Intro

Hi there, in this documentation I would like to go through the business case study. From planning phase to building phase as well as areas that need improvement.

Plan

At the first when I gone through data and business questions it appear to me to obtain source we can go with simple extract where have python code to extract csv data directly with the following urls:

https://davidmegginson.github.io/ourairports-data/airports.csv
https://davidmegginson.github.io/ourairports-data/airport-frequencies.csv
https://davidmegginson.github.io/ourairports-data/airport-comments.csv
https://davidmegginson.github.io/ourairports-data/runways.csv
https://davidmegginson.github.io/ourairports-data/navaids.csv
https://davidmegginson.github.io/ourairports-data/countries.csv
https://davidmegginson.github.io/ourairports-data/regions.csv

And apply upsert into the database with the assumption ‘id’ being unique so on id conflict we will apply upsert else ignore as well as insert if data doesn’t exist. Thereafter once data is ready we will setup DBT to our postgres database and start applying transformation as per business questions listed below:

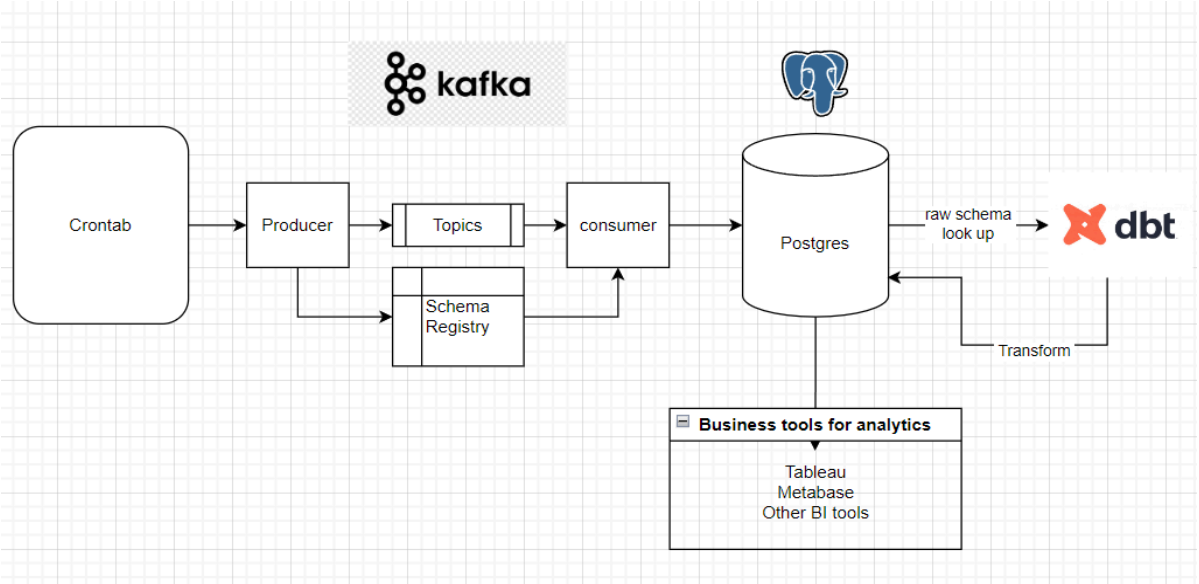
1. How many airports, airfields and heliports exist in each country and continent?
2. What is the average elevation of the airports, airfields and heliports in each country?
3. What is the estimated population of each country?
4. How many cities/towns/settlements in each country?
5. What is the min, max and average elevation of the cities per country?
6. Which are the highest and lowest elevated cities in the world with populations > 100000?
7. Which are the highest and lowest elevated airports, airfields and heliports on the planet?

Then we will set up a crontab to help us schedule our jobs. The crontab will schedule a time say daily or hourly to look up when source data (ourairports) was last updated via github's last modified date and trigger csv fetch then upsert to database again.

