Canalytics Project Plan

1. Project Overview

Objective: Develop a streamlined analytics pipeline for canals traffic data, delivering production-ready code, a concise analytical report, and a presentation slide deck.

Defense Period: June 11–18, 2025

---

2. Team & Responsibilities

- **Timur – Team Lead, Pipeline & Orchestration Engineer**
  - Overview of the domain.
  - Design project architecture.
  - Containerize scripts using Docker.
  - Design task orchestration via Apache Airflow within `docker-compose`.

- **Umar – Storage & Infrastructure Engineer**
  - Set up AWS S3 (for raw data) and ClickHouse (for structured, processed data).
  - Maintain access policies, availability, and backup schedules.
  - Provide and document a utility module for database ingestion.
  - Add logging, failure alerts, and task retry logic for reliability.

- **Dimitry – Lead Data Collector**
  - Develop Python scripts (using `requests`, `websockets`) to collect AIS json feeds and scrape news headlines.
  - Automate regular data pulls and store raw files in the shared S3 bucket.
  - Ensure robustness and coverage of data collection over time.

- **Maria – Data Analyst & Visualization Lead**
  - Design and implement ETL.
  - Perform core analysis, including:
    - Time-series analysis of freight traffic.
    - Basic geospatial mapping of congestion.
  - Lead drafting of the final report and preparing the presentation.

---

3. Simplified Tech Stack

Component	Tools
Collection	Python (`requests`, websockets)
Orchestration	Docker + Apache Airflow (`docker-compose`)
Storage	AWS S3 (raw), ClickHouse (processed), SQLite (logs)
Processing	PySpark (preferred) or pandas
Visualization	matplotlib, Jupyter, Reveal.js

---

4. Deliverables & Timeline

Sprint	Deliverable	Due Date
Sprint 1	Data collectors + Docker setup	May 31, 2025
Sprint 2	Airflow DAGs + Data stored in S3/RDS	Jun 7, 2025
Sprint 3	ETL scripts + analysis notebooks	Jun 12, 2025
Sprint 4	Final report + presentation slides	Jun 15, 2025
Defense Week	Live defense	Jun 11–18
Repo	Code, docs, and updates	Ongoing

---

5. Parallel Kickoff Plan

Each team member can begin work independently from May 28. Branch naming and sync points ensure clean integration.

- **Dimitry**
    1. Clone the GitHub repository.
    2. Start developing `collectors/ais_collector.py` and `news_collector.py`.

3. Push changes under branch: `feature/dimitry-collectors`.

- **Timur**
  1. Create `Dockerfile` inside the `pipeline/` folder.
  2. Build a basic `docker-compose.yml` to run data collectors.
  3. Push changes under branch: `feature/timur-docker`.

- **Umar**
  1. Provision AWS S3 and RDS (PostgreSQL).
  2. Create `storage/db_loader.py` with stub loaders for both S3 and DB.
  3. Push changes under branch: `feature/umar-storage`.

- **Maria**
  1. Set up Jupyter and Spark (or pandas).
  2. Build skeleton `analysis/etl.py` and create a notebook in `analysis/notebooks/`.
  3. Push changes under branch: `feature/maria-analysis`.

Sync Points:

- **May 30:** All feature branches must be ready for review via PRs.
- **May 31:** Merge reviewed PRs into `develop` for integration testing.

