Lichess Streaming + Batch Data Pipeline

Role	Name	Student ID
Member 1	Smambayev Zhusup	22B030443
Member 2	Em Igor	22B030612
Member 3	Kenesbek Asylmurat	22B030376

Objective and requirements coverage

DAG 1: Continuous ingestion from Lichess TV feed to Kafka.DAG 2: Hourly batch reads Kafka, cleans events, stores into SQLite.DAG 3: Daily analytics reads SQLite and writes to a separate summary table.

Figure 1 shows all three DAGs registered in Airflow.

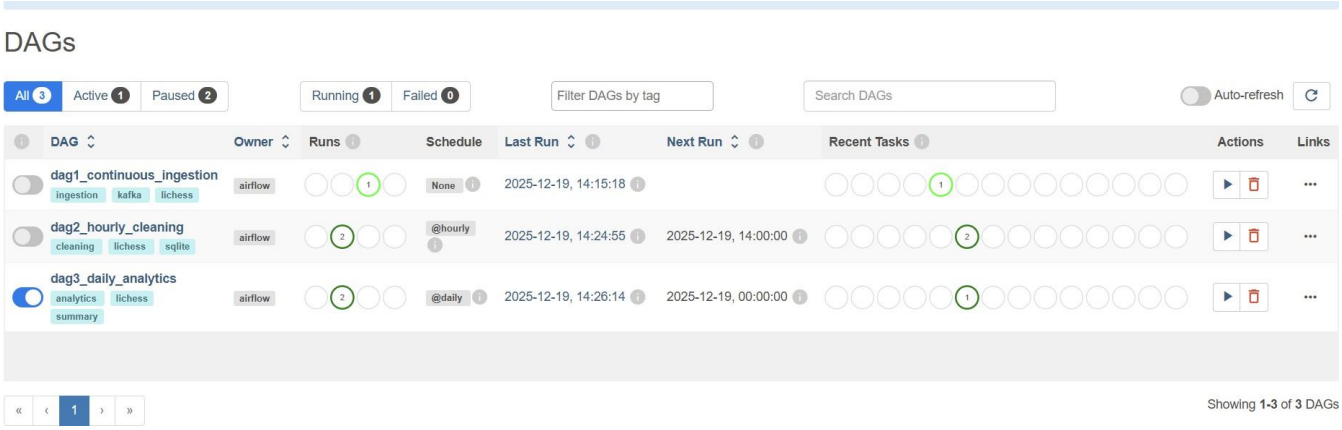

Figure 1. Airflow UI showing dag1, dag2, and dag3.

DAG 1: Continuous Ingestion (from API to Kafka)

DAG: dag1_continuous_ingestion (schedule: **None**, manual trigger). **Retries:** 3 (retry delay: 5 minutes). Reads NDJSON stream from Lichess TV feed (**LICHESS_API_BASE_URL** + **LICHESS_TV_FEED_ENDPOINT**). Sends each parsed JSON event to Kafka via *send_to_kafka(..., topic=KAFKA_TOPIC_RAW_EVENTS, key=game_id)*.

Kafka topic schema

Topic: *KAFKA_TOPIC_RAW_EVENTS* (configured in *config/config.py*). **Key:** *game_id* (from event['d']['id']). **Value:** raw JSON event from Lichess TV feed: event_type + game payload (players, ratings, fen, clocks, last_move).

