MUKESH N

Senior AI Engineer • RAG & Multi-Agent Systems • LangGraph, MCP | Full-Stack Python (FastAPI/React)
Dallas, TX • nmukeshoff@gmail.com • +1 (214) 356-4492 • linkedin.com/in/mukeshnoff

PROFESSIONAL SUMMARY

I build production AI systems that Fortune 500 companies actually ship to millions of users. Started building DevOps pipelines and GPU infrastructure at DXC Technologies, spent two years scaling AT&T's internal generative AI platform (Ask AT&T) for 68,000+ employees, and now architect CVS Health's AI-native consumer engagement platform—connecting CVS Pharmacy, Caremark, and Aetna through multi-agent systems, HIPAA-compliant RAG pipelines, and GPU-backed inference on GCP Vertex AI.

My stack is Python (FastAPI, Django) on the backend, React and Vue.js on the frontend, Kubernetes and Terraform for infrastructure. What separates me from most AI engineers is evaluation: I build the harnesses that measure whether AI systems actually work—LLM-as-judge pipelines, golden test datasets, hallucination detection with NeMo Guardrails, and drift monitoring across production workloads. The work I'm proudest of includes RAG pipelines that beat baseline retrieval by 35%+, multi-agent orchestration with sub-second latency, automated evaluation across 8 production use cases, and ~$1M in annualized infrastructure savings through intelligent model routing.

TECHNICAL SKILLS

Languages: Python, JavaScript, TypeScript, SQL, Bash
AI / ML / LLM: LangGraph, MCP (Model Context Protocol), LangChain, CrewAI, Haystack, RAG (hybrid vector search, semantic chunking, re-ranking), vLLM, NVIDIA Triton, NVIDIA NeMo Guardrails, Semantic Kernel, Ray, prompt engineering, LLM-as-judge evaluation, MLflow, Weights & Biases, LangSmith
Frontend: React.js (Hooks, Context, Redux Toolkit), Vue.js 3 (Composition API, Pinia), HTML5, CSS3, Tailwind, Streamlit, accessibility (WCAG), responsive layouts
Backend: FastAPI, Django, Flask, Node.js, REST API design, GraphQL (Apollo, Strawberry), WebSockets, Server-Sent Events, OAuth2/JWT, microservices, event-driven architecture, Kafka
Data: PostgreSQL, MySQL, Redis, MongoDB, Cosmos DB, OpenSearch, Pinecone, Qdrant, Snowflake, ETL pipelines (Spark, Flink), Pandas, NumPy
Cloud & DevOps: AWS (Bedrock, EKS, Lambda, SageMaker, CloudFormation), GCP (Vertex AI, Vertex AI Search, GKE), Azure (OpenAI, AKS, Stream Analytics), Docker, Kubernetes, Helm, Terraform, Pulumi, GitHub Actions, Jenkins, Ansible, Prometheus, Grafana, ELK Stack, OpenTelemetry
Testing & Evaluation: Pytest, Jest, React Testing Library, Playwright, Cypress, Great Expectations, golden dataset evaluation, hallucination detection, drift monitoring, A/B testing, Postman, n8n, Unstructured.io

