RAMACHARAN REDDY KASIREDDY

kramacharanreddy@gmail.com · +1 716-292-6481 · ramacharanreddy-k · ramacharanreddy-k

AI Engineer with 2+ years of experience designing and delivering production AI systems – multi-agent LangGraph workflows, RAG pipelines with hallucination reduction, vector search, and data ingestion engines – across both AWS and Azure, with end-to-end ownership from Terraform infrastructure to client-facing delivery.

SKILLS

AI/ML & LLMs	LangChain, LangGraph, MCP, RAG (hybrid k-NN + BM25), Prompt Engineering, Fine-Tuning, Multi-Agent Systems, ReAct, Tool Calling, Structured Outputs, Langfuse
NLP	Tokenization, Transformer Embeddings, Conversational AI, Semantic Search, Intent Classification, Text Generation
LLM Platforms	Azure AI Foundry, Azure OpenAI (GPT-4, Embeddings), AWS Bedrock (Claude, Titan Embeddings), Google Gemini
Deep Learning	PyTorch, TensorFlow, Hugging Face Transformers, LoRA, PEFT, QLoRA, Quantization, Distillation
Programming	Python, SQL, JavaScript, Java, React, FastAPI, Uvicorn, Pydantic, Jinja, Playwright, pytest
Databases	PostgreSQL, Cosmos DB, MongoDB, OpenSearch, SQLAlchemy
AWS	Bedrock, OpenSearch, EKS, ECR, Lambda, SQS, S3, Transcribe, DynamoDB, CodePipeline, KMS, IAM (IRSA)
Azure	OpenAI Service, Cosmos DB, AKS, ACR, Key Vault, Blob Storage, Entra ID, Managed Identity, Logic Apps
DevOps & Agile	Docker, Kubernetes, Helm, Terraform, Airflow, Git, GitHub Actions, Jenkins, GitLab CI/CD, Agile/Scrum
AI-Assisted Dev	GitHub Copilot, Cursor, Claude Code
APIs & Security	RESTful APIs, OpenAPI/Swagger, OAuth 2.0, JWT, MSAL, JWKS, SSE, WebSockets

WORK EXPERIENCE

Feuji INC AI Engineer | Aug 2024 – Present

Customer Analytics AI – PLZ Corp

Python, FastAPI, LangGraph, Azure OpenAI, Azure AI Foundry, Cosmos DB, PostgreSQL, Terraform, React

- **Architected a full-stack AI analytics platform** across 4 microservices (data engine, SSO gateway, conversational agent, React SPA frontend) deployed on AKS, **eliminating 48 hours/week of manual analysis** and **saving $150K/year** in equivalent headcount.
- **Engineered an AI agent using LangGraph and Azure OpenAI** with 20 specialized tools and a 7-layer middleware stack, backed by dual-domain RAG over Cosmos DB vector search (DiskANN, 1536-dim embeddings) **grounding all responses in verified data to reduce hallucinations**, augmented by summarization and short-term memory for conversational coherence.
- **Built an 8-stage async data pipeline** that ingests structured and unstructured data, transforms through a parse-chunk-embed-index workflow, and persists to a **12-table PostgreSQL warehouse** and a **Cosmos DB vector store**.
- **Designed a centralized SSO gateway** using FastAPI RESTful APIs and MSAL with OAuth2 authorization code flow, Azure AD group-based access control, JWKS S2S token validation, and zero hardcoded secrets via Key Vault and Managed Identity.
- **Provisioned the entire Azure environment using Terraform** across 7 modules with passwordless RBAC throughout, covering compute, data, AI, identity, storage, scheduling, and communication services.

Ask the Prospect (ATP) – TriNet

Python, FastAPI, LangGraph, AWS Bedrock, OpenSearch, Aurora, Langfuse, Airflow, EKS

- **Built a multi-mode LangGraph agent** powering 4 capabilities – RAG chatbot (dual-index: training docs + Gong call transcripts) with **memory continuity using short-term and long-term memory** via LangGraph PG checkpointer and store, persona-based sales role-play with AI coaching, analytics chat with text-to-OpenSearch query generation, and automated Jeopardy question generation.
- **Engineered a sales analytics pipeline** where LLM enriches each Gong call transcript with derived columns (win/loss, objections, competition mentions), then serves aggregate queries by generating OpenSearch queries from natural language via text-to-schema mapping.
- **Integrated Langfuse for full observability and bias detection** across all agent interactions, using **LLM-as-judge evaluations** for output quality scoring alongside trace-level debugging, latency monitoring, and cost tracking across every LLM call and tool execution in production.
- **Served as solution architect and client-facing lead**, owning the end-to-end technical vision across 4 microservices and 2 Airflow-orchestrated data pipelines deployed on EKS.

