

Course Project

Objective

The goal of this project is to apply everything we have learned in this course to build an end-to-end data pipeline.

Problem statement

Develop a dashboard with two tiles by:

- Selecting a dataset of interest (see [Datasets](#))
- Creating a pipeline for processing this dataset and putting it to a datalake
- Creating a pipeline for moving the data from the lake to a data warehouse
- Transforming the data in the data warehouse: prepare it for the dashboard
- Building a dashboard to visualize the data

Data Pipeline

The pipeline could be stream or batch: this is the first thing you'll need to decide

- Stream: If you want to consume data in real-time and put them to data lake
- Batch: If you want to run things periodically (e.g. hourly/daily)

Technologies

You don't have to limit yourself to technologies covered in the course. You can use alternatives as well:

- Cloud: AWS, GCP, Azure, ...
- Infrastructure as code (IaC): Terraform, Pulumi, Cloud Formation, ...
- Workflow orchestration: Airflow, Prefect, Luigi, ...
- Data Warehouse: BigQuery, Snowflake, Redshift, ...
- Batch processing: Spark, Flink, AWS Batch, ...
- Stream processing: Kafka, Pulsar, Kinesis, ...

If you use a tool that wasn't covered in the course, be sure to explain what that tool does.

Dashboard

You can use any of the tools shown in the course (Data Studio or Metabase) or any other BI tool of your choice to build a dashboard. If you do use another tool, please specify and make sure that the dashboard is somehow accessible to your peers.

Your dashboard should contain at least two tiles, we suggest you include:

- 1 graph that shows the distribution of some categorical data
- 1 graph that shows the distribution of the data across a temporal line

Ensure that your graph is easy to understand by adding references and titles.

Evaluation Criteria

- Problem description
 - 0 points: Problem is not described
 - 1 point: Problem is described but shortly or not clearly
 - 2 points: Problem is well described and it's clear what the problem the project solves
- Cloud
 - 0 points: Cloud is not used, things run only locally
 - 2 points: The project is developed in the cloud
 - 4 points: The project is developed in the cloud and IaC tools are used
- Data ingestion (choose either batch or stream)
 - Batch / Workflow orchestration
 - 0 points: No workflow orchestration
 - 2 points: Partial workflow orchestration: some steps are orchestrated, some run manually
 - 4 points: End-to-end pipeline: multiple steps in the DAG, uploading data to data lake
 - Stream
 - 0 points: No streaming system (like Kafka, Pulsar, etc)
 - 2 points: A simple pipeline with one consumer and one producer
 - 4 points: Using consumer/producers and streaming technologies (like Kafka streaming, Spark streaming, Flink, etc)
- Data warehouse
 - 0 points: No DWH is used

- 2 points: Tables are created in DWH, but not optimized
- 4 points: Tables are partitioned and clustered in a way that makes sense for the upstream queries (with explanation)
- Transformations (dbt, spark, etc)
 - 0 points: No transformations
 - 2 points: Simple SQL transformation (no dbt or similar tools)
 - 4 points: Transformations are defined with dbt, Spark or similar technologies
- Dashboard
 - 0 points: No dashboard
 - 2 points: A dashboard with 1 tile
 - 4 points: A dashboard with 2 tiles
- Reproducibility
 - 0 points: No instructions how to run the code at all
 - 2 points: Some instructions are there, but they are not complete
 - 4 points: Instructions are clear, it's easy to run the code, and the code works

Overview

This project is presented as part of the Data Engineering Zoomcamp by DataTalks.Club. It exclusively utilizes data sourced from <https://insideairbnb.com/get-the-data/>, focusing solely on Airbnb listings from Tokyo, Japan.

Problem Statement:

This project aims to analyze Airbnb listings in Tokyo, Japan, focusing on the distribution of neighborhoods, room types, and trends in average price fluctuations over the years. By understanding these aspects, we aim to provide valuable insights for hosts to optimize their offerings and pricing strategies, and for guests to make informed decisions when planning their stays in Tokyo.

Data Pipeline

The pipeline for this project will operate in batch mode, designed to run periodically on a quarterly basis