PROFESSIONAL EXPERIENCE

CVS Health | Dallas, TX May 2025 – Present
Senior AI Engineer

- Architect and build core components of CVS Health's AI-native consumer engagement platform on GCP Vertex AI, connecting experiences across CVS Pharmacy, Caremark, and Aetna for millions of members—owning the full stack from GPU inference clusters to React-based agent configuration portals.
- Design multi-agent communication pipelines using LangGraph stateful agent graphs and Model Context Protocol (MCP), achieving sub-second latency for secure tool invocation across pharmacy, insurance, and care-delivery healthcare systems with strict permission boundaries and full request/response audit logging.
- Build HIPAA-compliant RAG pipelines with Vertex AI Search and Unstructured.io for clinical decision support, drug interaction checks, and formulary optimization. Improved retrieval precision 35%+ over baseline keyword search through semantic chunking, hybrid vector search across Pinecone and OpenSearch, and cross-encoder re-ranking.
- Implemented NVIDIA NeMo Guardrails as the AI safety layer across all patient-facing outputs—real-time hallucination detection, toxicity filtering, and PII redaction—ensuring every response meets HIPAA and SOC 2 compliance standards before reaching end users.
- Built LLM-as-judge evaluation pipelines and curated golden test datasets across pharmacy, insurance, and care-delivery scenarios, enabling automated regression testing before every model update and continuous drift monitoring across production workloads.
- Deploy and operate vLLM on GKE GPU clusters (NVIDIA A100, T4) with horizontal pod auto-scaling, automated failover, and rolling deploys. Maintain a 99.9% SLA across multiple production inference workloads.
- Deployed LiteLLM as a unified model gateway with intelligent routing across GCP Vertex AI, AWS Bedrock, and Azure OpenAI, per-team rate limits, and cost guardrails. Built React-based observability dashboards tracking inference cost, latency, and throughput. Contributed to $1M+ in annualized infrastructure savings.
- Built Python backend services on FastAPI exposing REST and GraphQL endpoints for the agent platform and a lending decisioning system processing 50,000+ daily decisions, with OAuth2 authentication, JWT session management, and SOX-compliant audit logging.
- Developed a React + TypeScript admin console with real-time WebSocket feeds for model performance, sortable and filterable views over decision data, explainability drill-downs for individual cases, and streamed token rendering with partial-response loading states.
- Built Vue.js 3 interfaces using the Composition API and Pinia for the multimodal consumer health platform, integrated with FastAPI backend services through HIPAA-compliant data flows and end-to-end encryption.
- Designed a shared React component library with Redux Toolkit for state management across internal applications, enforcing visual consistency and cutting build time for new feature work by roughly 40%.
- Built n8n-based agentic workflows so non-engineering teams can deploy document-processing and claims-triage agents independently, reducing time-to-deploy from weeks to hours.
- Engineered iterative prompt engineering workflows with versioned prompt templates, systematic A/B testing across model providers, and automated quality scoring—treating prompts as testable, deployable artifacts rather than ad-hoc strings.
- Engineered ETL pipelines using Apache Spark to process claims and decisioning data into Snowflake, with Pandas and NumPy for pre-processing and quality validation before warehouse ingestion.
- Rapidly prototyped internal data applications using Streamlit, giving product and analytics teams a way to visualize model performance and inspect claims-processing outcomes without waiting for full UI builds.
- Set up CI/CD pipelines in GitHub Actions covering frontend, backend, and infrastructure—with linting, unit and integration tests (Jest, Playwright, Pytest), accessibility checks, and automated blue/green deployment on every pull request.

AT&T | Dallas, TX Jan 2023 – Apr 2025
Full Stack Engineer

- Contributed core React frontend features to Ask AT&T, the company's internal generative AI platform used by 68,000+ employees: multi-turn chat interface, multi-model selector (GPT-4, LLaMA, Falcon), persistent conversation history, file attachments, markdown rendering, and streaming token-by-token responses.
- Built production Azure OpenAI chatbot services with Semantic Kernel orchestration handling millions of monthly interactions with sub-2-second response time, enabling employees to interact with enterprise data through conversational AI.
- Rewrote 20+ legacy REST APIs as LLM-compatible tool interfaces using function-calling definitions and MCP for standardized tool orchestration, powering billing automation, account troubleshooting, and self-service resolution across customer-facing and internal channels.

