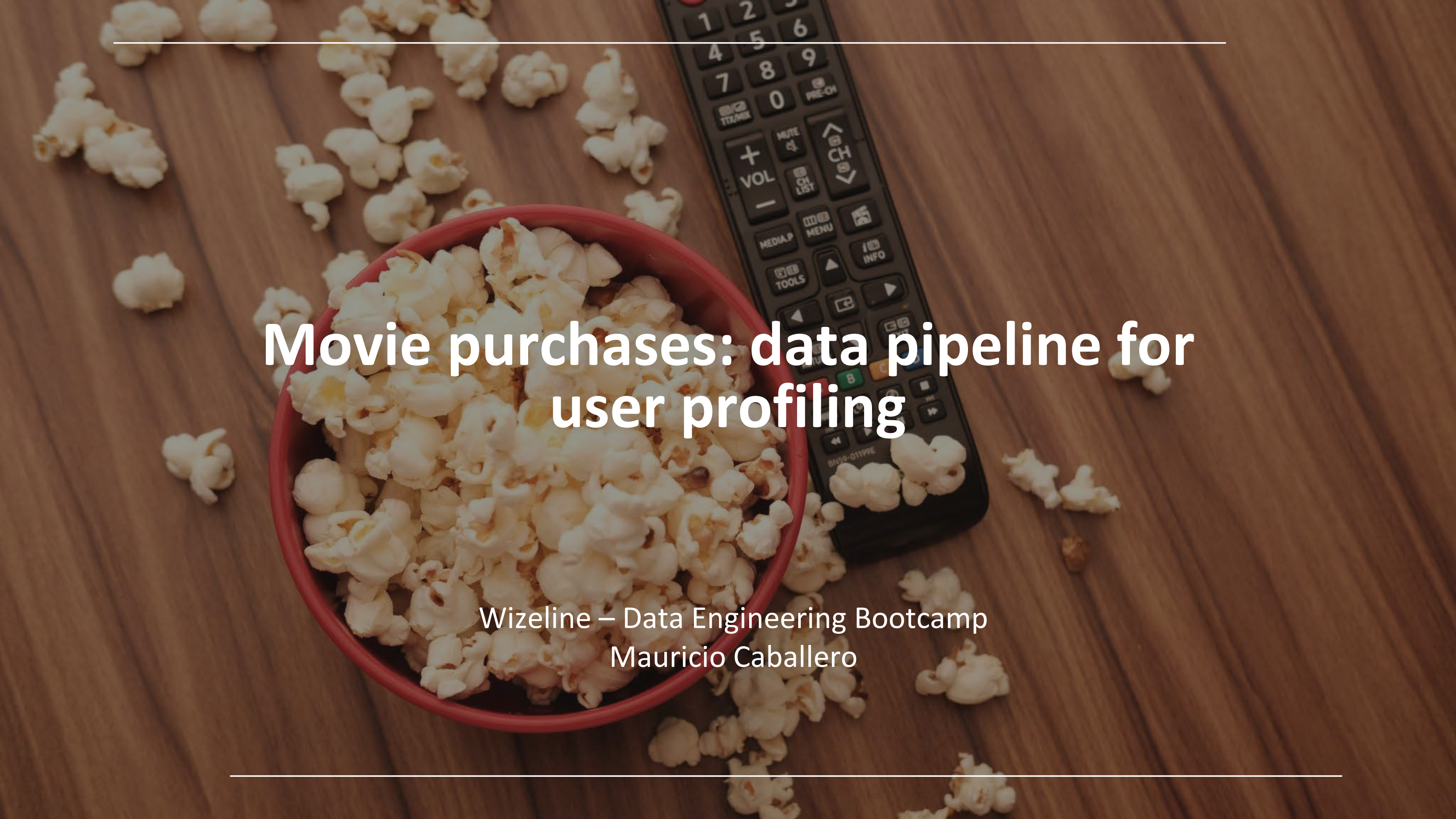


Movie purchases: data pipeline for user profiling

Wizeline – Data Engineering Bootcamp

Mauricio Caballero

Problem Statement

As part of a user behavior analytics firm, create a data pipeline that allows analysts to examine customers based on their movie purchases and reviews.

