Solution Sprint - Fase 4

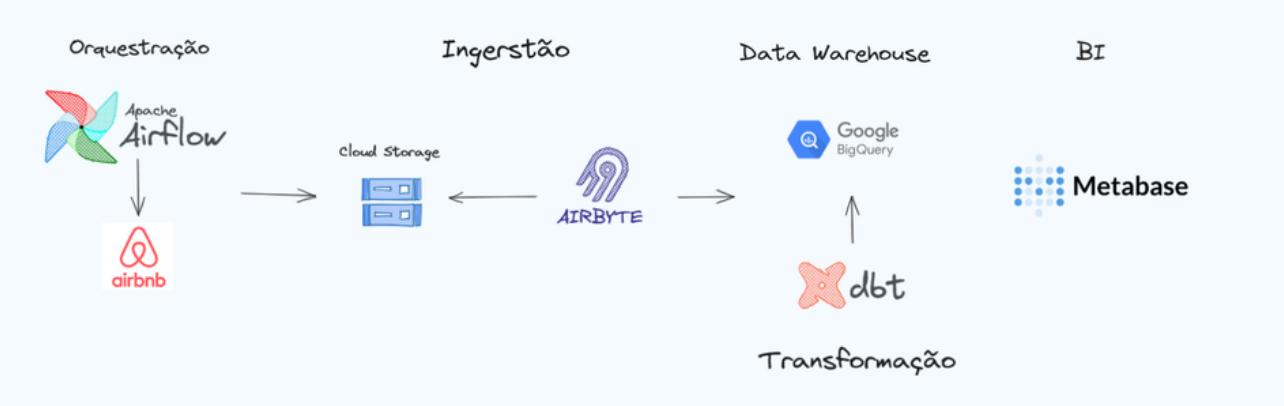
O desafio será construir uma plataforma de dados moderna (MDS) na nuvem e realizar integrações em real time para compor a solução. Para isso, vamos considerar as bases de dados do Inside Airbnb para o Rio de Janeiro e dados imobiliários sobre imóveis na cidade.

Requisitos Técnicos:

a) No mínimo uma ferramenta utilizada para ingestão de dados;
b) Uma camada de Storage, preferencialmente um Cloud Data Warehouse;
c) Uma camada de transformação diretamente no Storage;
d) No mínimo um dashboard ou report analisando os dados transformados;
e) No mínimo um modelo analítico envolvendo técnicas de aprendizado de máquina;
f) Uma solução para governança e catálogo de dados.

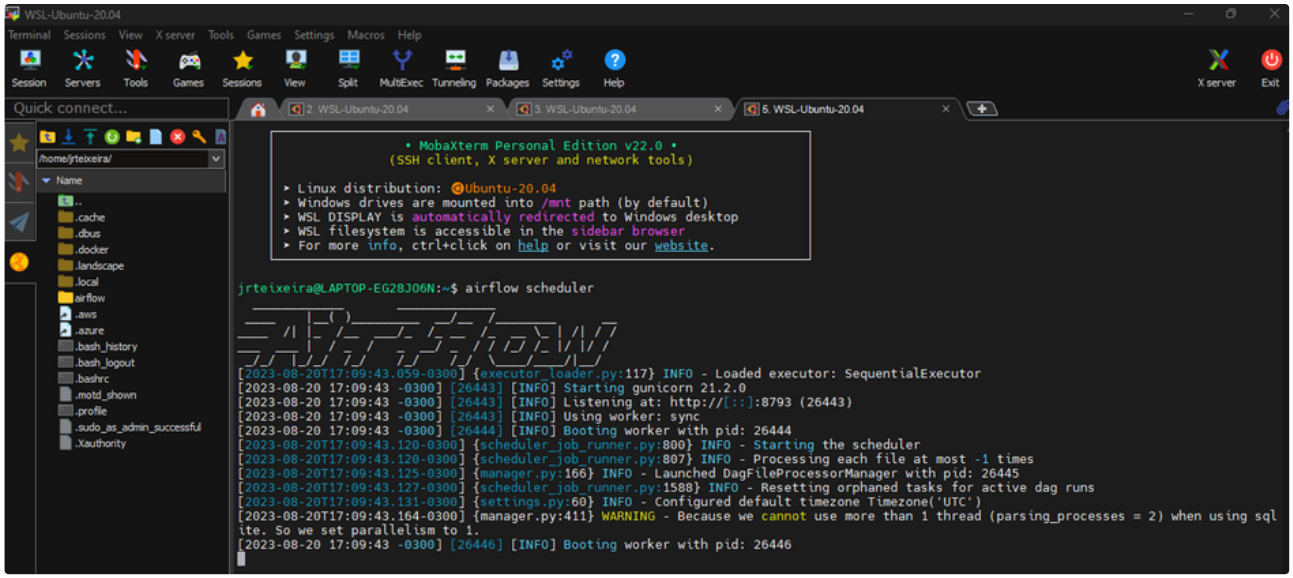
Entregável 1

Arquitetura projetada:

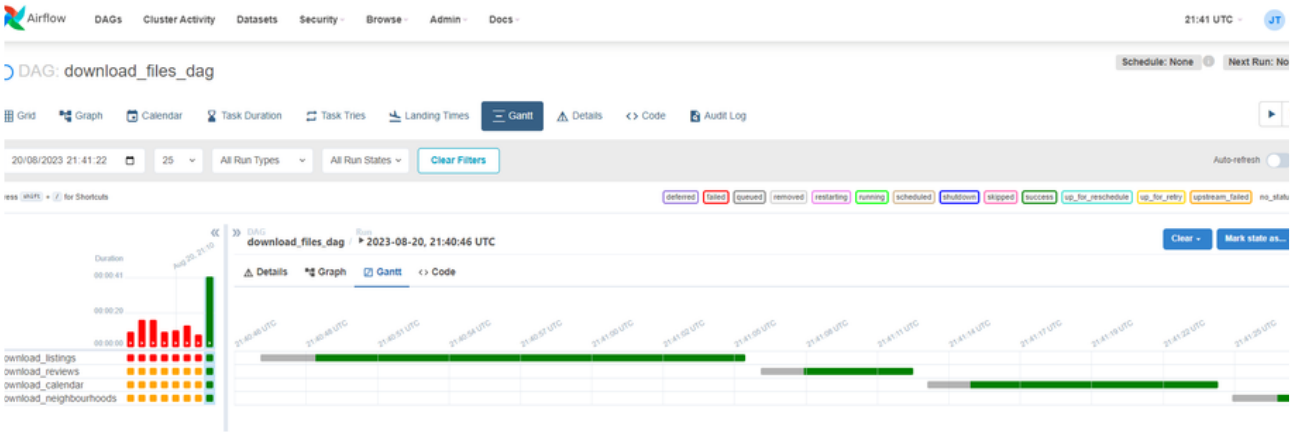


Orquestração: Airflow

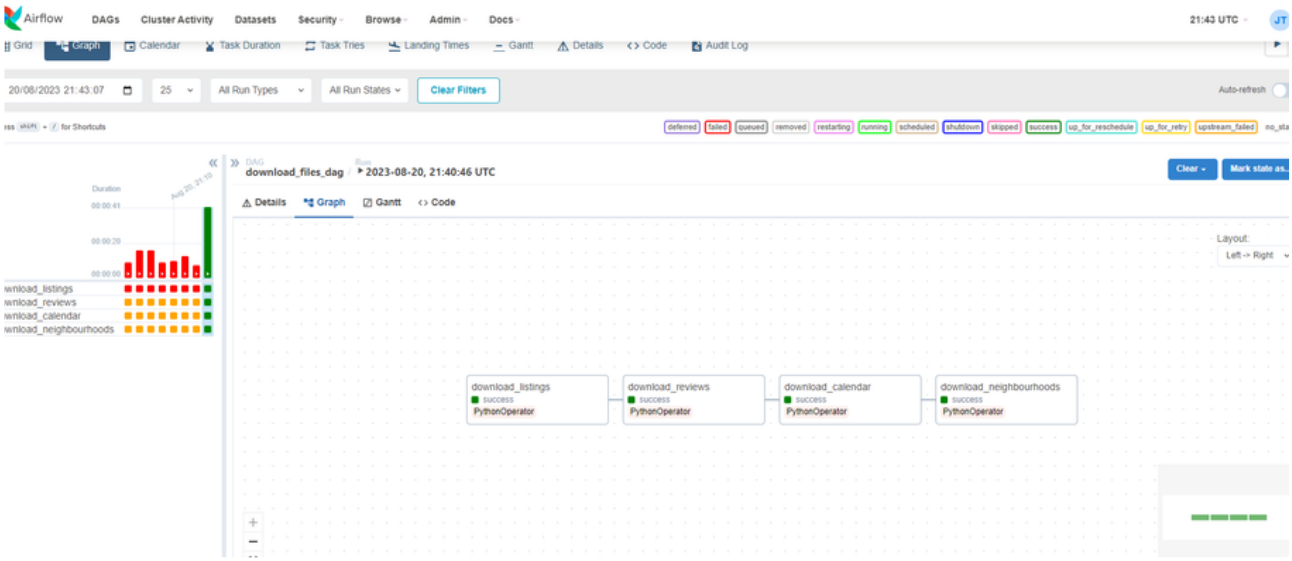
Evidências:



