distribution nodes as directed graphs rooted at the power source
- Designed graph node and edge classes plus failure-propagation algorithms, and visualised grid topology and damage scenarios with Matplotlib
- Integrated the simulation with MATLAB power-flow models for EPRI-funded research

zdc Jan 2025 – Present
- C++20 image encoder and decoder with zero third-party dependencies, no libpng, no zlib, no stb_image, implementing every format byte by byte against its specification using only the standard library
- Implemented the full compression stack from scratch, RFC 1951 DEFLATE with dynamic, fixed and stored Huffman blocks, LZ77 back-references, the zlib container, CRC-32 and Adler-32
- Produced zlib output byte-compatible with libpng, zlib and Pillow, shipped as both a linkable CMake library and a standalone CLI

Orb Jan 2026 – Present
Falaq.ai
- Released an Android local AI assistant on Google Play with offline chat, image understanding and voice input after a one-time model download
- Runs quantized GGUF vision-language models including Qwen3-VL, MiniCPM-V and Gemma with CPU fallback and Snapdragon Hexagon NPU acceleration
- Published the underlying React Native ONNX Runtime bindings to npm as native-onnx, with CPU, XNNPACK, NNAPI and Qualcomm QNN backend selection

LLamaPG Jan 2024 – Present
- Architected a production-ready multi-tenant RAG-as-a-Service platform automating pipeline deployment through containerized microservices with Docker, Kubernetes, and scheduled CRON jobs
- Built the backend with FastAPI, PostgreSQL, and pgvector for vector embeddings, integrating LlamaParse for document processing and the OpenAI API for semantic search

Daqa Jun 2026 – Present
- Building a dataset acquisition and preparation system that takes a description of needed data, discovers sources, acquires records and files, profiles the material, reconciles schemas, and proposes deterministic preparation plans
- Built the backend as a Go/Fiber v3 API on ent/Postgres with Valkey, including an agent and tool runtime that routes work across seven independent Python workers for annotation, synthesis, classification, image verification, RAG chunking, and document preprocessing

Wirecopy Jun 2026 – Present
- Building a tech-signal aggregation product end to end as sole developer, a single Go binary running as server or worker, a TanStack Start frontend, Postgres with pgvector for embeddings, Redis, and a Cloudflare Workers/D1/R2 production entrypoint
- Built scheduled multi-source ingestion across 20 source families covering roughly 2,500 to 4,000 free feeds and endpoints, with a two-tier sourcing policy and per-source licensing enforced in the pipeline rather than by convention

EDUCATION

American University of Sharjah **Sharjah, UAE**
Dual BS in Computer Science and Computer Engineering Sep 2020 – Jun 2024
- **Honors**: GPA 3.81 / 4.00

RESEARCH AND PUBLICATIONS

Dental Radiograph Analysis for Improved Diagnosis
Published on *IEEE Access* Jan 2024

SKILLS, LANGUAGES, INTERESTS

- **Languages**: English (Fluent), Arabic (Native)
- **Cloud & Infrastructure**: Kubernetes, Kubernetes (Deployments, Services, Ingress), HAMi GPU virtualization, Docker, Azure AKS, Azure Entra ID, Helm, Pulumi (infrastructure as code), DevOps, GitHub Actions, CI/CD, Git, Linux
- **Observability**: Grafana, Prometheus, Loki, Grafana Tempo
- **Model Serving & Inference**: vLLM, SGLang, NVIDIA Triton, TensorRT, ONNX Runtime, PyTorch, Hugging Face
- **LLM, RAG & Agentic Systems**: Core42 sovereign model endpoints, Azure AI Foundry, Bifrost gateway, LangChain, PydanticAI, Agno, FAISS, pgvector, PostgreSQL, Redis
- **Backend & Languages**: Python, Go, TypeScript, C++, SQL, FastAPI, Django, REST APIs, Microservices
- **Computer Vision**: OpenCV, YOLOv11, DepthAnythingV2
- **Frontend & Mobile**: React, Next.js, React Native
- **Interests**: Systems programming, Kubernetes internals, Open source

*This document was last updated on 2026-10-19 using ~~LaTeX~~ Typst