Input:
- Movie purchases records
- Movie reviews records

Output:
- User behavior metric table for analysts/dashboards

Input data – Sneak peek

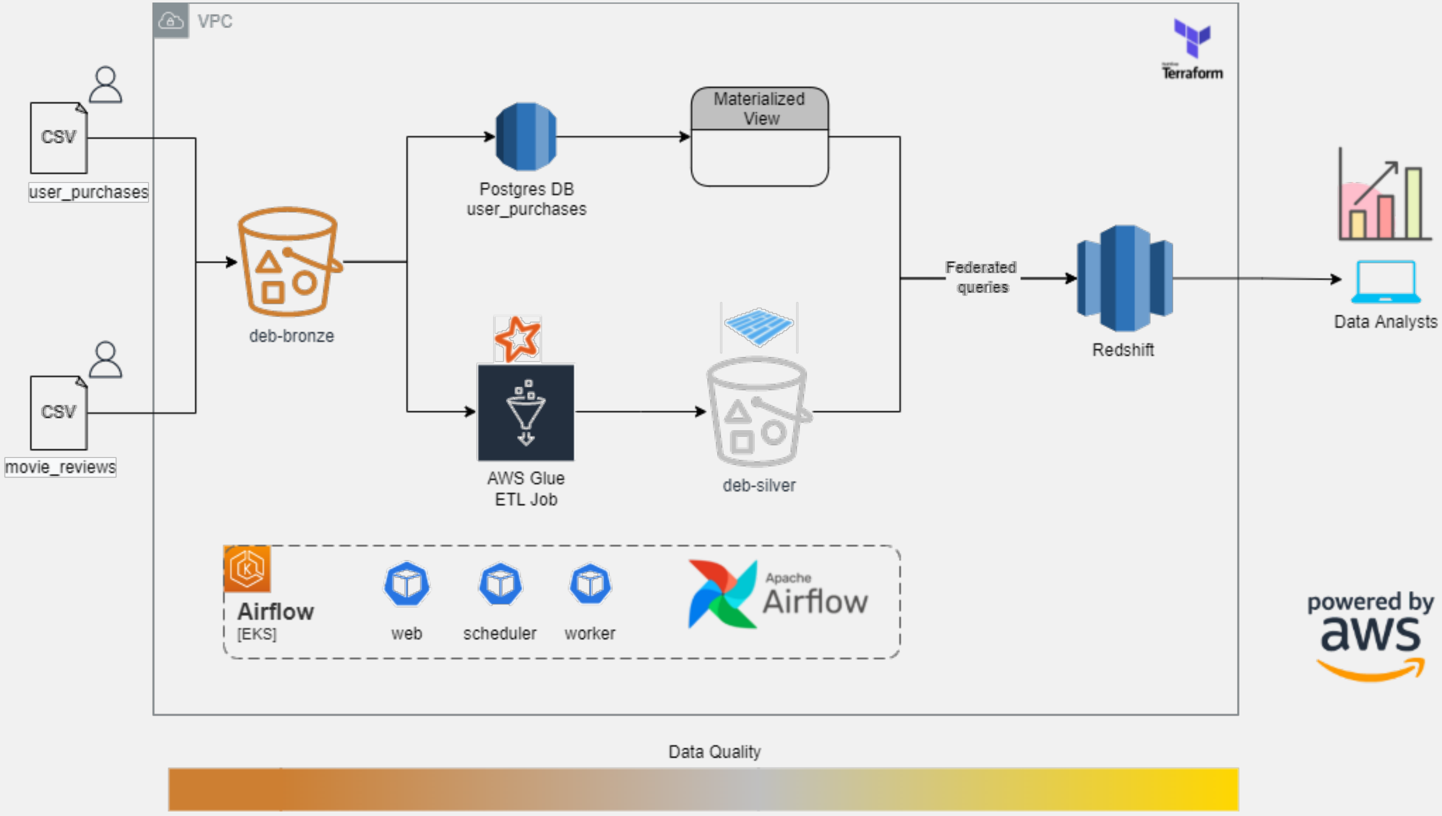
Movie purchases
CSV file with ~542k rows, 44 MBs