Technologies

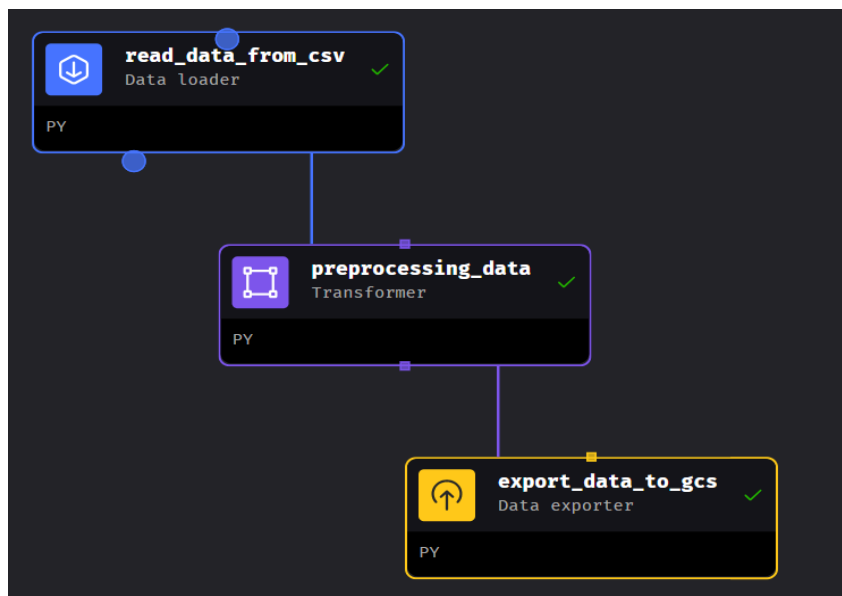
You don't have to limit yourself to technologies covered in the course. You can use alternatives as well:

- Cloud: GCP
- Infrastructure as code (IaC): Terraform
- Workflow orchestration: Mage
- Data Warehouse: BigQuery
- Batch Processing: DBT Cloud
- Data Visualisation: Google Looker

Selecting a dataset of interest

Creating a pipeline for processing this dataset and putting it to a datalake


Workflow Orchestration:



The process includes using Mage to extract, transform, and load data from an API to Google Cloud Storage (GCS).

1. Load data locally from the dataset, specify the data types and last_review as parse_date.

2. Preprocess and clean the data, remove duplicate and blank rows and columns as well as price with blank values. Format last_review column from datetime to date.
3. Partition the data using last_review date.
4. Export partitioned data parquet file to GCS bucket.
5. The pipeline is set to be executed once a month on the 1st of every month.

| Active | Type | Logs | Name | Description | Frequency | Next run date |
|-------------------------------------|----------|---|---------------|--|-----------|---------------------|
| <input checked="" type="checkbox"/> | schedule |  | airbnb_to_gcs | The pipeline will be executed on 1st of every month to export Airbnb data to GCS bucket. | @monthly | 2024-05-01 00:00:00 |

Creating a pipeline for moving the data from the lake to a data warehouse

Create a dataset on BigQuery.

1. Load raw data to Bigquery from Google Cloud Storage Bucket.
2. Dataset name: example_de_zoomcamp_dataset
3. Run query below to build a table with all available data.
4. Table name: airbnb_tokyo

```

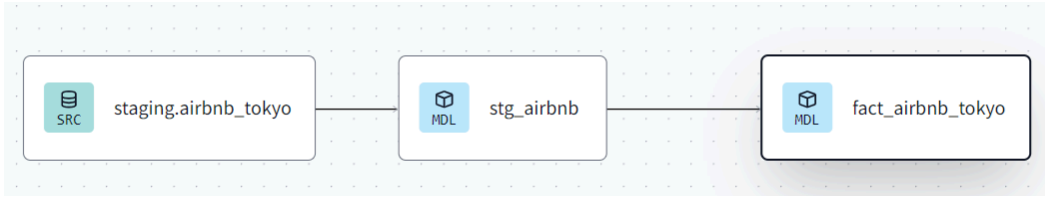
1 CREATE OR REPLACE EXTERNAL TABLE `example-de-zoomcamp.
  example_de_zoomcamp_dataset.airbnb_tokyo`
2 OPTIONS (
3   format = 'PARQUET',
4   uris = ['gs://example_de_zoomcamp_bucket/airbnb_data/
  review_date=*.parquet']
5 );|

```

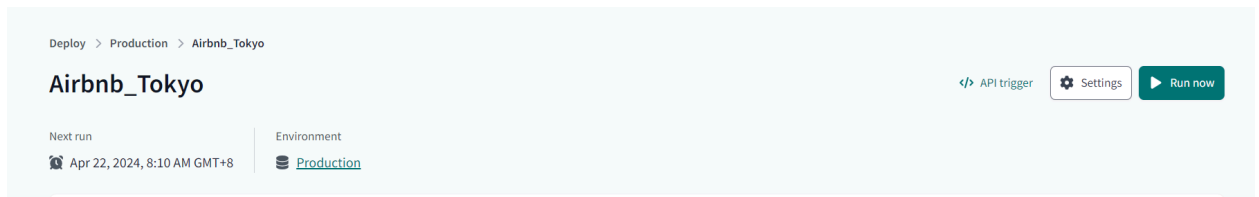
Transforming the data in the data warehouse: prepare it for the dashboard

Use DBT to load the data to BigQuery.

1. Staging:
 - a. Load data source from dataset_airbnb_tokyo.
 - b. Build a staging model.
 - c. Define the data types and rename columns.
 - d. Exposing the output of a dbt model in a warehouse as a view.
2. Core:
 - a. Contain only related production ready data.
 - b. Exposing the output of a dbt model in a warehouse as a table to Bigquery.