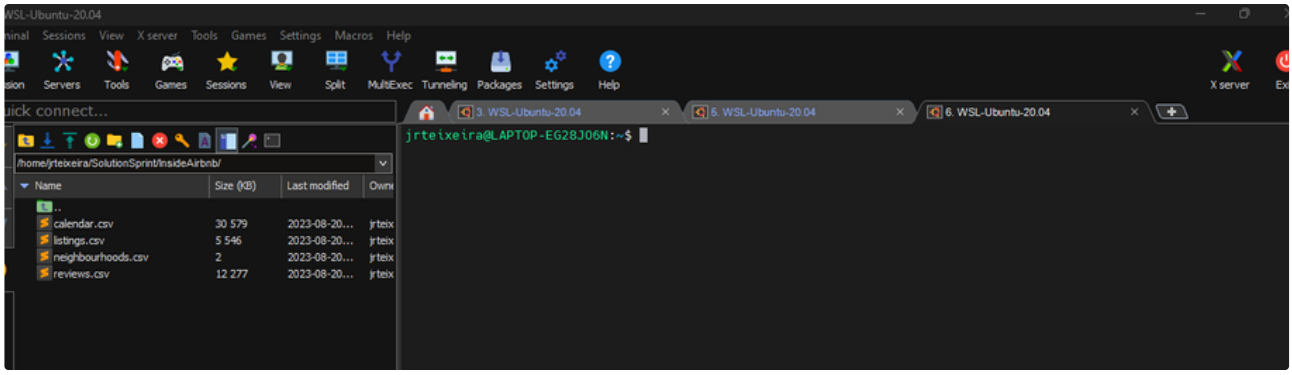
Start Airflow



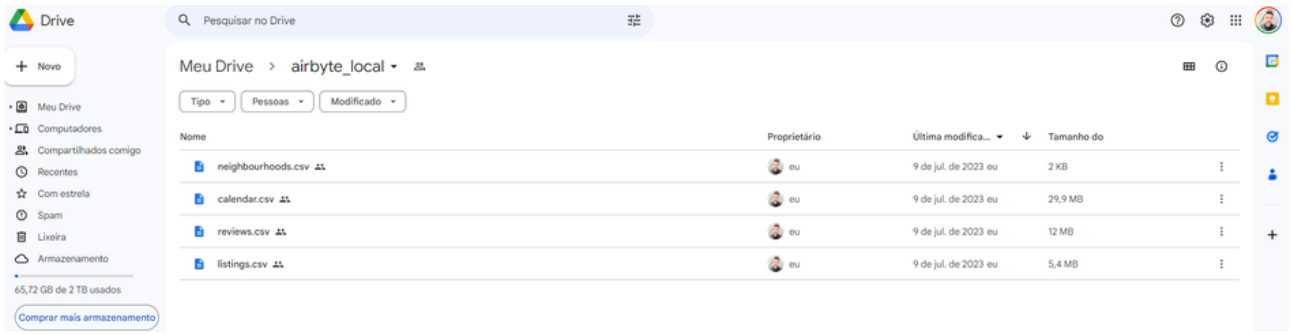
Airflow Tasks



Airflow Tasks

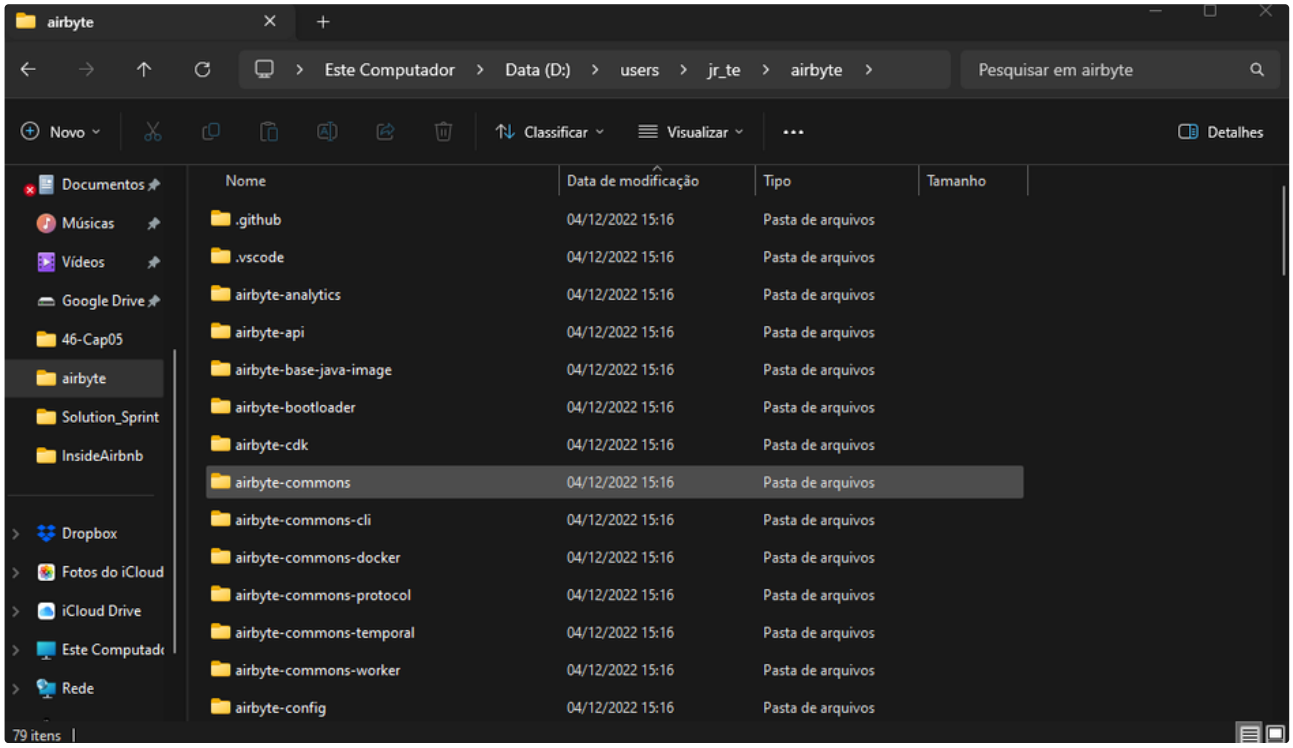


Arquivos listados no diretório local