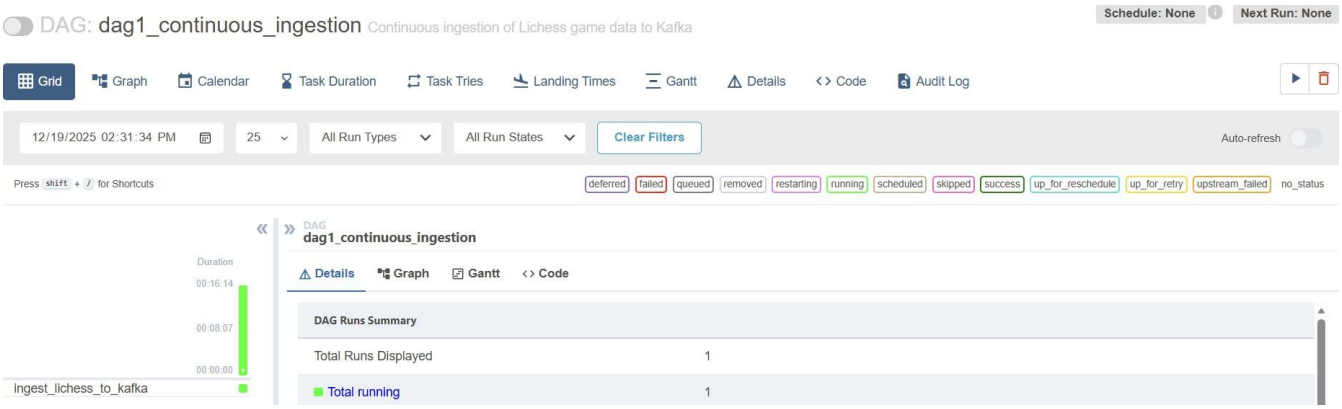

Figure 2. DAG 1 grid view.

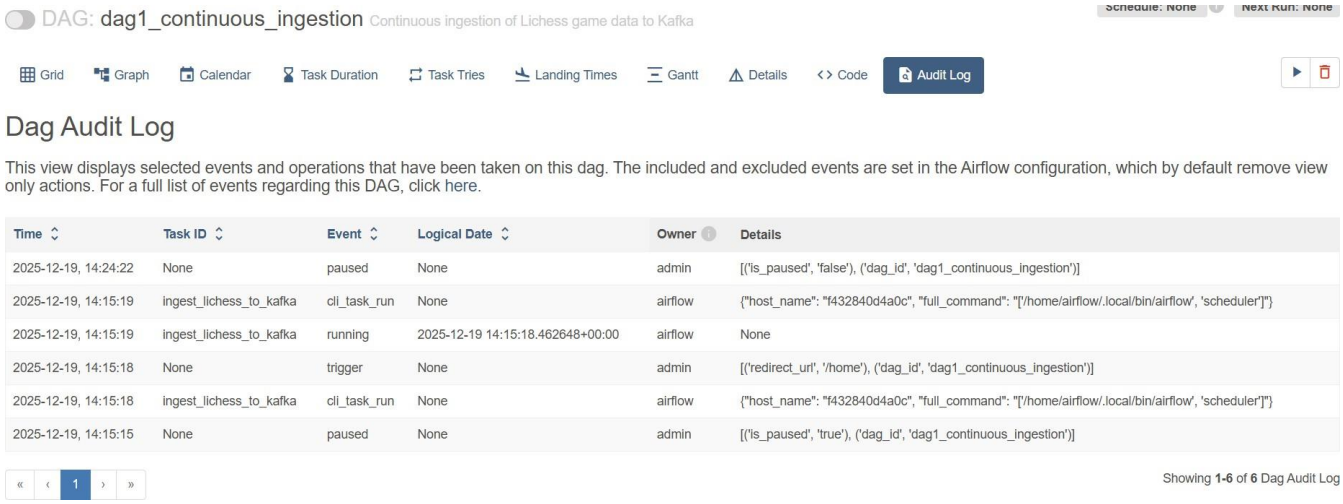

Figure 3. DAG 1 audit log (trigger + run).

DAG 2: Hourly Cleaning and Storage

DAG: dag2_hourly_cleaning (schedule: **@hourly**).**Retries:** 2 (retry delay: 5 minutes).Consumes Kafka topic **KAFKA_TOPIC_RAW_EVENTS** using **KAFKA_CONSUMER_GROUP_ID**.Cleans/normalizes events with **clean_game_event** and inserts into SQLite table **events**.Offsets are committed after DB writes to reduce duplicate processing.

Cleaning rules

Extract key fields (game_id, event_type, players, ratings, clocks, fen, last_move).Type casting for numeric fields and safe defaults for missing values.Add timestamps: *ingested_at* and *processed_at*.Deduplicate in SQLite using UNIQUE(game_id, ingested_at).

