Final Architecture and Explanation

Project: Social Media Sentiment Analysis Platform (POC)

Overview:

This proof of concept implements a real-time sentiment analysis platform using simulated social media data.

It ingests content from multiple platforms (Twitter, Facebook, Instagram, LinkedIn), processes it using PySpark,

and stores it in a structured format for export and analysis. The project is built using a modular, automation-aware

approach, designed to scale in production.

Tech Stack and Tooling:

- Kafka + Zookeeper (via Docker Compose) for real-time message queue
- Python ingestion scripts (one per platform)
- PySpark for streaming and batch processing
- TextBlob for sentiment analysis
- Streamlit for dashboarding
- SQLite and CSV for local export (PostgreSQL ready)
- Retry-enabled CLI and structured logging
- Parquet format used for scalable storage
- .env used for secrets/config management

Key Features:

- Real-time streaming from simulated social media ingestion to Spark
- Deduplication, retry logic, error-safe ingestion
- Modular directory structure and automated runners
- Fully exportable insights via CSV, SQLite, or PostgreSQL
- Visualization with Streamlit and Power BI-ready exports
- .env-based config control

Scaling to AWS Production:

- Kafka AWS MSK (Managed Kafka)
- Spark Streaming AWS Glue Streaming Jobs or EMR on EC2
- Parquet files stored in Amazon S3
- Aggregation Athena or Glue ETL
- Export PostgreSQL on RDS or Redshift for analytics
- API authentication Secrets Manager + API Gateway
- CI/CD GitHub Actions or CodePipeline

Security:

- Secrets separated into .env
- Kafka stream controlled with topic-level design
- No direct social APIs in this POC to avoid leakage

CI-Friendly:

- Retry logic via CLI wrapper
- Logging to file and console
- Scripts runnable independently or chained

Future Work:

- Add real social APIs (Twitter, Meta, LinkedIn official API keys)
- Add real-time anomaly alerts
- Enable delta lake or versioned parquet storage
- Connect to ML scoring engine

Submitted Project Includes:

- All ingestion scripts
- Kafka config
- PySpark processors
- Simulated dataset
- Retry runner
- Streamlit dashboard
- Exporters (CSV, SQL)
- Architecture diagram and README

Status: Ready for submission