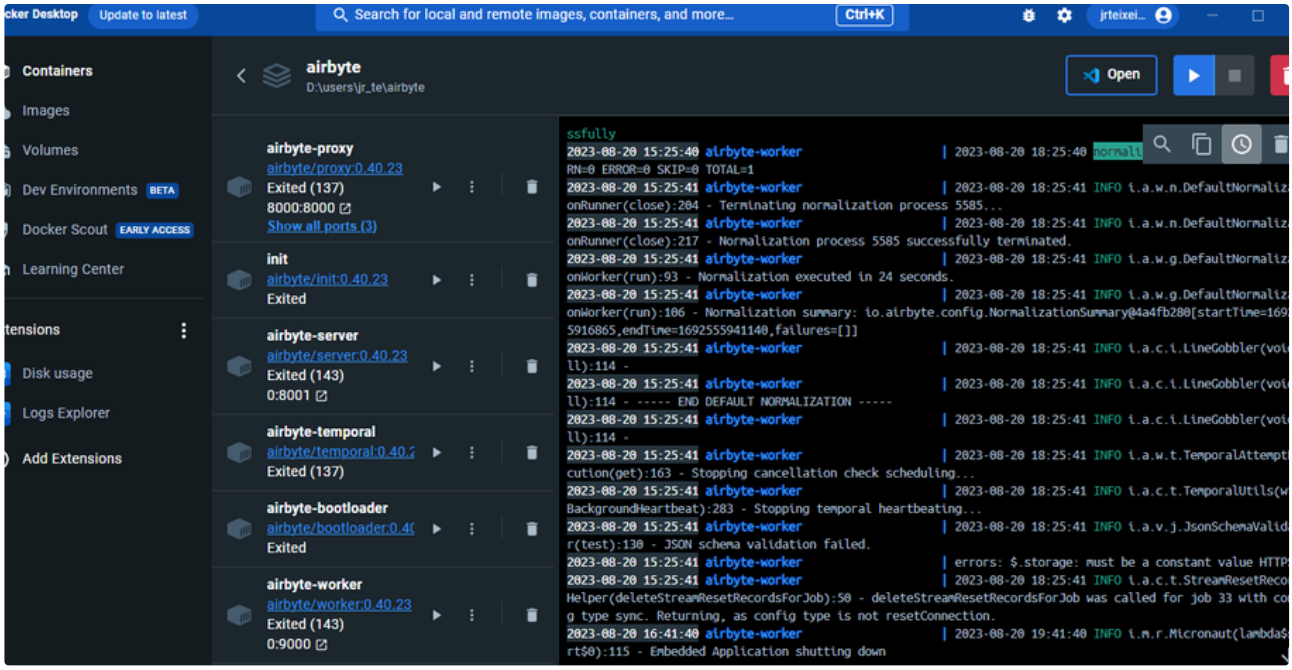


GDrive

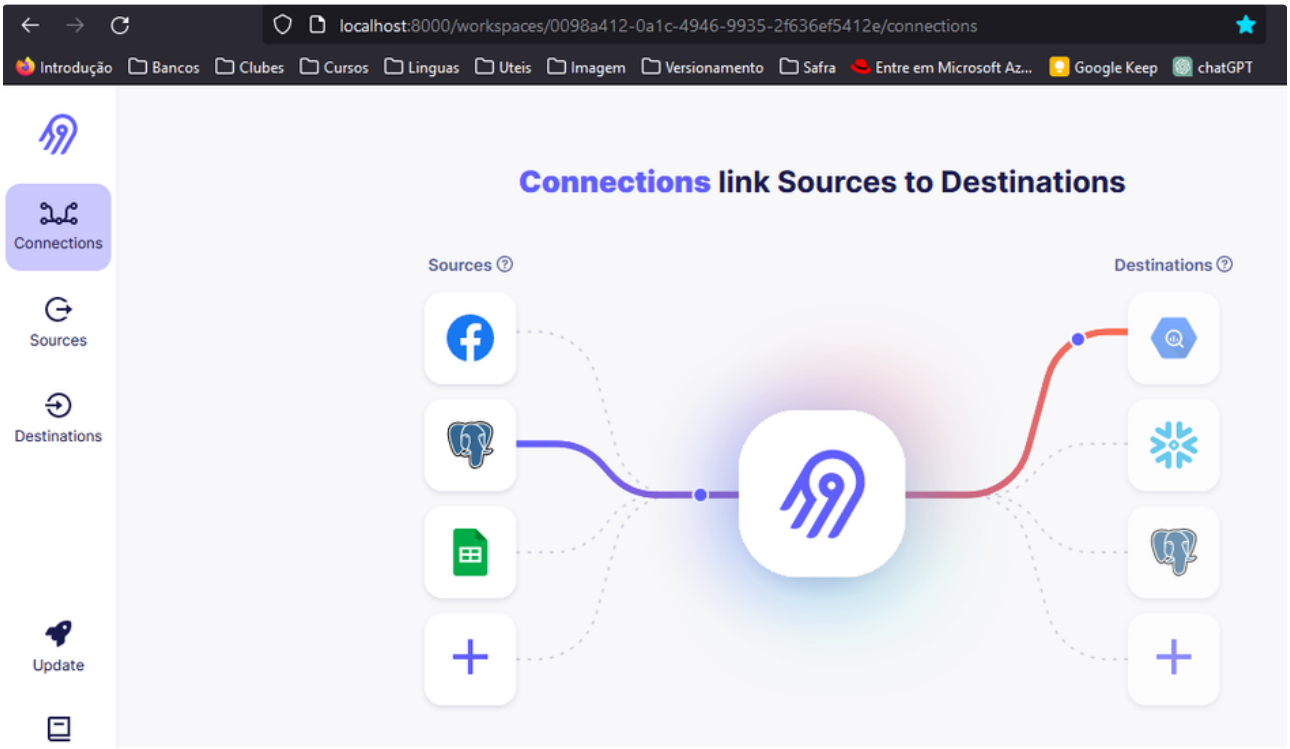
Ingestão: Airbyte



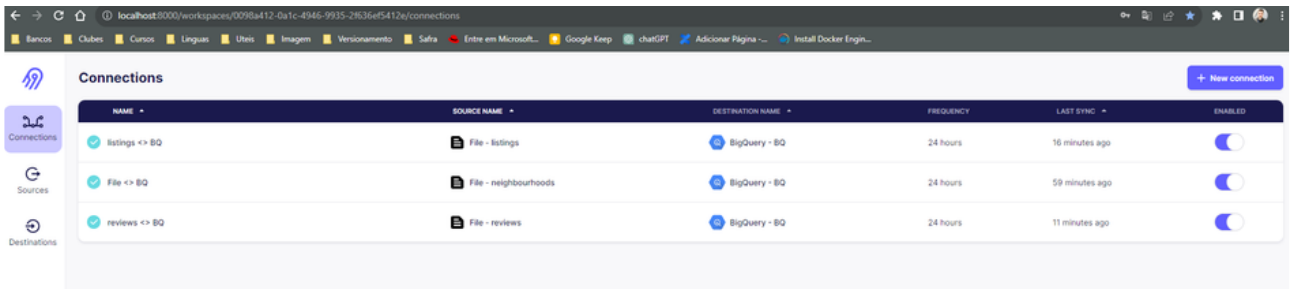
Git Clone do projeto Airbyte para diretório Local