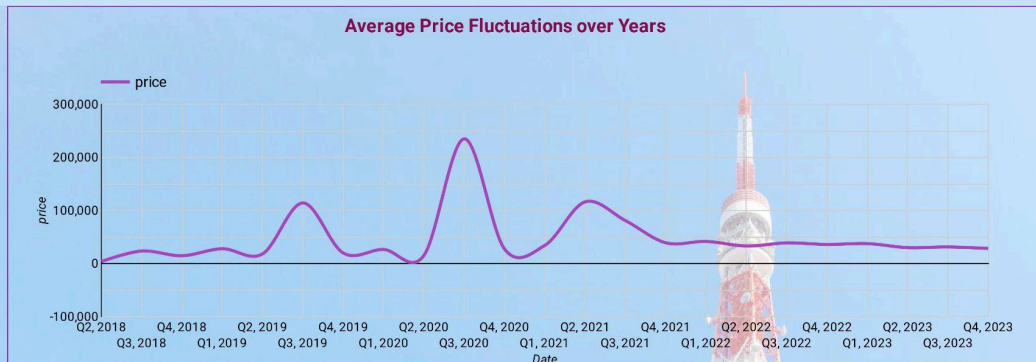
3. Environment and Job Triggers:
 - a. The production environment will be set up as well as the job of deploy the pipeline once a week on Monday.



Building a dashboard to visualize the data

1. Open fact_airbnb_tokyo dataset in BigQuery and select Explore in Looker
2. Build dashboard as below:

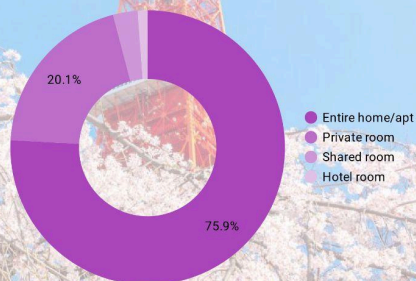
Airbnb in Tokyo From 2018/05/31 to 2023/12/28



10 Most Popular Neighbourhood in Tokyo

| | Neighbourhood | Records |
|-----|---------------|---------|
| 1. | Shinjuku Ku | 2,621 |
| 2. | Taito Ku | 1,742 |
| 3. | Sumida Ku | 1,610 |
| 4. | Toshima Ku | 1,216 |
| 5. | Shibuya Ku | 843 |
| 6. | Minato Ku | 538 |
| 7. | Ota Ku | 509 |
| 8. | Setagaya Ku | 434 |
| 9. | Nakano Ku | 387 |
| 10. | Kita Ku | 361 |

Distribution of Airbnb House Type

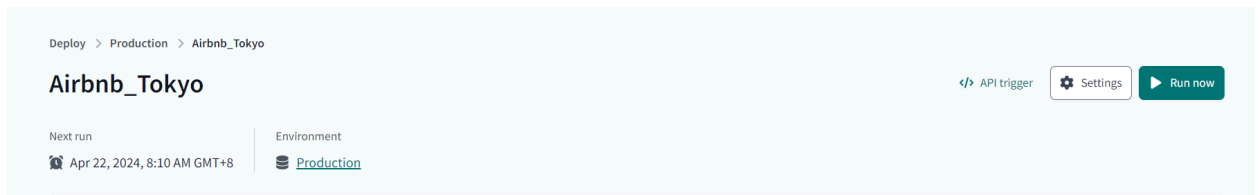


<https://lookerstudio.google.com/s/IHeLfPk99I0>

Reproducibility:

1. Create a new project on Google Cloud with name: example-de-zoomcamp
2. Setup service account for project:
 - a. Go to IAM & Admin > Service Accounts.
 - b. Create Service account
 - i. Enter Service account name: airbnb.
 - ii. Create and continue.
 - iii. Grant this service account access to project : Viewer, Storage Object Admin, Storage Admin, BigQuery Admin
 - iv. Continue > Done.

- c. Go to Manage Key> Add New Key> Json> Create.
3. Rename the json file to my_credential.json and Move the downloaded json file to folder keys.
4. Run commands:
 - a. terraform init
 - b. terraform plan
 - c. terraform apply
5. Mage:
 - a. docker compose up
 - b. Go to <http://localhost:6789/>
 - c. Run pipeline: airbnb_to_gcs to deploy the pipeline.
6. BigQuery:
 - a. Run query in airbnb_tokyo.sql.
7. DBT:
 - a. Set up a DBT project.
 - b. Add this github repo for version control.
 - c. Connect to Bigquery using my_credential.json.
 - d. Run build.
 - e. Set up the job in a production environment.
 - f. Job will be triggered on every Monday.



8. Go to BigQuery and table fact_airbnb_tokyo should exist in BigQuery Dataset.
9. Select the table and select Explore in Looker to develop the dashboard.