# Project Objective

Analyse Brazil property data from 1st Jan 2015 to 31st Dec 2016. Brazil property data contains sales and rent data. So my goal is to find the distribution between sales and rent data by creating a dashboard.

The dataset is available in available in google cloud public data set. <https://console.cloud.google.com/marketplace/product/properati/property-data-br>

# Project environment setup

Operating system – Windows

Tools used – Python, Terraform, Docker desktop, Git bash, [Github account](https://github.com/), [DBT cloud developer account](https://cloud.getdbt.com/login/), [Google data studio](https://datastudio.google.com/).

Please make sure these tools installed on your system – Python, [Terraform](https://www.terraform.io/), [Docker desktop](https://docs.docker.com/get-docker/), [Git bash](https://git-scm.com/downloads)

Google cloud is using for project infrastructure. Google cloud storage is using as Data Lake, Bigquery as Datawarehouse, Google Data Studio as Dashboard tool

In order to setup project in your local environment, First clone the project from github repository

<https://github.com/haris-m-aslam/dtc-de-project>

git clone [git@github.com:haris-m-aslam/dtc-de-project.git](mailto:git@github.com:haris-m-aslam/dtc-de-project.git)

# Google cloud setup

Create a [google cloud](https://cloud.google.com/) account if you don’t have one. Other wise login to the cloud account and go to [console](https://console.cloud.google.com/home/dashboard).   
Create new project and note down the project id

1. From GCP Dashboard, click on the down arrow next to **Google Cloud Platform** title to show the project list and click on NEW PROJECT.
2. Enter the project name(example: DTC DE Project). You can use the autogenerated Project ID. Leave the organization as No organization. Click on Create. **Note down the Project ID**
3. Select the newly created project from project list dropdown

## Setup a service account for this project and download the JSON authentication key files

IAM & Admin > [Service accoun](https://console.cloud.google.com/iam-admin/serviceaccounts)ts > Create service account

You can go to Service Account by searching on the dashboard and **Service Accounts in IAM & Admin**

https://console.cloud.google.com/iam-admin/serviceaccounts

1. Click on **CREATE SERVICE ACCOUNT**
2. Enter a service account name (example: DTC De Project User) Leave all other fields with the default values. Click on CREATE AND CONTINUE.
3. Grant the Viewer role (Basic > Viewer) to the service account and click on CONTINUE
4. There is no need to **Grant users access to this service account** at the moment. Click on DONE.
5. With the service account created, click on the 3 dots below Actions and select **Manage keys**.
6. ADD KEY > Create new key. Select JSON and click Create. The files will be downloaded to your computer.
7. Move and rename the downloaded key file to

C:\Users\<your name>\.google\credentials\ google\_credentials.json

## Google cloud access setup

We need to setup access first by assigning the Storage Admin, Storage Object Admin, BigQuery Admin and Viewer IAM roles to the Service Account, and then enable the **iam** and **iamcredentials** APIs for our project.

1. In GCP dashboard, go to IAM & Admin > [IAM](https://console.cloud.google.com/iam-admin/iam)

<https://console.cloud.google.com/iam-admin/iam>

1. Select the previously created Service Account and edit the permissions by clicking on the pencil shaped icon on the left
2. Add the following roles by clicking ADD ANOTHER ROLE and click on Save afterwards:

* Storage Admin: for creating and managing buckets.
* Storage Object Admin: for creating and managing objects within the buckets.
* BigQuery Admin: for managing BigQuery resources and data.
* Viewer should already be present as a role.

1. Enable APIs for the project (these are needed so that Terraform can interact with GCP)

* <https://console.cloud.google.com/apis/library/iam.googleapis.com>
* <https://console.cloud.google.com/apis/library/iamcredentials.googleapis.com>

# Terraform

### We are using Terraform as IAC tools and it will help us to create and configure data lake and datawarehouse in Google cloud

Open the cloned project files in your IDE and do the following

Update your google cloud region in **variables.tf** under region variable (example - **asia-south1**) (line number 11)

Update your gcp auth key location **variables.tf** under credentials variable(example - **C:/Users/haris/.google/credentials/google\_credentials.json**) (line number 17)

Go to Project folder -> Terraform folder

Open this location in git bash

Run the following commands. Terraform plan and terraform apply commands will ask for your **GCP project id**

terraform init  
terraform plan

terraform apply

Verify the Data lake and Datawarehouse created by the Terraform in the google cloud

Go to [Storage service](https://console.cloud.google.com/storage/browser) and verify Data lake created. Name will be like this **dtc\_de\_project\_data\_lake\_<your project id>**.

Go to [Big query service](https://console.cloud.google.com/bigquery) and verify Datawarehouse create. Dataset names will be like this   
br\_properties\_data\_all, development and production

Go to [Storage service](https://console.cloud.google.com/storage/browser) and note down the Data lake bucket id. In our case Bucket id will be like this **dtc\_de\_project\_data\_lake\_dtc-de-project-344607**

# Airflow

### We are using Airflow Pipeline orchestration tool

Go to **airflow** folder  
Copy and rename .env\_example file as .env file

Update GCP\_PROJECT\_ID and GCP\_GCS\_BUCKET with your Google project id and Google cloud storage bucket id in **.env** file

Update GCP\_PROJECT\_ID and GCP\_GCS\_BUCKET in docker-compose.yaml

**Start your Docker deskop**

### Run the following commands to set up Pipeline environment in docker

docker-compose build  
docker-compose up airflow-init  
docker-compose up -d

### Running the piplines

Pipelines will move our data from Google cloud public data set to our Datalake, Import our data from Data Lake to Datawarehouse(Big Query)