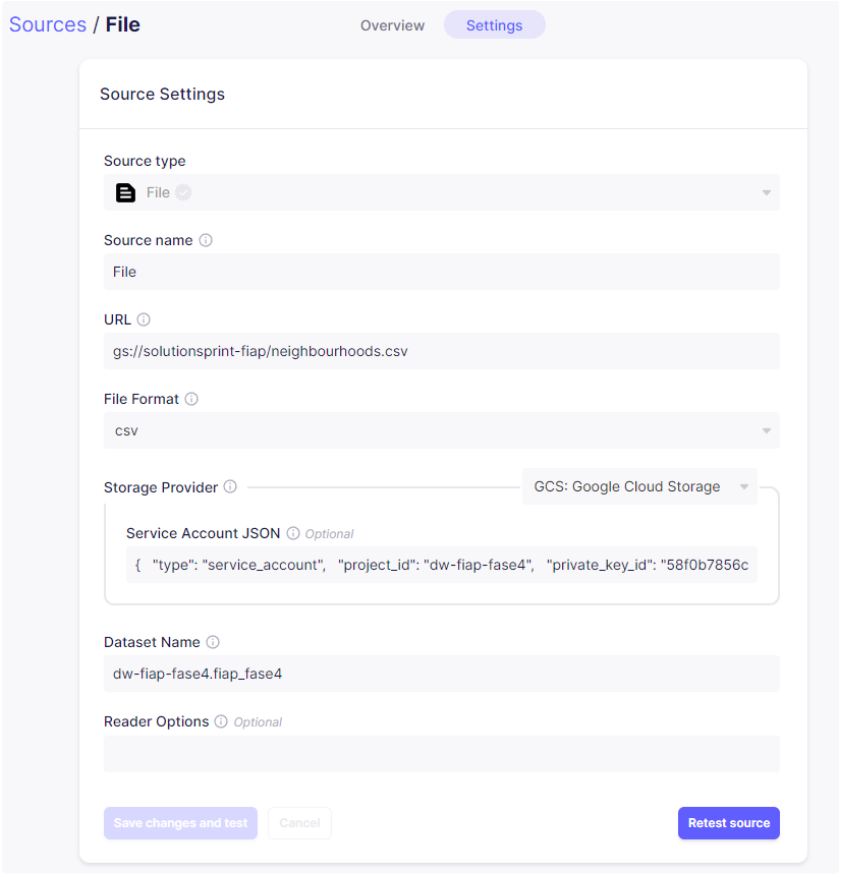
Airbyte Docker



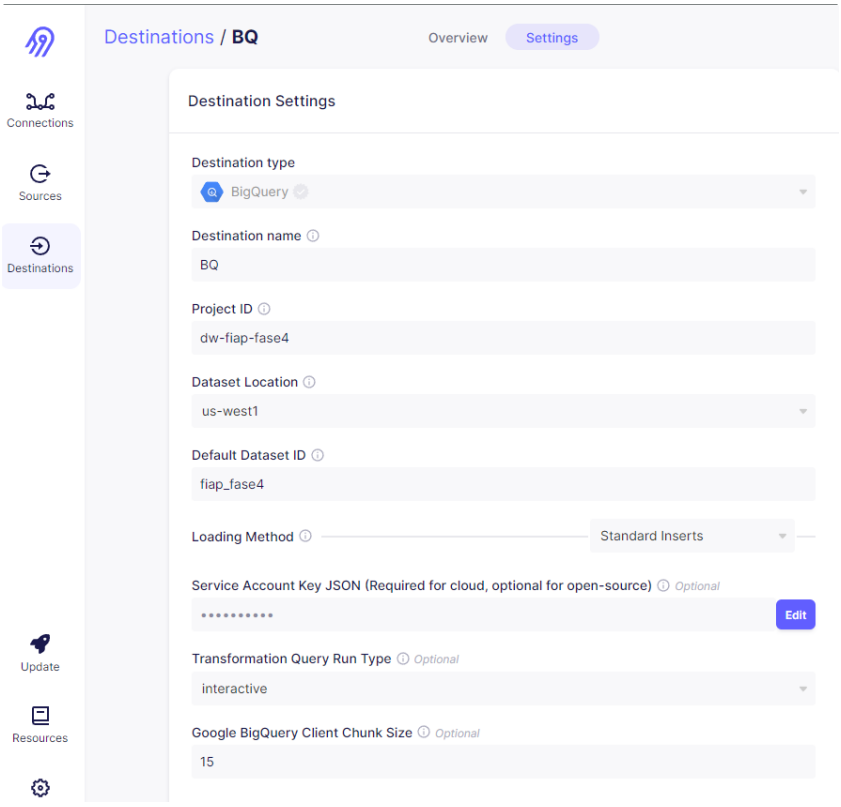
Página Inicia Airbyte



Conexões criadas

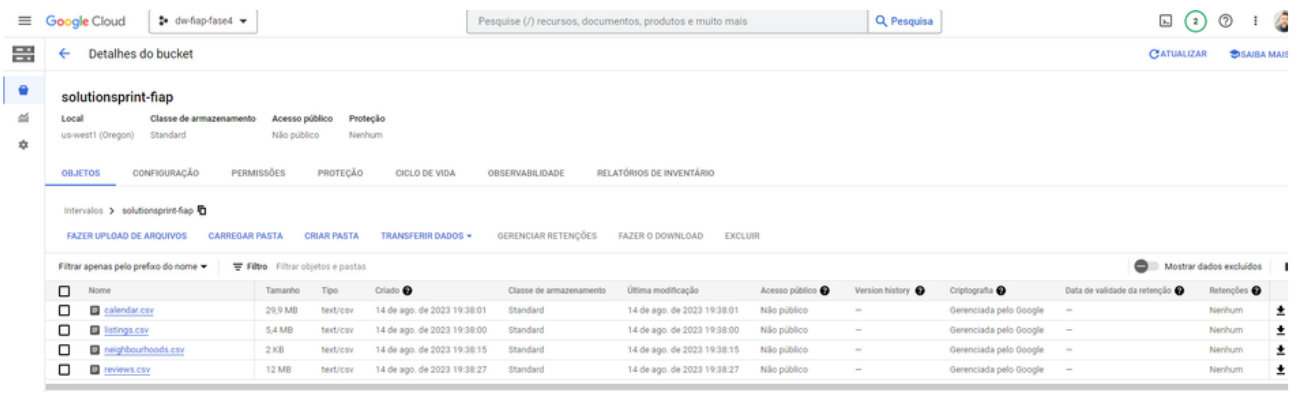


Configuração Source



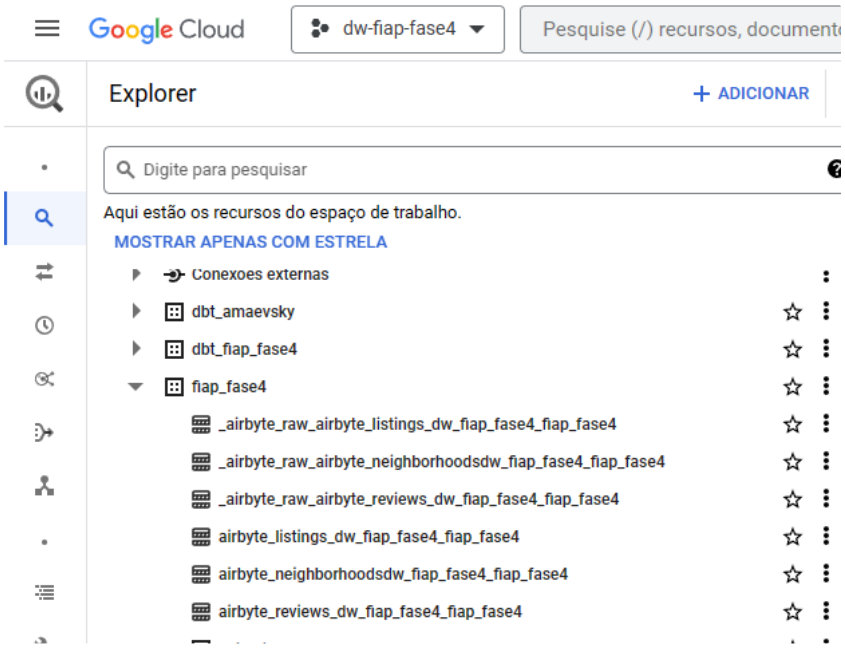
Configuração Destination

Camada de Armazenamento: Cloud Storage (GCP)

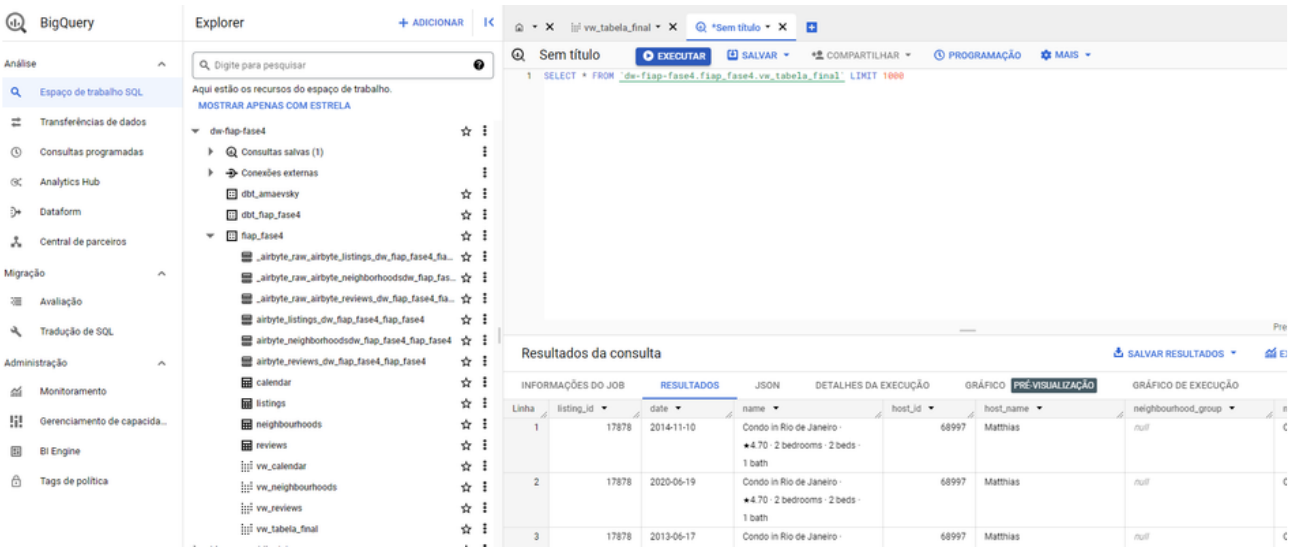


Arquivos armazenados Bucket

Data WareHouse: Google BigQuery:



