

Abdelhady Ali Mohamed

AI and DevOps Engineer

abdulhadi.ali.mhmd@gmail.com · 01286386834 · Cairo, Egypt

[LinkedIn](#) · [GitHub](#) · [Kaggle](#) · [Hugging Face](#)

PROFILE

AI and DevOps engineer with a software engineering foundation, focused on shipping production-grade AI systems and the infrastructure that keeps them running. Builds end-to-end: from LLM/RAG pipelines and multi-agent orchestration to CI/CD, container platforms, observability, and secure multi-tenant edges. Strong grounding in microservices, layered and clean architecture, and API design; comfortable across the backend and with the frontend fundamentals needed to ship complete internal tools.

PROFESSIONAL EXPERIENCE

AI Engineer

04/2026 – Present

AIVIS Studio

Remote

Real-Time PPE Compliance Detection System

Designed and built an end-to-end computer-vision system detecting 10 object and compliance-state classes (mask, gloves, hairnet, chest tag, knife, phone, person, and their non-compliant states), deployed at production sites on edge devices.

- Built an automated annotation pipeline (data engine) fusing YOLO11 + YOLO-World + OWLv2 via Weighted Box Fusion with custom post-processing (per-class confidence gating, mutual-exclusion resolution, class-agnostic NMS, cross-model agreement filtering); raised label F1 from 0.39 to 0.81 and cut false positives 95% (9,490 → 619), integrated with Roboflow for human-in-the-loop review.
- Solved the hardest sub-problem — present/absent PPE state — after proving zero-shot CLIP and pose-geometry approaches failed; trained a supervised CLIP ViT-L/14 + logistic-regression classifier per body region, reaching 99% / 97% / 95% accuracy on mask, hairnet, and glove compliance states.
- Trained a YOLO11 object detector with distributed multi-GPU training (DDP) and a leakage-safe data-splitting pipeline (video-timestamp clustering) that prevents near-duplicate frames leaking between train/test; reached mAP@50 0.77 with a custom per-class evaluation harness.
- Deployed to production on NVIDIA Jetson Orin NX edge devices (INT8 TensorRT); added ByteTrack for multi-object video tracking and temporal consistency.

Stack: Python, PyTorch, Ultralytics YOLO11, Hugging Face Transformers (OWLv2), YOLO-World, OpenCLIP, scikit-learn, OpenCV, ensemble-boxes, Roboflow, NVIDIA Jetson Orin NX (INT8 TensorRT), ByteTrack.

AI and DevOps Engineer

04/2025 – 04/2026

Zedny INC

Cairo, Egypt

Internal DevOps Dashboard

Built an internal DevOps dashboard unifying VM fleet and Kubernetes cluster management, live infrastructure monitoring, and DigitalOcean drift detection for a small ops team.

- Core platform: role-based auth (bcrypt, session + CSRF), VM/cluster CRUD with AES-GCM-encrypted SSH keys and kubeconfigs, and full audit trail on every mutating action.
- Self-hosted observability stack (Prometheus, Grafana, Loki, Alertmanager) with custom PromQL rules and webhook-driven Slack routing; declarative services inventory with blackbox-exporter probing from multiple network vantage points.
- DigitalOcean integration: rate-limit-aware API polling syncs droplet / firewall / volume / reserved-IP / Kubernetes-cluster state, with drift detection flagging services declared reachable that a Cloud Firewall doesn't actually allow.
- Kubernetes management: per-cluster API client caching, least-privilege kube-state-metrics + Grafana Alloy bundle via view-only RBAC, read-only cluster resource browsing, WebSocket pod-log tailing, and Ingress resources wired into the reachability-probe pipeline.

- Quality: pytest + Vitest on every secrets-touching and mutating code path; GitHub Actions CI with a real Postgres service container; Conventional Commits and documented ADRs.

Stack: FastAPI, SQLAlchemy (async), PostgreSQL, Redis, Alembic, Python 3.12, React, TypeScript, shadcn/ui (Tailwind + Radix), TanStack Query, Docker Compose, Prometheus, Grafana, Loki, Alertmanager, Grafana Alloy, kube-state-metrics, Kubernetes RBAC, pytest, Vitest, GitHub Actions.

Interactive Course Agents — AI Multi-Agent Course-Generation Platform

Backend platform that turns narrated lecture scripts/videos into interactive courses — automatically generating content-matched visuals (charts, infographics, diagrams, images, tables, 3D, and code-authored Manim animations), instructor layouts, and time-aligned keyword overlays — built on a FastAPI + LangGraph multi-agent pipeline.

