# Real-Time ML Platform — Capgemini

Saketh Sai Nigam Kanduri — Software Engineer, Capgemini

## ■■ Project Overview

Developed and maintained a production-grade Real-Time ML Platform that integrated Data Engineering, MLOps, and Backend microservices on AWS and Databricks. Enabled enterprise apps to consume ML predictions with low latency and high reliability.

## ■ System Architecture

| | |
|---|---|
| Data Sources | Kafka, RDS, S3, Partner APIs |
| Ingestion & Processing | Databricks Autoloader, Delta Lake (Bronze → Silver → Gold) |
| Feature Store | Databricks Feature Store (Train/Serve parity) |
| Model Management | MLflow Tracking & Registry |
| Serving Layer | FastAPI microservices (Dockerized) on EKS & OpenShift |
| CI/CD | Jenkins & GitHub Actions (Tests, Security, Helm Deploy) |
| Monitoring | Grafana, Prometheus, Splunk (Drift, Latency, Errors) |

## ■ Key Results

- Improved system uptime and scalability by **35%**
- Achieved **<150 ms** average prediction latency
- Reduced model downtime via proactive drift detection and automated rollback
- Deployed containerized ML APIs across Kubernetes and OpenShift clusters seamlessly

## ■ Technical Stack

Python, FastAPI, Databricks, MLflow, Delta Lake, Kafka, AWS (EKS, S3, DMS, IAM), Docker, Jenkins, GitHub Actions, Helm, Prometheus, Grafana, Splunk

## ■ End-to-End Workflow

1■■ Ingest data from Kafka & RDS → Delta Lake via Databricks Autoloader
2■■ Transform & validate data → Register in Feature Store
3■■ Collaborate with Data Scientists → Train and version models in MLflow
4■■ Deploy model APIs with FastAPI → Containerize & push to EKS/OpenShift
5■■ CI/CD automation (build, test, deploy) via Jenkins & GitHub Actions
6■■ Continuous monitoring with Grafana & Splunk → Trigger rollback on anomalies

## ■ Summary

Delivered a scalable, observable, and automated ML deployment ecosystem bridging Data Engineering and MLOps. Enabled faster releases, resilient model rollouts, and measurable business impact for enterprise clients.