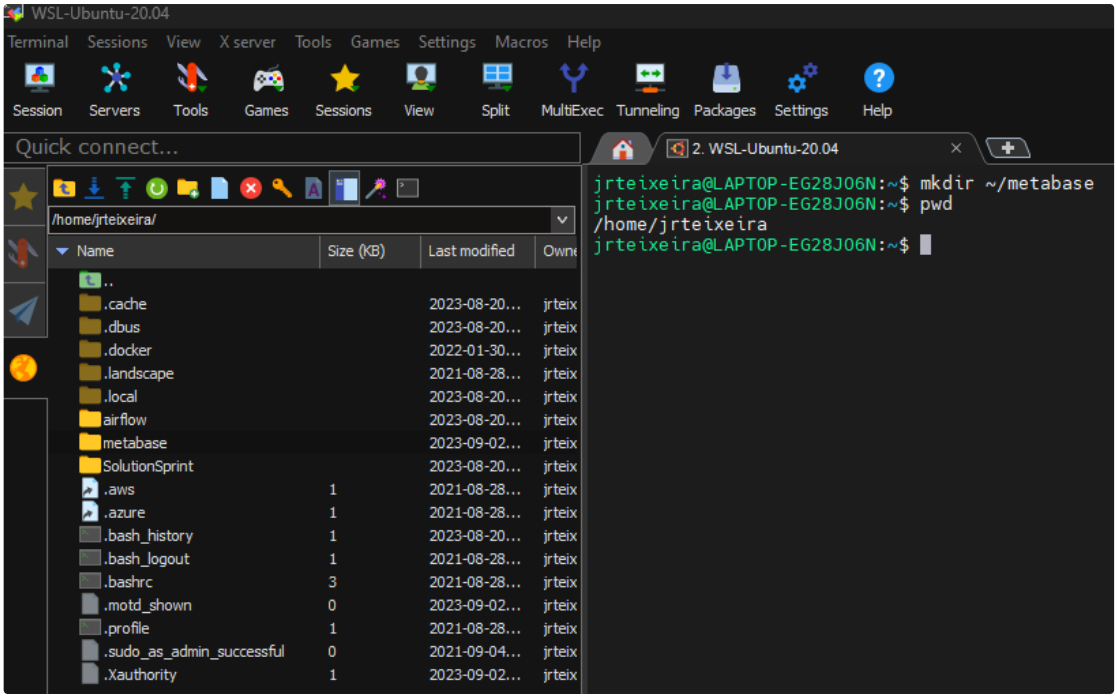
Datasets criados via Airflow



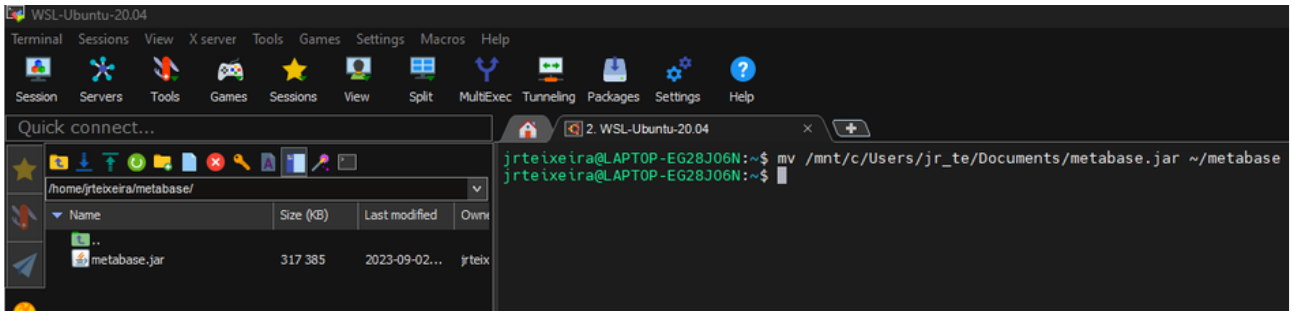
View Criada processo dbt

Camada de Apresentação: Metabase

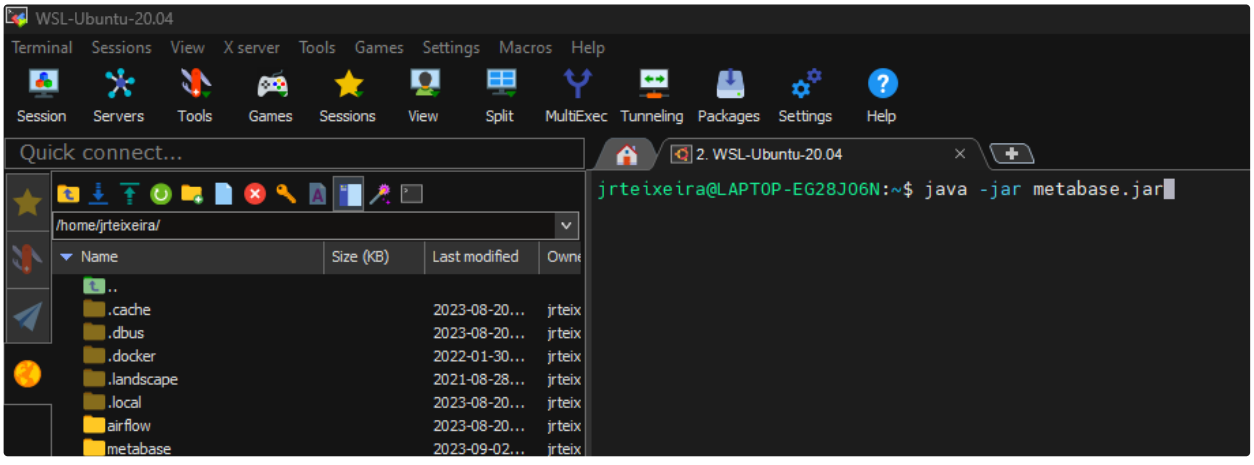
Ver arquivo procedimento de instalação metabase.txt