- Deterministic representation map routing each paragraph's instruction type to the optimal visual agent (chart / infographic / image / table / Manim / 3D) via a pure-Python rule engine with content-signal overrides; diagnosed and fixed a visual-monoculture defect misrouting nearly all paragraphs to expensive animation (~\$3/run), restoring a balanced visual mix.
- Correct-by-construction animation component library (charts, tables, concept cards, callouts) with render-and-extract snapshot tests, driving broken-render rate to ~0; layered a type-aware planner tailoring storyboard beats to content (equation walkthroughs, process build-ups, cycle loops, timelines).
- Restored the production Manim rendering pipeline (recovered a missing worker package that had disabled the feature) with a self-correcting author → extract → repair loop, graceful degradation, and budget / timeout controls.
- Layout engine reclaims reserved space when captions are hidden; CI/CD concurrency guard on auto-deploy eliminates Docker container-recreate race conditions that were failing production deploys.
- Sustained 1,300+ test suite green with zero regressions across incremental PRs on a shared integration branch with local + live Swagger verification.

Stack: Python, FastAPI, LangGraph, LangChain, PostgreSQL (SQLAlchemy async + Alembic), Redis, Docker, OpenAI / Anthropic / Gemini LLMs, Manim, pytest.

DevOps Operations (parallel to project work)

- Reliability & CI/CD: rebuilt an entire production LMS from a dead VM with no backups (prod + dev clean rebuild, architecture consolidation); built self-healing GitHub Actions pipelines across three services with CI gate → build/push → deploy → migration → health-gate → verification → automatic rollback, self-bootstrapping from a .env.
- Multi-tenant edge & environment isolation: prod and dev fully isolated on shared VMs (separate Compose projects, networks, volumes, ports); host nginx across multi-app servers with Let's Encrypt (HTTP-01 and DNS-01 wildcard); multi-tenant subdomains gated by an edge allow-list with API-driven auto-renewing cert automation.
- Data platform & AI-systems ops: operated PostgreSQL / Supabase, Alembic, pgvector, MongoDB, Qdrant, Redis, MinIO / S3; ran AI workloads (RAG services, OpenAI embeddings + vector search, Celery / Modal workers) with vector-DB provisioning and embedding pipelines; rewrote a broken Supabase Management-API migration runner back to standard Alembic.
- Security posture: secrets on the VM / in GitHub Secrets (never in pipelines), token rotation with log/history scrubs, loopback-bound datastores, firewall allow-listing for exposed services.

AI Intern

01/2025 – 03/2025

ACME – Integrated Engineering Systems (IES)

Cairo, Egypt

- Developed and evaluated YOLOv8 object detection models for automation use cases within national-scale digital transformation initiatives.
- Built NLP classification and NER models with Hugging Face Transformers; contributed to data annotation, preprocessing, and cross-validation benchmarking pipelines.
- Delivered OpenCV + TensorFlow computer vision pipelines for image segmentation, integrated into production backend services.
- Contributed to sprint planning, code reviews, and technical documentation for AI subsystems on a cross-functional engineering team.

Freelance — Computer Vision & AI Developer

02/2025 – 03/2025

Independent

Remote

- Delivered end-to-end OCR systems (PyTorch, Tesseract, OpenCV) for Arabic and Latin document formats with robust pre- and post-processing.
- Built real-time YOLOv8 object detection services deployed as FastAPI REST APIs; integrated the Mistral API for downstream LLM-based document interpretation.
- Ran the full ML lifecycle independently — data curation, training, evaluation, versioning, containerized deployment — on time to client specification.

PROJECTS

Autonomous DevOps Agent SDK

Multi-agent Python SDK that autonomously deploys code, monitors infrastructure, investigates incidents with RAG, and executes safe remediation.

- Designed a layered agent architecture with a central orchestrator, event bus, and pluggable agents (CI/CD, Monitoring, Knowledge/RAG, Self-Healing, Alerting) that collaborate through a shared context and state manager.
- Architected a provider abstraction layer isolating monitoring (Prometheus, Datadog, CloudWatch), logging (Loki, Elasticsearch), LLMs (via LiteLLM), vector DBs (Qdrant, FAISS), infrastructure (Docker, Kubernetes, SSH), and notification backends (Slack, email, PagerDuty) behind uniform interfaces for zero-lock-in swaps.
- Designed the Knowledge Agent's RAG pipeline as a single Qdrant collection with a source_type discriminator over runbooks, past incidents, architecture docs, and logs; knowledge snapshots distributed via GitHub Releases for reproducible build-time-only data collection.
- Delivered the SDK across Python API, CLI (devops-agent monitor / investigate / remediate), and FastAPI service surfaces to fit both automation and interactive workflows.

Stack: Python, FastAPI, LiteLLM, Qdrant, Prometheus, Loki, Kubernetes, Docker, GitHub Actions.

MAISYS — Medical AI System (Graduation Project, Team Lead)

Bilingual (Arabic / English) medical AI platform architected as 13 microservices on AWS ECS Fargate, orchestrated via LangGraph.

