Real-Time Social Media Analytics Pipeline

Project Kickoff Report

Alexander Huynh Koehler

June 10, 2025

1 Project Kickoff Questions Team Discussion

1.1 What are the specific goals of this project?

This is primarily a learning project to master real-time data pipeline technologies (Kafka, Spark, Azure) while building a functional social media analytics system. The goal is to progress from basic data collection through machine learning implementation to a complete integrated system, ultimately proving readiness to scale to production-level data volumes like the Bluesky firehose.

1.2 How do we define the project scope clearly to avoid scope creep?

The project is strictly limited to three sequential phases: Phase 1 (data pipeline only), Phase 2 (ML analytics development), and Phase 3 (integration and dashboard). Each phase has explicit exclusions documented upfront, with a feature freeze at 70% completion and focus on MVP functionality before any enhancements.

1.3 What deliverables must be completed at different phases?

Phase 1 delivers a working Bluesky → Kafka → Spark → Azure pipeline processing 50-100 posts/hour. Phase 2 delivers validated sentiment analysis, trend detection, and user behavior models tested on known datasets. Phase 3 delivers an integrated system with dashboard running reliably for 1+ week on live data.

1.4 What are the major milestones, and what deadlines should we set?

Major milestones are: Month 1 - complete data pipeline, Month 2 - analytics models validated, Month 3 - integrated system with dashboard. The 12-week timeline includes buffer time for learning curves, with bi-weekly checkpoints to assess progress and adjust scope if needed.

1.5 Do the team's capabilities align with these goals? Are there any gaps that need to be addressed early on?

As a first-time user of Kafka, Azure, and Bluesky API, there are significant learning gaps that require the extended 12-week timeline with 50% buffer time built in. Early focus should be on fundamental tutorials for each technology and establishing a working development environment before attempting integration.

1.6 Do you have a dataset ready to use for the current project?

No dataset is ready yet - Phase 1 will focus on establishing Bluesky API access and collecting initial data at a manageable rate (10-20 posts per API call). Phase 2 will use established Kaggle social media datasets for ML model development and validation before applying to live Bluesky data in Phase 3.

2 Skills & Tools Assessment

2.1 Are there external resources or team members with expertise in the areas where we lack skills?

Since this is a solo learning project, external resources will be critical - primarily online documentation, tutorials, and community forums like Stack Overflow, Reddit communities for Kafka/Spark, and Microsoft Learn for Azure. The Confluent Kafka tutorials and Databricks community edition will be key learning platforms for hands-on practice.

2.2 Which tools, frameworks, and libraries are most suitable for the project's scope?

For Phase 1: Python with kafka-python library, Apache Spark with PySpark, Azure SDK for Python, and basic JSON handling libraries. For Phase 2: existing NLP expertise with VADER, transformers library for RoBERTa, scikit-learn for clustering, and pandas for data manipulation - leveraging current strengths while integrating new pipeline knowledge.

2.3 How can we ensure that each team member is comfortable with the tools selected?

As a solo project, comfort will come through structured learning - dedicating the first 2 weeks to basic tutorials for each technology, setting up simple "hello world" examples for Kafka and Spark, and maintaining a learning journal to track progress and troubleshooting solutions. The 12-week timeline with 50% buffer accommodates the learning curve.

2.4 Have specific tasks been assigned based on individual strengths, and are team members clear on their roles?

Phase 1 tasks focus on learning data engineering fundamentals (new skills), while Phase 2 leverages existing NLP and ML expertise to implement analytics on the pipeline built in Phase 1. This sequence allows mastering infrastructure first, then applying known techniques to the new system architecture.

3 Initial Setup

3.1 What development environment setup is necessary for this project?

Local development requires Python 3.8+, Docker for running Kafka locally, Java 8+ for Spark, and an Azure free account with storage and SQL database configured. A dedicated project directory with virtual environment isolation will keep dependencies clean and allow for easy troubleshooting.

3.2 Have we successfully configured version control (such as Git)? Does everyone have access to the repository?

Git repository will be initialized from day one with separate branches for each phase (phase1-pipeline, phase2-analytics, phase3-integration) to maintain clear development separation. Private GitHub repository will include comprehensive README documentation and setup instructions for future reference or portfolio presentation.

3.3 Have we installed and configured all required software, libraries, and tools?

Week 1 checklist includes: Python environment with kafka-python and azure-storage-blob libraries, Docker installation with Kafka container running, basic Spark installation, and successful API calls to Bluesky endpoints. Each component will be tested individually before attempting integration.

3.4 What testing can we run to ensure that the development environment is functioning correctly?

Simple validation tests include: Kafka producer/consumer sending test messages, Spark job processing sample JSON data, Azure blob storage upload/download operations, and Bluesky API returning valid post data. Each test should run successfully and be documented for troubleshooting reference.

3.5 What troubleshooting steps should we take if the setup does not work as expected?

Maintain a troubleshooting log with common issues and solutions, start with official documentation before community forums, test each component in isolation before integration, and keep backup installation methods (different Python versions, alternative Kafka setups). If major blockers occur, pivot to using cloud services (Azure HDInsight for Spark) instead of local installation.

4 Plan Revision

Since this is the initial planning phase, no revisions are needed yet, but the 12-week timeline with 50% buffer time built in anticipates learning curve challenges. Progress will be tracked through bi-weekly self-assessments and documented in a learning journal, with flexibility to adjust scope (reduce features) rather than extend deadlines if Phase 1 data engineering proves more challenging than expected. The modular three-phase approach allows for early pivots - if Kafka/Spark integration becomes problematic, the project can shift to simpler batch processing while still demonstrating core analytics capabilities.