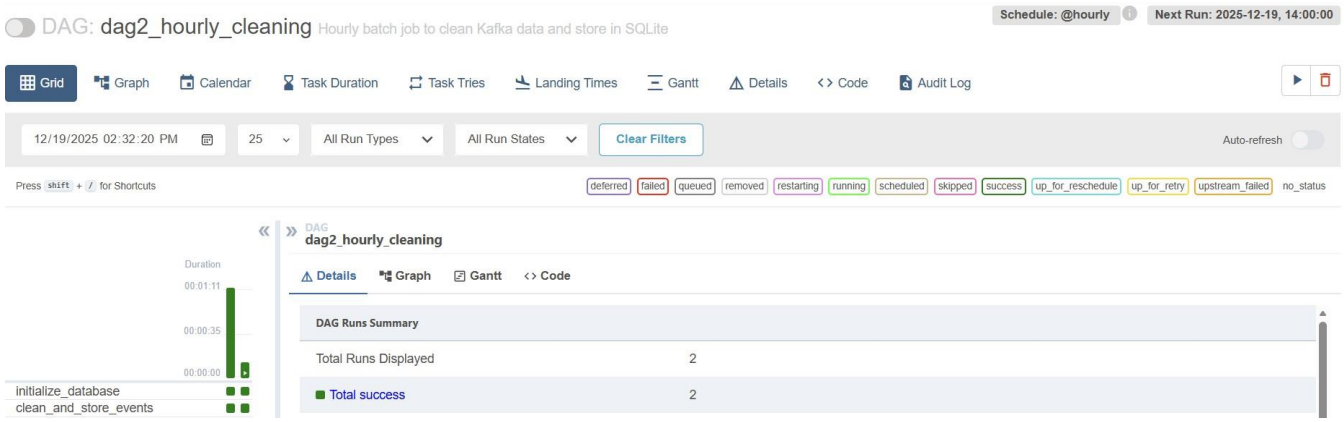

Figure 4. DAG 2 grid view (hourly batch run).