- Led a small team from concept and data collection through AI development, infrastructure design, and deployment; owned per-service governance and locked architectural decisions.
- Engineered a dual-embedding RAG pipeline (hybrid BM25 + dense vector search) over curated medical sources (PubMed, RxNorm, FDA); measured hallucination-rate reduction to 3.2% vs. 14.7% baseline.
- Built five AI modules — Medical Chatbot, Drug Interaction Checker, Symptom Checker, Lab Test Explainer, Research Assistant — with OCR pipelines converting lab reports and medical documents into structured, explainable insights.
- Delivered cross-platform frontends (React web + Flutter mobile) with full Arabic RTL support; integrated a multi-layer AI safety system with emergency detection, disclaimers, and source-linked citations.
- Owned infrastructure: Terraform IaC, GitHub Actions CI/CD; ran a LoRA fine-tuning pipeline on medical-domain LLMs; used a snapshot-then-destroy demo pattern for cost control.

Stack: Python, FastAPI, LangGraph, RAG, BM25, PubMed / RxNorm / FDA APIs, LoRA fine-tuning, OCR, Terraform, AWS ECS Fargate, GitHub Actions, Docker, React, Flutter.

CodeFlow — AI-Powered Code Intelligence Platform

Production-grade AI platform that transforms codebases into living documentation, architecture diagrams, and interactive knowledge systems.

- Engineered an end-to-end pipeline for repository cloning, deep static analysis, symbol extraction, dependency mapping, and semantic embedding — enabling RAG-powered chat with precise, source-cited answers over large codebases.
- Orchestrated CrewAI autonomous agents for code analysis, documentation generation, diagram synthesis, and AI-assisted refactoring / review.
- Delivered a VS Code-like web workspace with live editing, Git integration, versioned AI artifacts, and collaborative workflows on a secure FastAPI + PostgreSQL backend and modern React UI; optimized for GPU-accelerated inference with reliable CPU fallback.

Stack: Python, FastAPI, React, TypeScript, CrewAI, RAG, ChromaDB, Sentence Transformers, Google Gemini, Qwen3, Ollama, Mermaid CLI, PostgreSQL, pgvector.

EDUCATION

Bachelor's Degree in Computer Science and Artificial Intelligence

09/2022 – 06/2026

Benha University — Faculty of Computers and Artificial Intelligence

SKILLS

Programming & Engineering: Python, TypeScript, Clean Code Practices, Modular System Design, Performance Optimization, Testing (pytest, Vitest)

Architecture & Design: Microservices Architecture, Layered / Clean Architecture, API Design, Multi-Agent System Design, Provider Abstraction Patterns, Event-Driven Systems

AI & Machine Learning: Machine Learning, Deep Learning, Retrieval-Augmented Generation (RAG), Multi-Agent Systems (LangGraph, CrewAI), LLM Integration & Orchestration, Fine-Tuning (LoRA), Model Evaluation & Optimization, Ensemble Methods (Weighted Box Fusion), Computer Vision, NLP

AI Frameworks & Libraries: PyTorch, TensorFlow, Hugging Face Transformers, LangGraph, LangChain, CrewAI, Ultralytics YOLO11 / YOLOv8, OpenCLIP, Sentence Transformers, LiteLLM

Backend (primary): FastAPI, REST APIs, WebSockets, Async Programming, JWT / Session Auth, SQLAlchemy (async), Alembic Migrations, Celery, Modal

Frontend (working knowledge): React, TypeScript, TanStack Query, shadcn/ui (Tailwind + Radix), Vite

MLOps & DevOps: Docker, Docker Compose, Kubernetes (RBAC, kube-state-metrics), GitHub Actions (CI/CD), GHCR / Docker Hub, Blue/Green-style Rollback, Self-Healing Deploys, GPU-Accelerated Inference

Cloud & Infrastructure: AWS ECS Fargate, DigitalOcean (Droplets, Firewalls, Managed Kubernetes), Supabase, nginx (reverse proxy, multi-tenant edge), Let's Encrypt (HTTP-01, DNS-01 wildcard), Cloudflare, Terraform, Linux / SSH ops

Observability: Prometheus (PromQL), Grafana, Loki, Alertmanager, Grafana Alloy, blackbox-exporter

Data & Storage: PostgreSQL, pgvector, ChromaDB, FAISS, Qdrant, Redis, MongoDB, MinIO / S3

AI Systems & NLP: Semantic Search, Vector Databases, OCR (Tesseract), Speech-to-Text, Text-to-Speech, Prompt Engineering, LLM Provider Abstraction

Security & Ops: Secrets Management, Token Rotation, AES-GCM Encryption, CSRF Protection, Firewall Allow-listing, Audit Trails

Tools & Practices: Git, Conventional Commits, Human-in-the-Loop Labeling, Roboflow, Kaggle GPU, Reproducible Experimentation, Ablation Studies

CERTIFICATES

[HCIA Huawei](#) · [ACME-IES](#) · [ACME-Recommendation](#) · [NTI Training](#)

LANGUAGES

Arabic — Native English — Professional