- Built automated evaluation pipelines covering 8 production use cases—measuring accuracy, hallucination rate, and task completion using LLM-as-judge scoring, curated golden datasets, and regression suites that run before each model update, with weekly reporting to leadership.
- Deployed NVIDIA Triton Inference Server on AKS for ensemble ML models with dynamic batching and multi-model concurrency, improving throughput 4x over the previous serving setup and bringing prediction latency under one second.
- Fine-tuned and deployed NLP models including Tiny BERT and DeBERTa into production, improving the accuracy of AT&T's global user feedback analysis system processing input from 100+ languages via integrated ASR, TTS, and NMT pipelines.
- Built an Azure-native fraud detection pipeline using Stream Analytics, Event Hubs, Kafka, and ensemble ML models, processing real-time event streams to flag anomalous account activity.
- Built an operations portal in React with a Flask backend on Cosmos DB: real-time agent monitoring, cost dashboards, per-team usage analytics, and admin controls consumed through REST and GraphQL APIs.
- Built n8n automation workflows for ticket classification, sentiment routing, and knowledge base auto-updates, enabling faster triage without manual intervention.
- Cut claim adjudication time from 45 minutes to 12 by building a RAG-based claims-processing system using LangChain and Haystack, with automated entity extraction, document verification, and confidence-scored outputs.
- Developed iterative prompt engineering practices across the platform—systematic prompt versioning, temperature and chain-of-thought tuning, and automated quality benchmarks—reducing hallucination rates by 25% across production use cases.
- Optimized token consumption through semantic caching with Redis, response deduplication, and tiered model selection (routing simpler queries to lighter models), with live cost-tracking surfaced through the operations dashboard.
- Improved client-side performance with code-splitting, lazy-loaded conversation history, and memoized rendering, keeping sessions with 500+ messages responsive across Chrome, Safari, Edge, and mobile webviews.
- Designed database schemas and managed data integrity across Cosmos DB and PostgreSQL, optimizing query performance through indexing strategies, materialized views, and connection pooling for high-throughput workloads.
- Worked with compliance and responsible-AI teams on model cards, bias auditing through Weights & Biases, and explainability reports surfaced directly inside the platform UI.

DXC Technologies | Bengaluru, India May 2020 – Jun 2022
Python Developer & AI Infrastructure Engineer

- Built CI/CD pipelines for Python microservices using Jenkins, Terraform, and AWS CloudFormation, with automated testing gates, secret management through AWS Secrets Manager, and rollback procedures built into the deployment process.
- Managed Kubernetes clusters across EKS and on-premises infrastructure for GPU-accelerated model training and inference, using Ray and Dask for distributed compute orchestration across multi-team ML experimentation workloads.
- Automated provisioning of VPCs, GPU compute instances, and networking infrastructure using Terraform and Pulumi, reducing environment provisioning from days to minutes and eliminating manual configuration drift.
- Wrote Ansible playbooks to automate the setup and configuration of AWS EC2 instances and EKS clusters, ensuring consistent, repeatable environments across development, staging, and production.
- Built monitoring and observability from scratch with Prometheus, Grafana, and custom exporters for service latency, throughput, GPU utilization, and infrastructure cost, integrated with PagerDuty for on-call alerting.
- Piped application and system logs into a centralized ELK stack (Elasticsearch, Logstash, Kibana) for cross-service debugging, root-cause analysis, and compliance auditing.
- Wrote data-validation pipelines using Pandas, Scikit-learn, and Great Expectations across 10+ datasets, with MLflow and Weights & Biases experiment tracking integrated into model training workflows.
- Built document-labeling and annotation pipelines for deep-learning model training across 50,000+ samples, with evaluation benchmarks, accuracy reporting, and version-controlled dataset management for OCR models in production.
- Developed Python backend services with Django and Flask, creating RESTful APIs for internal tooling with OAuth2 authentication, input validation, and test coverage using Pytest.
- Built internal dashboards in React for engineering and business teams to visualize pipeline health, model training metrics, and infrastructure utilization.

EDUCATION

M.S. Computer Science (AI & Machine Learning), University of South Florida
Coursework: Deep Learning, NLP, Distributed Systems, Advanced Algorithms, Cloud Computing, Statistical Machine Learning, Reinforcement Learning

B.Tech Computer Science & Engineering, Geethanjali College of Engineering and Technology, India

CERTIFICATIONS

AWS Certified Machine Learning Engineer – Associate • Generative AI with Large Language Models (DeepLearning.AI × AWS)