Criando diretório metabase

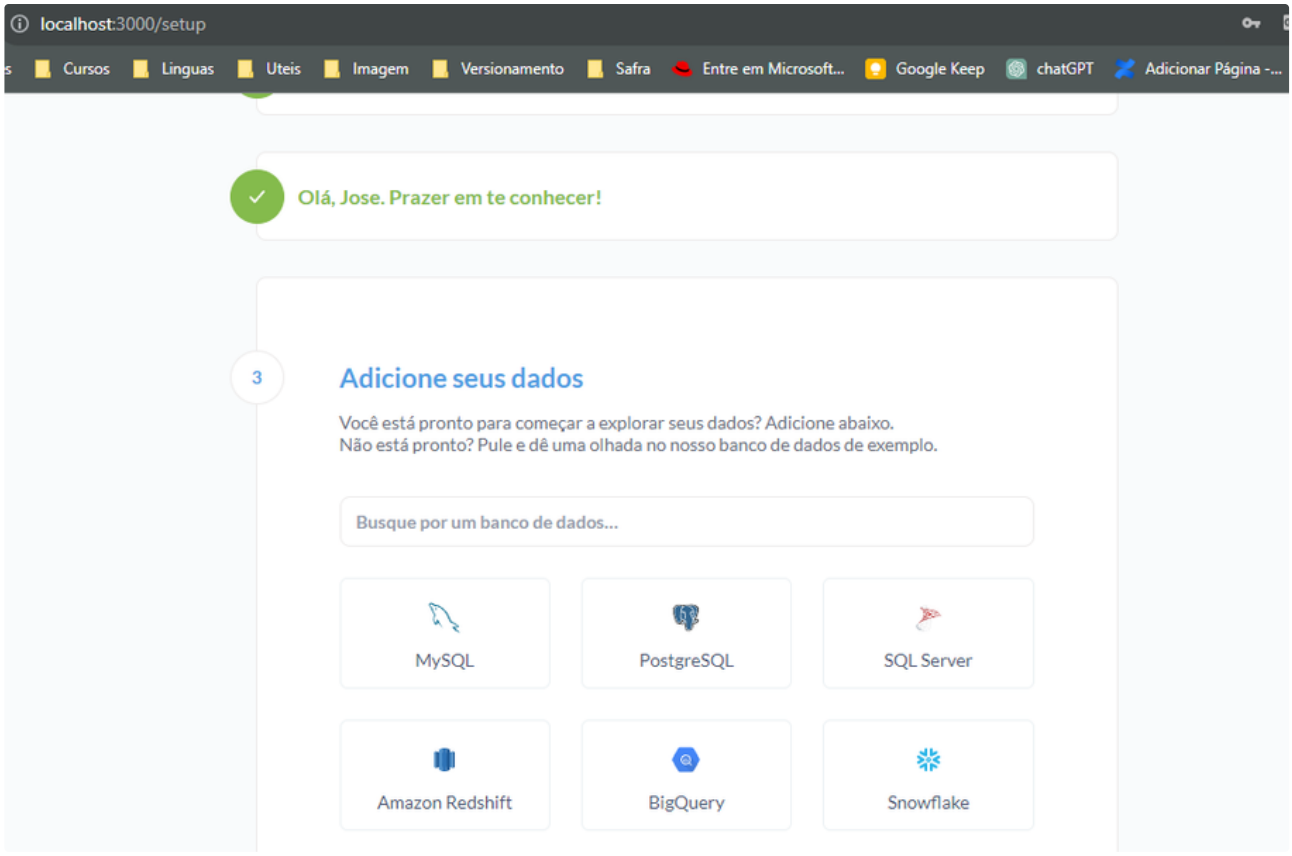


Move arquivo JAR para novo diretório

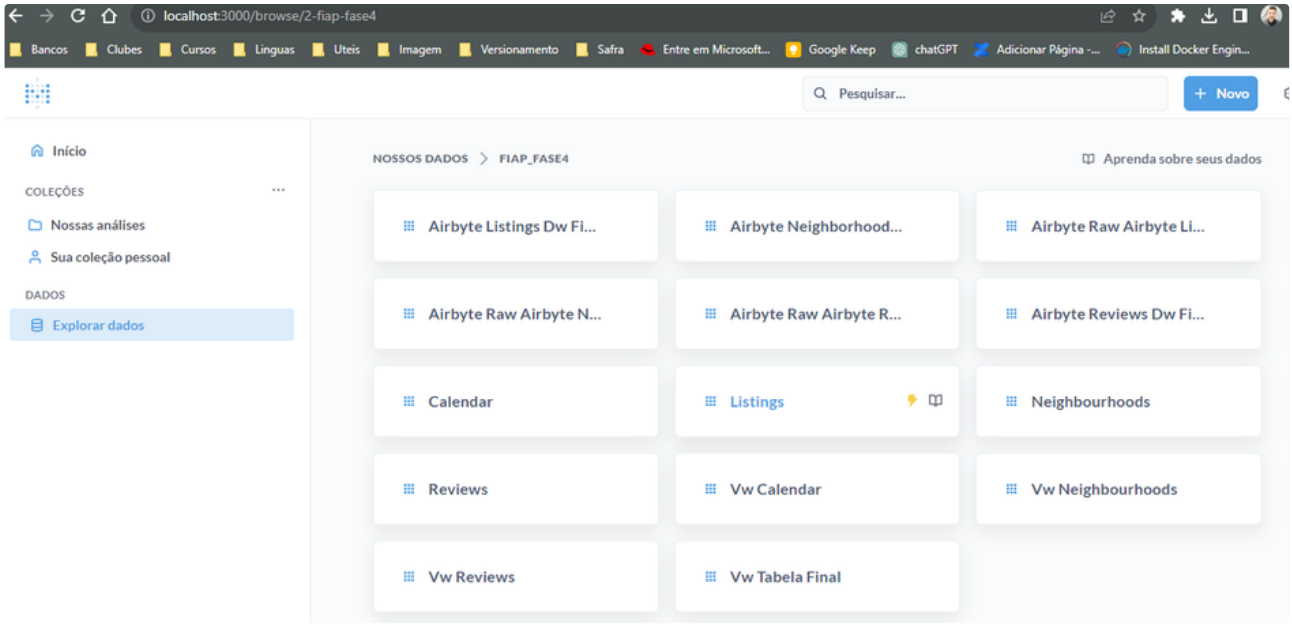


Run JAR

```
jrteixeira@LAPTOP-EG28JO6N:~/metabase$ echo $JAVA_HOME
/usr/lib/jvm/java-11-openjdk-amd64
jrteixeira@LAPTOP-EG28JO6N:~/metabase$ java -jar metabase.jar
2023-09-02 16:17:05,609 INFO metabase.util :: Maximum memory available to JVM: 2.2 GB
2023-09-02 16:17:10,825 WARN db.env :: WARNING: Using Metabase with an H2 application database is not recommended for production deployments. For production deploym
commend using Postgres, MySQL, or MariaDB instead. If you decide to continue to use H2, please be sure to back up the database file regularly. For more information,
ase.com/docs/latest/operations-guide/migrating-from-h2.html
2023-09-02 16:17:11,324 INFO util.encryption :: Saved credentials encryption is DISABLED for this Metabase instance.
 For more information, see https://metabase.com/docs/latest/operations-guide/encrypting-database-details-at-rest.html
2023-09-02 16:17:28,027 INFO driver.impl :: Registered abstract driver :sql
2023-09-02 16:17:28,062 INFO driver.impl :: Registered abstract driver :sql-jdbc (parents: [:sql])
2023-09-02 16:17:28,079 INFO metabase.util :: Load driver :sql-jdbc took 308.4 ms
2023-09-02 16:17:28,081 INFO driver.impl :: Registered driver :h2 (parents: [:sql-jdbc])
2023-09-02 16:17:28,243 INFO driver.impl :: Registered driver :mysql (parents: [:sql-jdbc])
2023-09-02 16:17:28,363 INFO driver.impl :: Registered driver :postgres (parents: [:sql-jdbc])
2023-09-02 16:17:37,078 INFO metabase.core ::
Metabase v0.47.0 (682ec2d release-x.47.x)

Copyright © 2023 Metabase, Inc.

Metabase Enterprise Edition extensions are NOT PRESENT.
2023-09-02 16:17:37,108 INFO metabase.core :: Starting Metabase in STANDALONE mode
2023-09-02 16:17:37,302 INFO metabase.server :: Launching Embedded Jetty Webserver with config:
 {:port 3000}

2023-09-02 16:17:37,461 INFO metabase.core :: Starting Metabase version v0.47.0 (682ec2d release-x.47.x) ...
2023-09-02 16:17:37,530 INFO metabase.core :: System info:
 {"file.encoding" "UTF-8",
  "java.runtime.name" "OpenJDK Runtime Environment",
  "java.runtime.version" "11.0.20.1+1-post-Ubuntu-0ubuntu120.04",
  "java.vendor" "Ubuntu",
  "java.vendor.url" "https://ubuntu.com/",
  "java.version" "11.0.20.1",
  "java.vm.name" "OpenJDK 64-Bit Server VM",
  "java.vm.version" "11.0.20.1+1-post-Ubuntu-0ubuntu120.04",
  "os.name" "Linux",
  "os.version" "5.10.102.1-microsoft-standard-WSL2",
  "user.language" "en",
  "user.timezone" "America/Sao_Paulo"}

2023-09-02 16:17:37,535 INFO metabase.plugins :: Loading plugins in /home/jrteixeira/metabase/plugins...
2023-09-02 16:17:38,646 INFO util.files :: Extract file /modules/sparksql.metabase-driver.jar -> /home/jrteixeira/metabase/plugins/sparksql.metabase-driver.jar
2023-09-02 16:17:39,046 INFO util.files :: Extract file /modules/athena.metabase-driver.jar -> /home/jrteixeira/metabase/plugins/athena.metabase-driver.jar
2023-09-02 16:17:39,696 INFO util.files :: Extract file /modules/sqlserver.metabase-driver.jar -> /home/jrteixeira/metabase/plugins/sqlserver.metabase-driver.jar
2023-09-02 16:17:39,793 INFO util.files :: Extract file /modules/presto-jdbc.metabase-driver.jar -> /home/jrteixeira/metabase/plugins/presto-jdbc.metabase-driver.ja
2023-09-02 16:17:40,222 INFO util.files :: Extract file /modules/sqlite.metabase-driver.jar -> /home/jrteixeira/metabase/plugins/sqlite.metabase-driver.jar
2023-09-02 16:17:40,593 INFO util.files :: Extract file /modules/vertica.metabase-driver.jar -> /home/jrteixeira/metabase/plugins/vertica.metabase-driver.jar
2023-09-02 16:17:40,596 INFO util.files :: Extract file /modules/redshift.metabase-driver.jar -> /home/jrteixeira/metabase/plugins/redshift.metabase-driver.jar
2023-09-02 16:17:40,640 INFO util.files :: Extract file /modules/druid.metabase-driver.jar -> /home/jrteixeira/metabase/plugins/druid.metabase-driver.jar
2023-09-02 16:17:40,671 INFO util.files :: Extract file /modules/mongo.metabase-driver.jar -> /home/jrteixeira/metabase/plugins/mongo.metabase-driver.jar
2023-09-02 16:17:40,813 INFO util.files :: Extract file /modules/oracle.metabase-driver.jar -> /home/jrteixeira/metabase/plugins/oracle.metabase-driver.jar
2023-09-02 16:17:40,822 INFO util.files :: Extract file /modules/snowflake.metabase-driver.jar -> /home/jrteixeira/metabase/plugins/snowflake.metabase-driver.jar
2023-09-02 16:17:41,840 INFO util.files :: Extract file /modules/googleanalytics.metabase-driver.jar -> /home/jrteixeira/metabase/plugins/googleanalytics.metabase-d
2023-09-02 16:17:41,928 INFO util.files :: Extract file /modules/bigquery-cloud-sdk.metabase-driver.jar -> /home/jrteixeira/metabase/plugins/bigquery-cloud-sdk.meta
2023-09-02 16:17:43,439 DEBUG plugins.lazy-loaded-driver :: Registering lazy loading driver :athena...
2023-09-02 16:17:43,440 INFO driver.impl :: Registered driver :athena (parents: [:sql-jdbc])
```