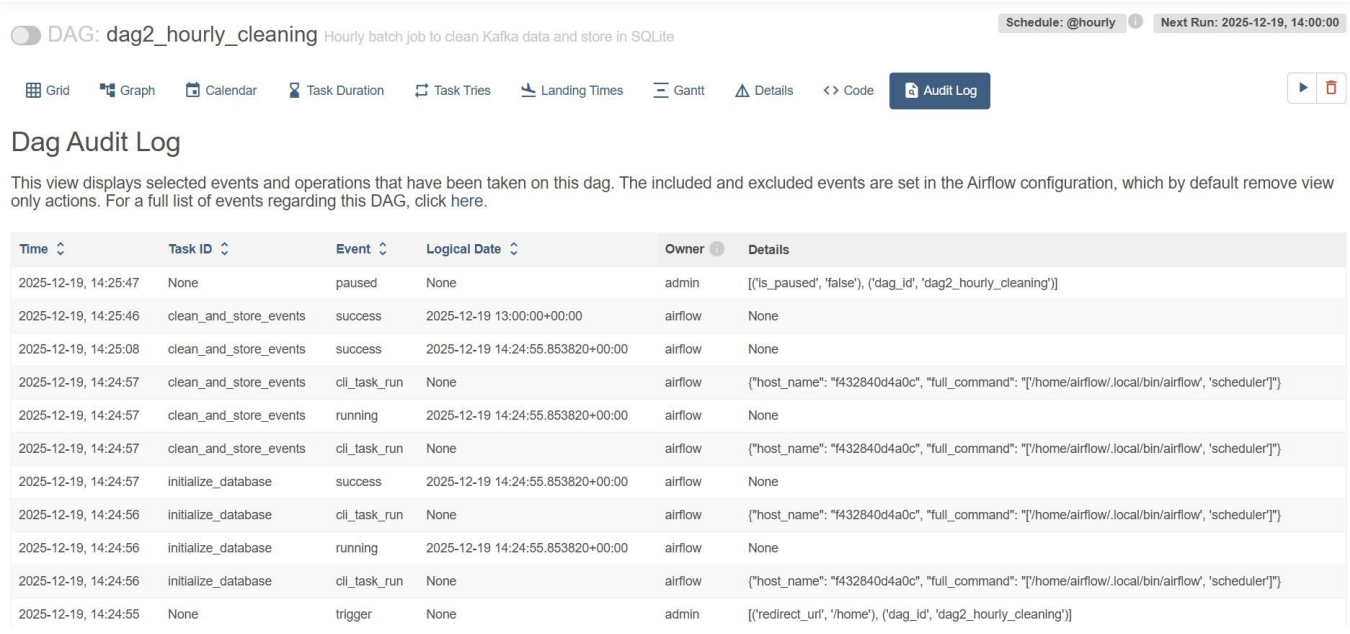

Figure 5. DAG 2 audit log (successful tasks).

DAG 3: Daily Analytics

DAG: dag3_daily_analytics (schedule: **@daily**). **Retries:** 2 (retry delay: 5 minutes). Computes: total_games, avg ratings, min/max, duration estimate, titled players, rating buckets, most_common_opening (proxy). Writes one row per date into SQLite table **daily_summary** (unique summary_date).

SQLite schema

```
-- Table: events
CREATE TABLE IF NOT EXISTS events (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    game_id TEXT NOT NULL,
    event_type TEXT NOT NULL,
    orientation TEXT,
    white_player TEXT NOT NULL,
    white_rating INTEGER,
    white_title TEXT,
    white_seconds INTEGER,
    black_player TEXT NOT NULL,
    black_rating INTEGER,
    black_title TEXT,
    black_seconds INTEGER,
    fen TEXT,
    last_move TEXT,
    white_clock INTEGER,
    black_clock INTEGER,
    ingested_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    processed_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    UNIQUE (game_id, ingested_at)
);

CREATE INDEX IF NOT EXISTS idx_events_game_id
    ON events (game_id);

CREATE INDEX IF NOT EXISTS idx_events_processed_at
    ON events (processed_at);


-- Table: daily_summary
CREATE TABLE IF NOT EXISTS daily_summary (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    summary_date DATE NOT NULL UNIQUE,
    total_games INTEGER,
    avg_white_rating REAL,
    avg_black_rating REAL,
    min_rating INTEGER,
    max_rating INTEGER,
    avg_game_duration_seconds REAL,
    total_titled_players INTEGER,
    rating_distribution_1000_1500 INTEGER,
    rating_distribution_1500_2000 INTEGER,
    rating_distribution_2000_2500 INTEGER,
    rating_distribution_2500_plus INTEGER,
    most_common_opening TEXT,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

Database verification

Snapshot from **lichess.db**: **19** events and **1** daily summary record(s). Latest daily_summary: summary_date=2025-12-19, total_games=19, avg_white=2901.74, avg_black=2890.95.

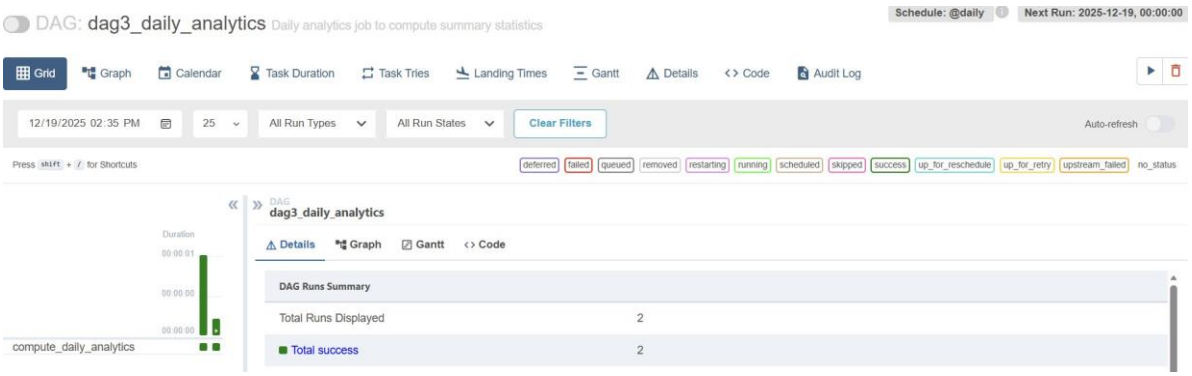

Figure 6. DAG 3 grid view (daily analytics).

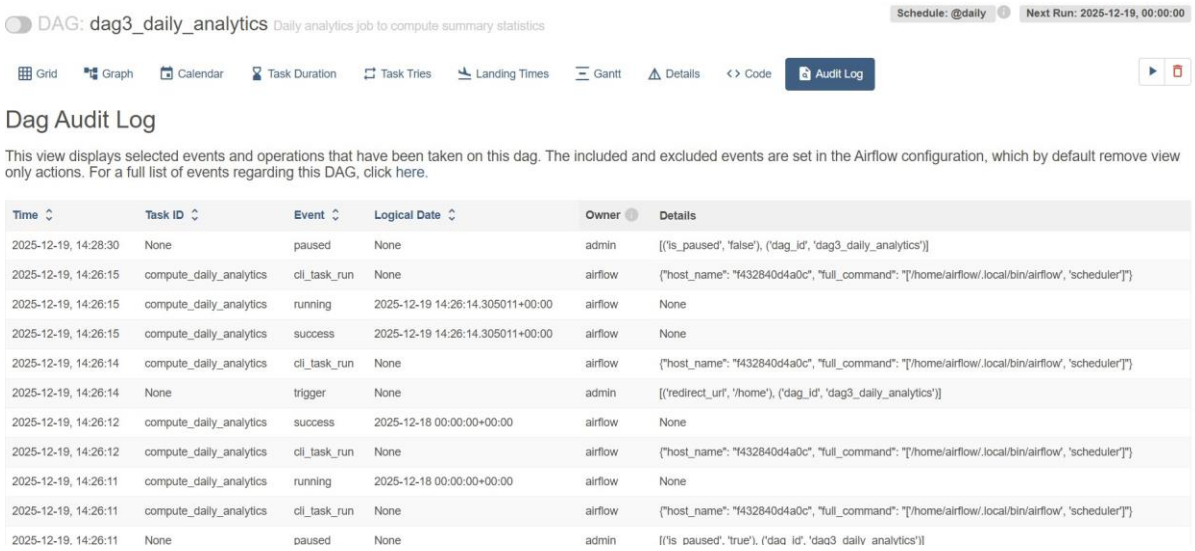

Figure 7. DAG 3 audit log (successful run).