---

6. GitHub Repository Structure

https://github.com/ITMO-Canalytics/canalytics

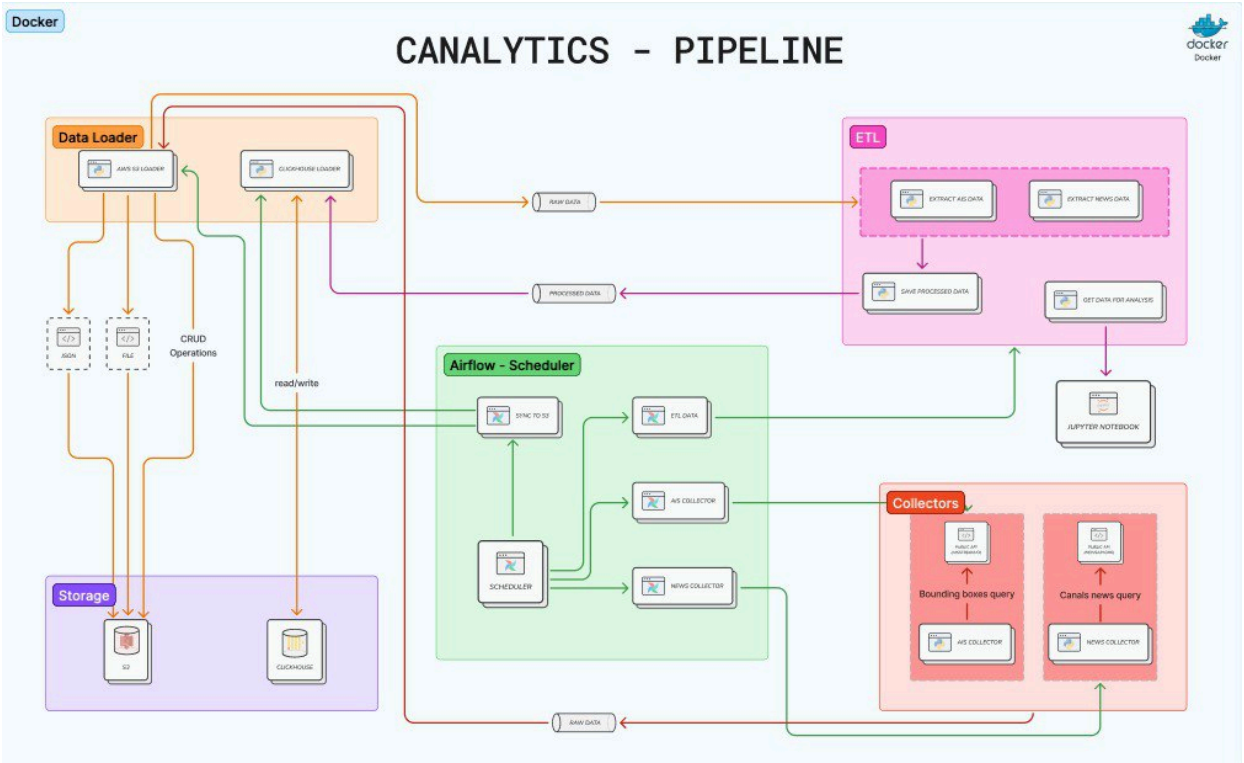
```
canalytics/
├── data/                  # Raw and processed data storage
│   ├── raw/               # Raw CSV and JSON from collectors
│   └── processed/         # Cleaned datasets
├── collectors/            # Data collection scripts
│   ├── ais_collector.py   # AIS data fetcher
│   └── news_collector.py  # News headlines scraper
├── pipeline/              # Orchestration & ingestion
│   ├── Dockerfile         # Container for collectors
│   └── airflow/           # Airflow DAGs and configs
│       └── dags/
│           └── dag_collect_ais.py
```

```
│           └── dag_collect_news.py
│           └── dag_etl.py
│           └── dag_sync.py
├── storage/                  # Storage utilities
│   ├── s3_loader.py          # S3 storage operations
│   ├── clickhouse_loader.py  # ClickHouse database operations
│   ├── db_loader.py          # Combined loader utilities
│   └── __init__.py           # Package initialization
├── analysis/                 # Data preparation and analysis
│   ├── etl.py                # ETL pipeline scripts
│   └── notebooks/            # Jupyter notebooks with exploratory analysis
│       └── suez_canal_analysis.ipynb
├── report/                   # Report and slides
│   └── report.md             # Markdown source for report
├── docker-compose.yml        # Local development orchestration
├── requirements.txt          # Python dependencies
└── README.md                 # Project overview and setup instructions
```

---

7. Project pipeline

https://www.figma.com/board/DK79bg3lldmBIVLVsIcCUF/CANALYTICS---PIPELINE?node-id=0-1&t=UwFfVoICSTn00HSf-1



8. Storage architecture