Coal Mining Data Pipeline

1. Objective
To streamline coal mining operations by collecting and processing data from production logs, IoT sensors, and weather services. This system generates insights that help with production planning, equipment monitoring, and operational efficiency.

2. System Overview
The pipeline extracts raw data, transforms it into usable formats, calculates relevant KPIs, stores it in an analytical database, and visualizes it through dashboards.

3. Components Breakdown

3.1 Data Sources
- **Production Database (PostgreSQL)**
  - Table: `production_logs`
  - Fields: `date`, `mine_id`, `shift`, `tons_extracted`, `quality_grade`
- **Equipment Sensors (CSV File)**
  - File: `equipment_sensors.csv`
  - Fields: `timestamp`, `equipment_id`, `status`, `fuel_consumption`, `maintenance_alert`
- **Weather Data (Open-Meteo API)**
  - Coordinates: 2.0167°N, 117.3000°E
  - Fields: Daily temperature, precipitation
  - Notes: Supports last 92 days; older dates use generated weather patterns

3.2 ETL Pipeline (Python)
- **Extraction**
  - Queries data from production SQL
  - Reads CSV files with timestamp filters
  - Fetches weather data using REST API
- **Transformation**
  - Cleans and validates raw records
  - Aggregates daily KPIs
  - Joins data by date and mine ID
  - Applies anomaly detection (e.g., zero output on working days)
- **Loading**
  - Writes processed metrics into ClickHouse
  - Uses bulk inserts and partitioning by `date`
- **Error Handling**
  - Logs issues into `/etl/logs`
  - Retries API calls with exponential backoff
  - Skips corrupt CSV lines while logging them

3.3 Metrics Generated

Metric	Description
`total_production_daily`	Sum of tons extracted per day
`average_quality_grade`	Weighted average of quality grade by tons
`equipment_utilization`	Time percentage machines are marked as active
`fuel_efficiency`	Liters of fuel per ton mined
`weather_impact`	Correlation of rain/temp with production

3.4 Database – ClickHouse

ClickHouse is used to store all processed metrics. It is designed for reading large sets of structured data efficiently. Tables are set up to match daily records, with fields that allow filters by mine, shift, and time. Partitions by date allow for fast access to specific time periods.

3.5 Visualization – Metabase

- Visualizes data directly from ClickHouse
- Dashboards include:
    - Daily production trend (line chart)
    - Mine performance comparison (bar chart)
    - Weather correlation (scatter plot)
- Hosted at: `http://localhost:3000`

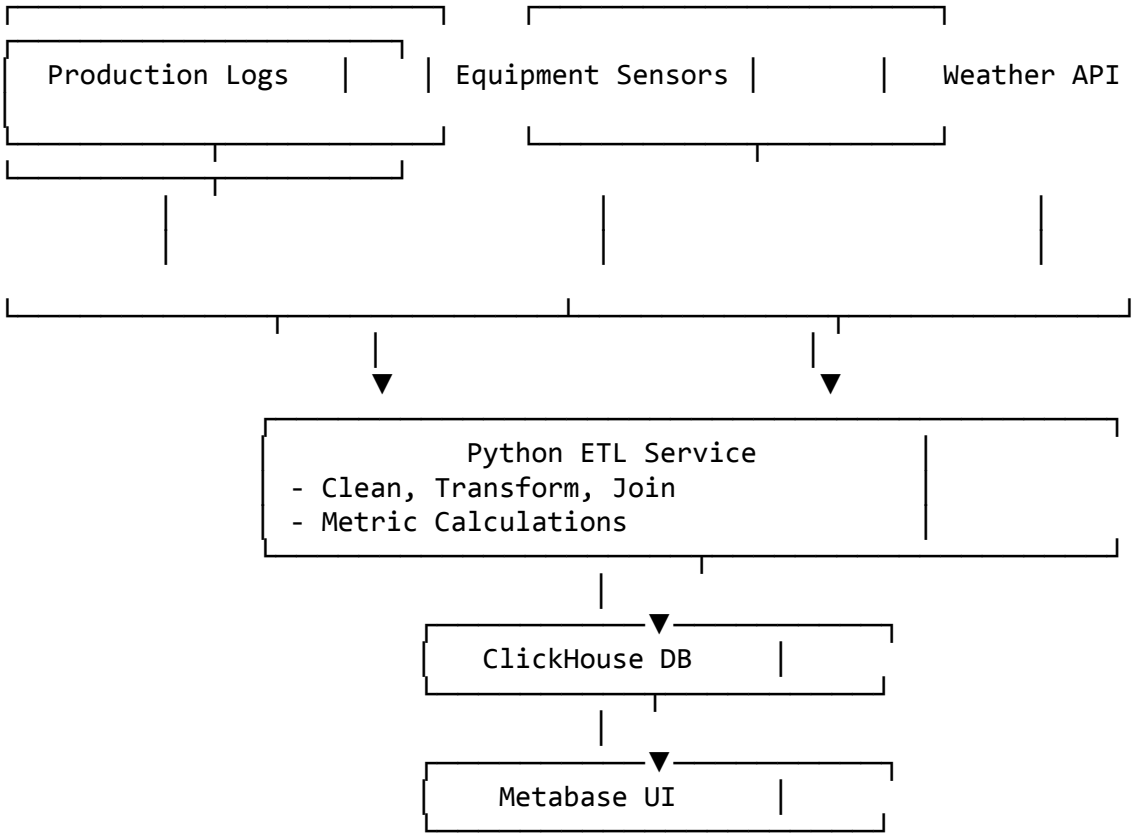
3.6 Docker Compose

Runs the entire stack:

```
services:
  clickhouse:
    image: clickhouse/clickhouse-server

  metabase:
    image: metabase/metabase
    ports:
      - "3000:3000"

  postgres:
    image: postgres

  coal-mining-etl:
    build: ./etl
    volumes:
      - ./etl/logs:/app/logs
```

```
    depends_on:
      - clickhouse
```

4. Workflow Diagram



5. Performance Notes

ClickHouse is set up with partitions and LZ4 compression, which helps speed up how data is stored and read. The ETL script uses bulk inserts when writing to the database. This avoids the slow process of writing each row one at a time. To reduce unnecessary load, the ETL also caches weather data so that repeat requests for the same date don't need to call the external API again.

6. Limitations

The Open-Meteo API only offers weather history going back 92 days. If older data is needed, the ETL generates estimated values based on past trends. The equipment sensor logs are stored as flat CSV files and must be updated frequently. This can be done manually or with a scheduled cron job. Finally, the whole system runs in batch mode. It doesn't currently support streaming or live updates.

7. Setup & Run

```
git clone <repo>
cd metabase-clickhouse
docker-compose up -d
```

- Visit `http://localhost:3000` for dashboards
- Run ETL manually: `cd etl && python coal_mining_etl.py`