```
PS D:\System Folders\Desktop\New folder (5)> docker exec -it airflow-webserver sqlite3 /opt/airflow/data/lichess.db
SQLite version 3.40.1 2022-12-28 14:03:47
Enter ".help" for usage hints.
sqlite> SELECT COUNT(*) FROM events;
19
sqlite> SELECT * FROM events LIMIT 3;
1|ZN01438a|featured|white|trojanhorse2500|3033|IM|3|cyberspace_xd|2759||3|rnbqnrk1/pp3pbp/3p2p1/2pP4/2P1p3/1P2P3/P1N1BPPP/2BQNRK1 w - - 4 12||0|0|2025-12-19T14:24:38.067389|2025-12-19T14:25:46.780094
2|Mxr23Mkm|featured|black|cyberspace_xd|2758||3|trojanhorse2500|3035|IM|3|rnbqkbnr/pppppppp/8/8/8/8/PPPPPPPP/RNBQKBNR w KQkq - 0 1||0|0|2025-12-19T14:24:38.103031|2025-12-19T14:25:46.796906
3|i8dsDmfo|featured|white|trojanhorse2500|3026|IM|3|cyberspace_xd|2767||3|rnbqkbnr/pppppppp/8/8/8/5N2/PPPPPPPP/RNBQKB1R b KQkq - 1 1||0|0|2025-12-19T14:24:38.181763|2025-12-19T14:25:46.796954
sqlite> SELECT * FROM daily_summary;
1|2025-12-19|19|2901.74|2890.95|2758|3035|0.0|19|0|0|0|38||2025-12-19T14:26:15.661207
sqlite>
```

Figure 8. SQLite queries: COUNT(*) and sample rows.