Metabase: localhost:3000

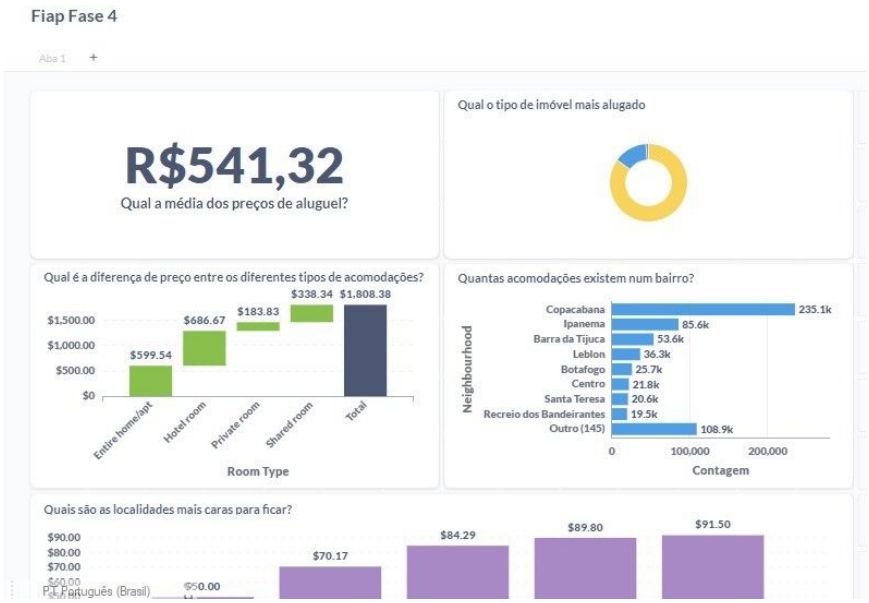


Metabase conectado ao BigQuery

Dicionário das variáveis

- `id` - número de identificação do imóvel
- `name` - Título do anúncio da propriedade
- `host_id` - número de identificação do proprietário
- `host_name` - nome do anfitrião
- `neighbourhood_group` - coluna sem valores válidos
- `neighbourhood` - nome do bairro
- `latitude` - coordenada de latitude da propriedade
- `longitude` - coordenada de longitude da propriedade
- `room_type` - tipo de acomodação oferecida
- `price` - valor do aluguel
- `minimum_nights` - menor quantidade de noites para alugar
- `number_of_reviews` - número de reviews
- `last_review` - data do último review
- `reviews_per_month` - quantidade de reviews em um mês
- `calculated_host_listings_count` - quantidade de imóveis do mesmo anfitrião
- `availability_365` - número de dias de disponibilidade dentro de 365 dias

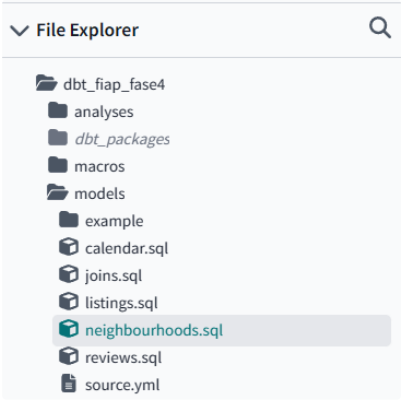
Dashboard desenvolvido:



Camada de Transformação DBT CLOUD

Criamos a conexão entre o Bigquery e o DBT Cloud

configuramos o arquivo source.yml para acessar as bases de dados do bigquery

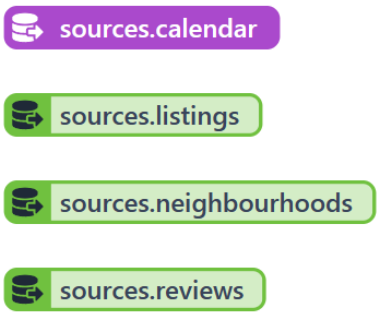


Detalhe da configuração do source:

models > source.yml

```yaml
version: 2

sources:
  - name: sources
    database: dw-fiap-fase4
    schema: fiap_fase4
    tables:
      Generate model
      - name: calendar
      Generate model
      - name: listings
      Generate model
      - name: neighbourhoods
      Generate model
      - name: reviews
```

Lineage:



Create das view no DBT cloud:

Criando a view calc_calendar

models > calendar.sql

```sql
with calc_calendar as (
    select
        *
    from {{source('sources','calendar')}}
)
select * from calc_calendar
```

Criando a view calc_listings

models > listings.sql

```sql
with calc_listings as (
select
*
from {{source('sources','listings')}}
)
select * from calc_listings
```

Criando a view calc_`neighbourhoods`

models > neighbourhoods.sql

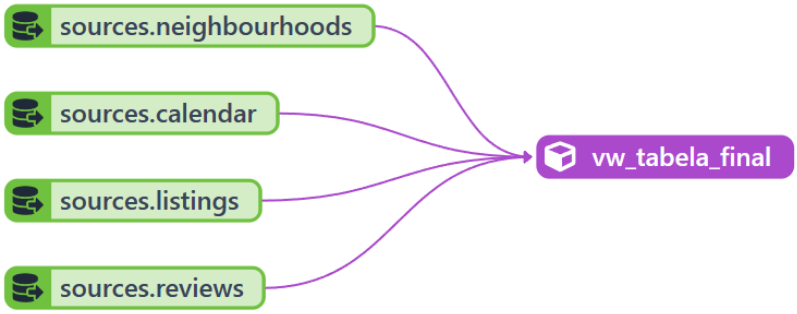
```sql
with calc_neighbourhoods as (
select *
from {{source('sources','neighbourhoods')}}
)
select * from calc_neighbourhoods
```

