Iteration 2 Report: Real-Time IoT Data Streaming Pipeline

Sai Kiran Anumalla, Varun Sai Danduri

June 9, 2025

1. Project Kickoff

Goals: Design and implement a fully open-source, cloud-hosted IoT data streaming pipeline using AWS infrastructure. The system should handle high-throughput real-time data ingestion, stream processing, durable storage, and live monitoring.

Scope: The project will simulate real-time sensor data using AWS Lambda, ingest it into Apache Kafka (hosted on EC2), process it using Apache Spark (on EC2 or EMR), and store results in PostgreSQL RDS and S3 (Parquet format). Visualization will be handled through Grafana dashboards, with alerting managed by Prometheus + Alertmanager. The infrastructure will be provisioned using AWS CDK.

Deliverables and Milestones:

- **Weeks 1–2:** CDK environment setup, VPC and EC2 scaffolding
- **Weeks 3–4:** Kafka on EC2, Lambda sensor simulator
- **Weeks 5–6:** Spark ingestion and processing, PostgreSQL RDS + S3 write
- **Weeks 7–8:** Prometheus and Grafana deployment, dashboard prototyping
- **Weeks 9–10:** Alert rule definition, integration, final testing, and report

Dataset: We will use synthetic sensor data generated in real-time by AWS Lambda. Each simulated sensor emits temperature, humidity, and air quality values every few seconds.

Team Readiness: The team has completed architectural planning, task division, and timeline structuring. GitHub repository and tracker are ready.

2. Team Discussions

Sai Kiran Anumalla:

- Strengths: AWS CDK, infrastructure provisioning, backend architecture
- Responsibilities: CDK setup, Kafka and Spark deployment, PostgreSQL RDS setup

Varun Sai Danduri:

- Strengths: Python, scripting, dashboards, monitoring
- Responsibilities: Lambda sensor simulator, Prometheus + Grafana setup, dashboard and alerting logic

Programming Languages and Platforms:

- TypeScript (CDK), Python (Lambda), SQL
- AWS (EC2, Lambda, RDS, S3), Docker, SSH

Identified Gaps: Initial learning is needed around Spark streaming on EC2 and Prometheus configuration, which will be addressed in early development phases.

3. Skills and Tools Assessment

Tools and Frameworks:

- **Streaming:** Apache Kafka on EC2, Spark Structured Streaming
- **Storage:** PostgreSQL RDS, Amazon S3 (Parquet format)
- **Monitoring:** Prometheus, Alertmanager, Node Exporter
- **Visualization:** Grafana (self-hosted on EC2)
- **Infrastructure-as-Code:** AWS CDK (TypeScript)

External Resources: We are leveraging official open-source documentation and AWS examples for setup. No external mentors are currently involved.

Role Clarity: Responsibilities are clearly distributed and each team member is accountable for specific components based on technical strengths.

4. Submission for This Iteration

Tasks Completed:

- Finalized project architecture and open-source tool selection
- Discussed team roles, technologies, and deliverables
- Created GitHub repository for project tracking
- Completed the Excel tracker with milestones and task ownership

Challenges Anticipated:

- Configuring and tuning Spark and Prometheus services on EC2
- Securing and coordinating service communication via VPC and Security Groups

Next Steps:

- Build CDK stacks for VPC, Kafka, and Spark EC2 instances
- Begin implementing Lambda-based sensor data simulator
- Integrate PostgreSQL and S3 storage, then connect Spark pipeline

Data Hosting: No external datasets will be used. All data will be generated on-the-fly using Lambda and stored in RDS or S3.

Excel Tracker: `project_tracker.xlsx` is included and documents all key deadlines, task responsibilities, and timeline breakdown.