purchaseid	invoiceno	stockcode	description	quantity	invoicedate	unitprice	customerid	country
1	536365	85123A	WHITE HANGI…	6	2010-12-01 08:26:00	2.55	17850	United Kin…
3	536365	84406B	CREAM CUPID…	8	2010-12-01 08:26:00	2.75	17850	United Kin…
5	536365	84029E	RED WOOLLY …	6	2010-12-01 08:26:00	3.39	17850	United Kin…
7	536365	21730	GLASS STAR F…	6	2010-12-01 08:26:00	4.25	17850	United Kin…
9	536366	22632	HAND WARME…	6	2010-12-01 08:28:00	1.85	17850	United Kin…

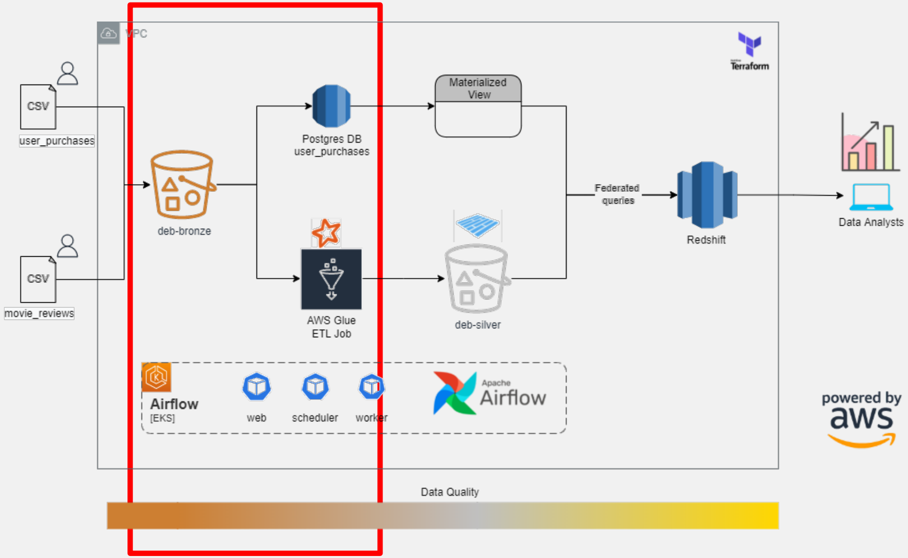
Movie reviews
CSV file with ~100k rows, 129 MBs

cid	review_str
13756	Once again Mr. Costner has dragged out a movie for far longer than necessary.
15738	This is an example of why the majority of action films are the same. Generic ar
15727	First of all I hate those moronic rappers, who could'nt act if they had a gun pre:
17954	Not even the Beatles could write songs everyone liked, and although Walter Hi