Sales data and rent data are moved to Datawarehouse two separate pipelines and another pipelines for making partitions in datawarehouses

To run Pipelines go to <http://localhost:8080/> (Port number will be the port number of dtc-de\_airflow-worker container. Container can be found be running this command docker ps -a in your terminal(git bash))

Login to Airflow. Username – airflow, password – airflow

run the Dags in the following order. Dag can be run clicking the Play in the right side -> **Trigger DAG**

run sell\_data\_ingestion\_dag  
run rent\_data\_ingestion\_dag  
run bq\_partioning\_dag

Verify the tables (**rent\_data\_external\_table, sell\_data\_external\_table, rent\_data\_partitioned, sell\_data\_partitioned**) created in **br\_properties\_data\_all** dataset

**sell\_data\_ingestion\_dag** - Dag is using for moving property sales data of the years 2015-2016 from Google public dataset to Google cloud storage(Data lake) and then to BIgQuery(Data warehouse)

**rent\_data\_ingestion\_dag** - Dag is using for moving property rent data of the years 2015-2016 from Google public dataset to Google cloud storage(Data lake) and then to BIgQuery(Data warehouse)

**bq\_partioning\_dag** – Dag is using for partitioning the both sales and rent tables by **created\_on** column(for optimizing the query performance as we filter data by date in dashboard). Then clustering the tables by state\_name column

# DBT Cloud

We are using [DBT cloud](https://cloud.getdbt.com/) to do some transformation on our data in Datawarehouse. We need to merge Sales data and rent data to single table.

For that we are running the dbt in DBT cloud.

https://cloud.getdbt.com/

1. create new project in DBT cloud from Account Settings menu and click New Project.
2. Enter a Name for project (example - DTC DE Project) and Continue
3. Set Up a Database Connection and select BigQuery
4. In BigQuery settings section – Upload the GCP Auth key json file to the field **CREATE FROM FILE**
5. In Development Credentials section – Enter DATASET value as **development**
6. Click **Test** and then **Continue**
7. Create a new folder in your local machine and copy the dbt folder content from project folder to newly created folder
8. Then initialize git in new folder. Run the following commands in gitbash from newly created folder  
   git init

git remote add origin <url new dbt repo url>  
example - git remote add origin git@github.com:<your username>/dtc-proj-dbt-2.git

copy dbt folder content her  
git add .  
git commit -m “initial commit”

git push origin master

1. Click **Start Developing**
2. Create a new branch and name it as **dev**
3. Update database with your GCP Project id in models/staging/schema.yml
4. Run **dbt deps**
5. Run **dbt run**
6. Click **Commit and Open pull request**. It will leads to github repository and commit changes
7. After commit and merge to master
8. In dbt cloud, refresh and click **pull changes from master**
9. You can verify the dbt set up by running **dbt run** in console and it will create **Views** in **development** dataset in [Bigquery](https://console.cloud.google.com/bigquery?)
10. Create new environment from the left side(3 lines in top) and click on New Environment

Name – Production, Type – Deployment, Dataset – **production** and leave other fields as default

1. Create new job and clicking the production environment   
   NAME - dbt build, ENVIRONMENT – Production,   
   DBT VERSION - Inherit from Production (v1.0 (latest stable))  
   THREADS – 4  
   TARGET NAME – default  
   Execution Settings - default

Commands - **dbt run --var 'is\_test\_run: false'**Notifications - Default  
Triggers – Schedule on > daily -> every 6

1. Click **Save** and run the job by pressing **Run now**

It will create fact\_properties table in production dataset in datawarehouse

# Datastudio

Go to datastudio <https://datastudio.google.com/>

We will be using Datastudio to create charts to find the insights from fact\_properties table

1. Create data source by clicking the Create -> Data Source -> Select Big Query
2. Your GCP Project name -> production -> fact\_properties and connect
3. Select the Default aggregation (sum) for fields price, surface\_covered, surface\_total and for other fields as None
4. Click **CREATE REPORT**

Adding charts on the Datastudio dashboard

For adding charts in datastudio dashboard refer this [video](https://www.youtube.com/watch?v=39nLTs74A3E&list=PL3MmuxUbc_hJed7dXYoJw8DoCuVHhGEQb)

### For adding the chart of total sales and rent

Click Add a chart -> select scorecard -> Date range dimension :: none -> Metric :: record count

### For adding the chart of Total Number of rooms

Click Add a chart -> select scorecard -> Date range dimension :: none -> Metric :: rooms

### Sell/Rent Distribution chhart

Click Add a chart -> Pie chart -> Date range dimension :: none -> Dimension :: operation -> Metric :: record count

### Total sales per month

Click Add a chart -> stacked column chart (Bar chart) -> **ADD A FIELD**(in bottom right corner) -> name created\_month -> Formula MONTH(created\_on) , click SAVE and DONE-> Date range dimension :: created\_on -> Dimension :: created\_month -> Break down dimension :: operation -> Metric :: record count

### State wise sales

Click Add a chart -> Table with Heatmap (Table chart) -> Date range dimension :: none -> Dimension :: state\_name -> Metric :: record count

You can view my dashboard by clicking this [link](https://datastudio.google.com/reporting/8358b104-39b3-4268-ab85-dfa8aacf9408)