DBT transformation jobs can utilise crontab too so that it will execute say daily or hourly depending on the need.

Before I started building I noticed as per requirement
Note a working solution needs to be presented in the interview, with an etl tool (ie: KETL, talend, apache kafka. etc) and a database backend (ie: mysql, Oracle, postgreSQL. etc)**

Where an etl tool is required. So I decided to go with apache kafka as it's quick to spin up with docker. Thereafter my solution changed, see diagram below:



As per diagram above we will use crontab to schedule a job that will look up last modified data from github and trigger extraction of csv data and produce data to kafka with its schema (user defined) then have consumer/s to upsert into raw schema of our database base on the schema given. Thereafter based on business requirement needs we write DBT script to perform transformation and store back into insight schema.

Building

Please see attached zip for implementation.
Apologies in advance there are some areas I didnt code it up as I'm running out of time. Current work has been a bit demanding this week. I will describe a bit more in Areas of improvement section.

One can follow the instruction below to fire up my solution:

- Ensure you have docker desktop or docker core installed that you will be able to run docker-compose up command.
- Navigate to where zip file is being unzipped
  Eg. cd path-of-unzipped-location\fnb-pipeline
  And execute '**docker-compose up --build**'
  PS This may take a while depending on network connection. As well as data extract. So if you like feel free to go grab a cup of tea or coffee 🙂
- To exit just run **'docker-compose down**'

Areas of improvement

Here I would like to mention areas where I can improve my code:

1. In this demo I setup Crontab where we could have set up an Airflow instance as an orchestrator for a nice friendly UI. Note. Crontab doesn't work at this moment within docker container for some reason which means we will need to execute scrape_world_population.py as well as dbt run manually. (Apologies that I couldn't get this done in time!)

2. I could potentially set up a S3 bucket to download CSV (For reloading history if needed) before extraction.
3. Since the above 2 points I'm fully aware that my solution provided won't update further csv pull automatically unless re-execution of the 'airports.py' script (provided DB volume didn't get reset.)
4. CSV schema in the code you will notice I have manually setup in 'config.py' where avro_schema is manually configured and there isn't a schema registry service, but since csv files can be unpredictable it is very important that producer will provide a schema so we can keep track of change of data and setup schema and data evolution strategy
5. For kafka instance , since it is for demo purposes it is not fully set up in a way that is secure as well as setting up retention policy and other configurations.
6. My configuration for DB param, Kafka bootstrap server, can go into the docker environment variable and obtain it that way to improve security.
7. Unit tests as well as DBT tests should be set up to help us monitor our deployment as well as our code, incoming data violation and data structure
8. Database end ideally raw source tables should be created based on schema provided via schema registry dynamically.
9. I could possibly do some data validation to ensure our data are valid and trustworthy but this will require proper planning and feedback.
10. Producer and consumer could do some more optimisation as well as fine tuning on batch processing.

Business Requirements:

- How many airports, airfields and heliports exist in each country and continent?
- What is the average elevation of the airports, airfields and heliports in each country?
- What is the min, max and average elevation of the cities per country?
- Which are the highest and lowest elevated airports, airfields and heliports on the planet?

Above questions you will be able to find it in my dbt transformation script after executing dbt run
However the sample data are still limited to provide accurate results.

- What is the estimated population of each country?
- Which are the highest and lowest elevated cities in the world with populations > 100000?

Above questions I couldn't answer them as the population dataset wasn't really included in our sample data. Having this said I did go ahead and scrape world population from https://www.worldometers.info/world-population/population-by-country/ but since this is country data so cities question I wont really be able to answer.

- How many cities/towns/settlements in each country?

This one I answered base on airports data set on municipality instead of via pgeocode lib as it provides more data for more countries where pgeocode have limit supported countries. I could possibly combine them to get the best possible results.

Conclusion

Thank you very much for looking through the above implementation and documentation. Please enlighten me if I miss any area that I could improve further.