Criando a view calc_reviews

models > reviews.sql

```sql
with calc_reviews as (
select *
from {{source('sources','reviews')}}
)
select * from calc_reviews
```

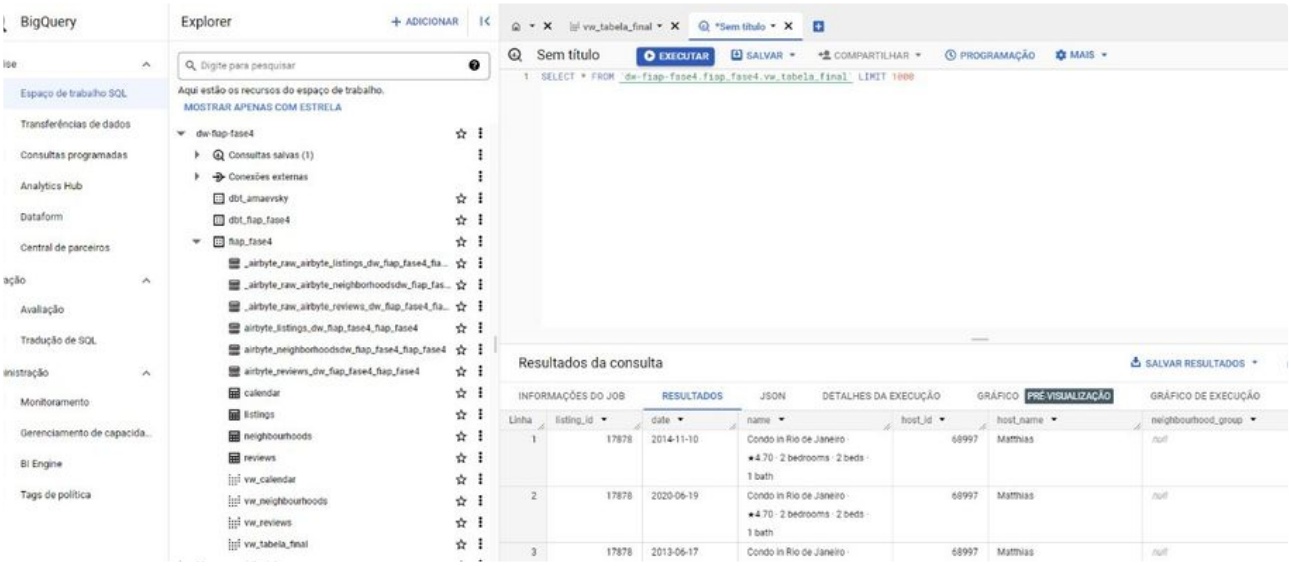
Criando a view final com a união de todas as views



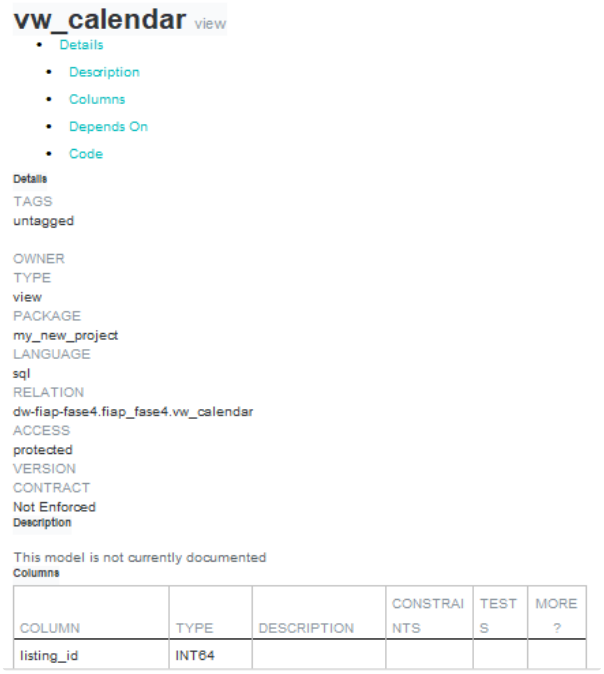
models > joins.sql

```sql
with calc_table_final as (
select
*
from {{source('sources','calendar')}} "calendar"
left join {{source('sources','listings')}} "listings" on ("calendar".listing_id = "listings".id)
left join {{source('sources','neighbourhoods')}} "neighbourhoods" on ("neighbourhoods".neighbourhood_group = "listings".neighbourho
left join {{source('sources','reviews')}} "reviews" on ("reviews".listing_id = "listings".id)
)
select * from calc_table_final
```

Gravando os dados da view do DBT no Bigquery



Uma solução para governança e catálogo de dados:



vw_calendar view

- Details
  - Description
  - Columns
  - Depends On
  - Code

Details

TAGS
untagged

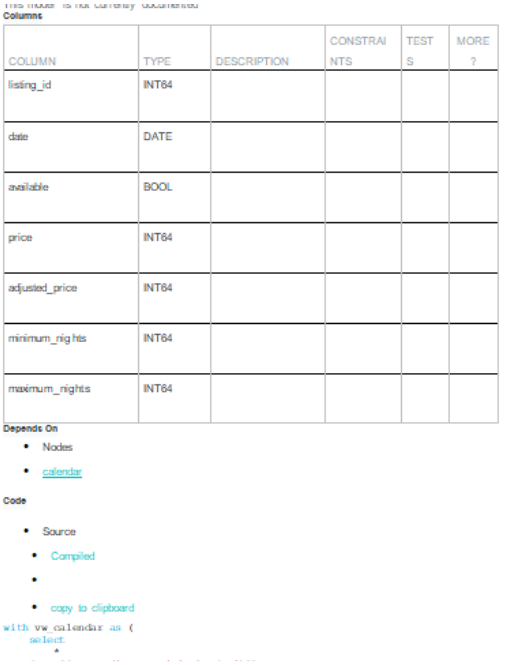
OWNER
TYPE
view
PACKAGE
my_new_project
LANGUAGE
sql
RELATION
dw-fiap-fase4.fiap_fase4.vw_calendar
ACCESS
protected
VERSION
CONTRACT
Not Enforced

Description

This model is not currently documented

Columns

COLUMN	TYPE	DESCRIPTION	CONSTRAINTS	TESTS	MORE?
listing_id	INT64				



Columns

COLUMN	TYPE	DESCRIPTION	CONSTRAINTS	TESTS	MORE?
listing_id	INT64				
date	DATE				
available	BOOL				
price	INT64				
adjusted_price	INT64				
minimum_nights	INT64				
maximum_nights	INT64				

Depends On

- Nodes
- calendar

Code

- Source
- Compiled
- copy to clipboard

```
with vw_calendar as (
    select
```

Entregável 2

Detalhes dos datasets:

Link Inside Airbnb

Análises, relatórios ou dashboards para o negócio respondendo as perguntas relacionando dados do Airbnb correlacionando com data set obtido sobre imóveis:

Entregável 3

Detalhes da instalação e configuração realizada para implementação da arquitetura sugerida para solução

- Usamos WSL para simular um servidor Ubunto com Airflow configurado;
- Instalamos Docker para simular o seridor com Airbyte;
- Baixamos imagem Airflow via Terminal (Airflow >> Evidência) ;
- Baixamos os arquivos para pasta local e depois subimos para o Google Drive via script Python;
- clonamos repositório AirByte para acessá-lo via Docker (AirByte >> Evidência);
- Criamos as tabelas usando os conectores no Airbyte (GCS e BigQuery) (Evidências BigQuery);
- Acessamos link da página: Metabase | Business Intelligence, Dashboards, and Data Visualization
  Em Documentation >> Installation and operation overview >> Running the Metabase JAR file >> Douwnload
- Iniciamos procedimento para instalação no Ubunto WSL (ver arquivo instalacao_metabase.txt) (Evidências Metabase)

Script(s)/comandos utilizados na transformação de dados, modelo(s) de Machine Learning e qualquer outra etapa adicional/opcional à arquitetura

download_files.py
15 jul. 2023, 10:56 PM

script_config_airflow.txt
20 ago. 2023, 09:54 PM

instalacao_metabase.txt
02 set. 2023, 06:34 PM

Definição detalhada da modelagem de dados