Implemented architecture

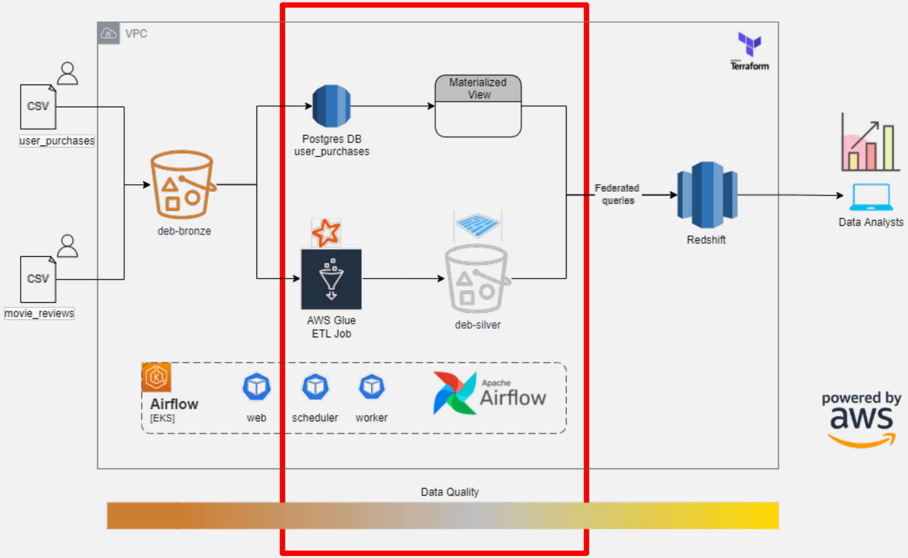


Raw Layer



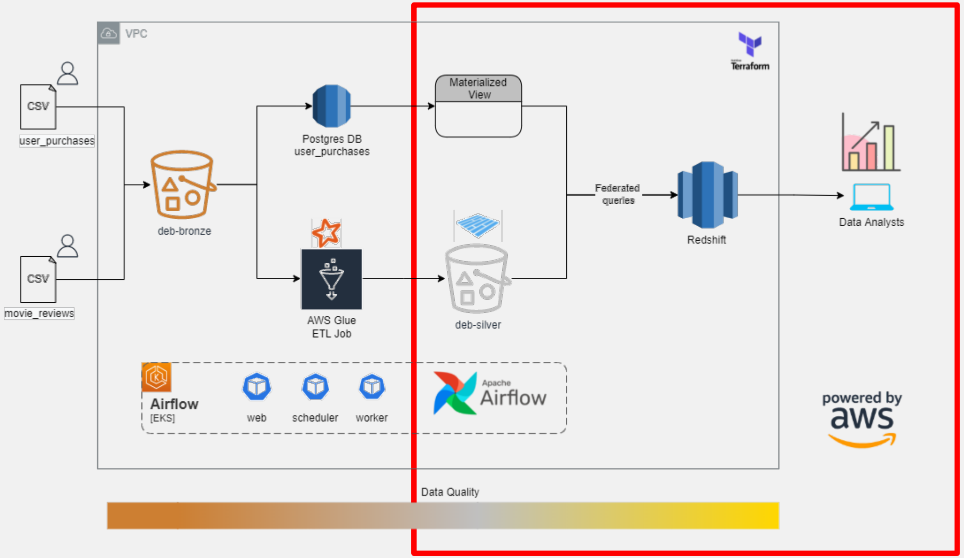
- DAG to upload raw purchases to a Postgres DB
- Reviews stay in the S3 bucket

Staging Layer



- Materialized view created to provide cleaned purchases data
- Glue ETL job that runs on spark to classify reviews as positive or negative, and writes parquet files

Production Layer



- Federated queries used by Redshift
- User behavior metric table created, rows inserted

Result – User Behavior Metric table

Amazon Redshift (DW) populates the user_behavior_metric table
889 rows, generated in 3.2s

customerid	amount_spent	review_score	review_count	insert_date
13047	3237.54	62	156	2021-12-09
14688	5630.87	58	172	2021-12-09
12431	6487.45	45	159	2021-12-09
13767	17220.36	60	167	2021-12-09
12791	192.6	51	161	2021-12-09
14307	2995.72	42	165	2021-12-09
12838	683.13	64	188	2021-12-09
18085	689.95	47	177	2021-12-09
15983	1475.02	47	145	2021-12-09
12868	1607.06	62	171	2021-12-09

Lessons Learned

Future work

- Use more advanced NLP models for the sentiment analysis classifier
- Evaluate and create dashboards (Amazon QuickSight, Tableau)
- Introduce data cleaning as an ETL step
- Generate aggregated tables as ETL for DW (Redshift) consumption
- Evaluate Amazon Kinesis or Kafka for data streaming

THANKS!

Do you have any questions?