ReQon – Automated Testing Tool

Python, LangChain, LangGraph, Google Gemini, Playwright MCP, Aurora PostgreSQL, Docker, pytest

- **Engineered a LangGraph agent with Playwright MCP** that replays a user-recorded browser flow, autonomously

navigates every page multiple times, and extracts locators, identifiers, functional elements, and behavioral data – storing the full application map in S3 and Aurora PostgreSQL.
- **Automated the entire QA authoring workflow** – from captured browser behavior the agent generates structured test cases in Aurora PG, executes them via Playwright MCP, and outputs verified pass/fail results, eliminating manual test case writing.
- **Built a code generation engine** using Jinja templates that converts captured page metadata into **pytest-compatible regression scripts** with **Page Object Model classes** for every page discovered, producing a ready-to-run test suite from a single recording.
- **Designed and dockerized 3 microservices** – Recorder (Playwright codegen capture), MCP Service (AI agent, knowledge extraction, test and script generation), and Execution Service (isolated regression runner) – saving **100+ team-hours weekly** and cutting QA costs by an estimated **$230K/year**.

AI-Powered IVR Automation – Cotiviti

Java, Python, LangGraph, AWS Bedrock, Transcribe, DynamoDB, Lambda, SQS, KMS, Twilio

- **Designed an AI pathfinding workflow using LangGraph and AWS Bedrock** that autonomously navigates Medicare's IVR system daily – recording calls via Twilio, transcribing with **AWS Transcribe**, and mapping the correct DTMF navigation path to **DynamoDB** as the source of truth for **200,000+ daily production calls**.
- **Built a mock Medicare IVR simulator** in Java as a regression testing harness, validating AI agent accuracy against simulated IVR decision trees before production deployment.
- **Configured the AWS CI/CD pipeline** (CodeCommit, CodeBuild, CodePipeline, CodeDeploy) with **KMS-encrypted PII handling** across all environments.
- Part of a **cross-functional team** that saved **730+ engineer-hours annually** and cut infrastructure costs by **˜60%** through serverless auto-scaling, with an operations dashboard for regulatory oversight.

University at Buffalo AI Research Assistant | Jan 2024 – May 2024

Energy-Efficient LLM Inference and Carbon-Aware Optimization

Python, PyTorch, TensorFlow, Hugging Face, NumPy, Pandas

- Conducted applied research on **LLM fine-tuning, model evaluation, and performance tuning** by benchmarking **parameter-efficient fine-tuning (PEFT)** techniques – including **LoRA**, **knowledge distillation**, and **post-training quantization (4-bit/8-bit)** – on GPT-2 across multiple task configurations, measuring GPU energy consumption and carbon emissions.
- Achieved a **19.8% reduction in CO2 emissions** with only a **6% increase in perplexity** using 8-bit quantization, demonstrating that significant energy savings are attainable with minimal degradation in language modeling quality.
- Demonstrated a **45.2% emissions reduction** by combining distillation with quantization, quantifying the accuracy-efficiency trade-off curve to identify viable deployment configurations for resource-constrained and carbon-aware inference environments.

Flable AI Machine Learning Engineer | Mar 2023 – Aug 2023

Flable Digital Assistant

Python, LangChain, OpenAI, PostgreSQL, FastAPI, Docker, Jenkins, GitLab

- Built a **LangChain-based conversational AI** integrated with **OpenAI** that classifies user intent using **NLP techniques (transformer-based embeddings)**, maps queries to structured data categories in **PostgreSQL**, and generates natural-language responses grounded in real database results.
- Developed an **on-the-fly data visualization engine** that dynamically queries PostgreSQL, renders chart images from the returned data, and delivers them inline through the **Telegram bot interface** – giving non-technical users instant access to business analytics without dashboards or manual reporting.
- Deployed the full stack (**FastAPI** backend + Telegram bot frontend) using **Docker** containers with **GitLab CI/CD** and **Jenkins** pipelines, reducing client's weekly analysis time by **60%**.

OPEN SOURCE

langgraph-checkpoint-cosmos | github.com/LangModule/checkpoint-cosmos

- Published a LangGraph checkpoint saver for Azure Cosmos DB with sync/async support, tip-document optimization, transactional consistency, and DefaultAzureCredential keyless authentication.

EDUCATION

University at Buffalo, The State University of New York Buffalo, New York
Master of Science, Computer Science, CGPA: 3.7/4 2023 – 2025
Coursework: Data Intensive Computing, Computer Vision, Machine Learning, Deep Learning