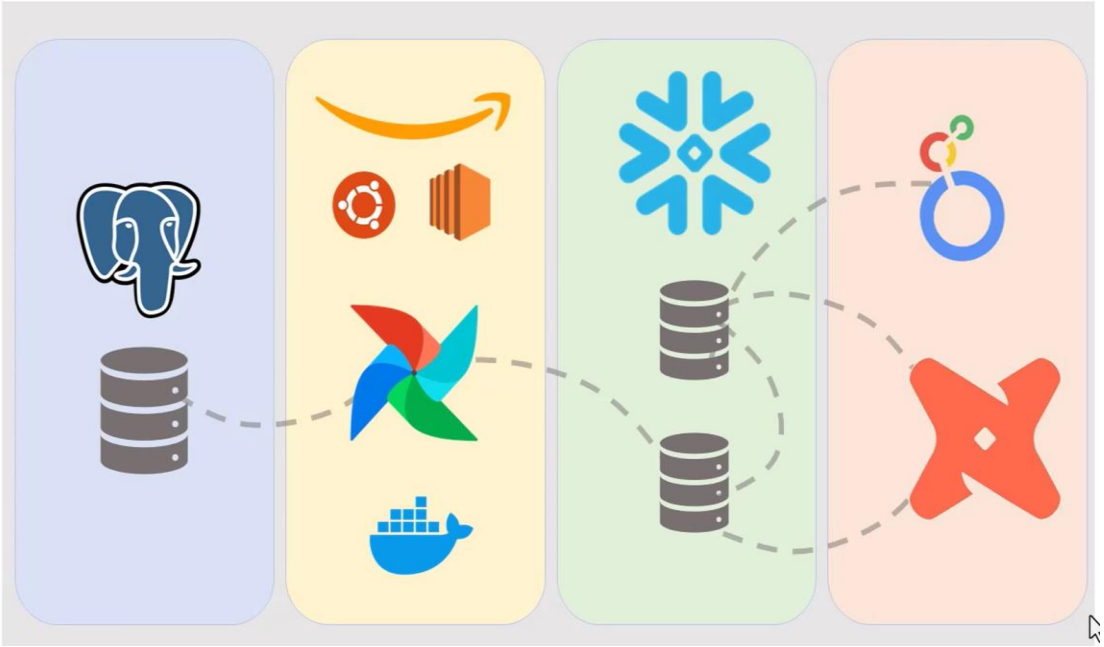
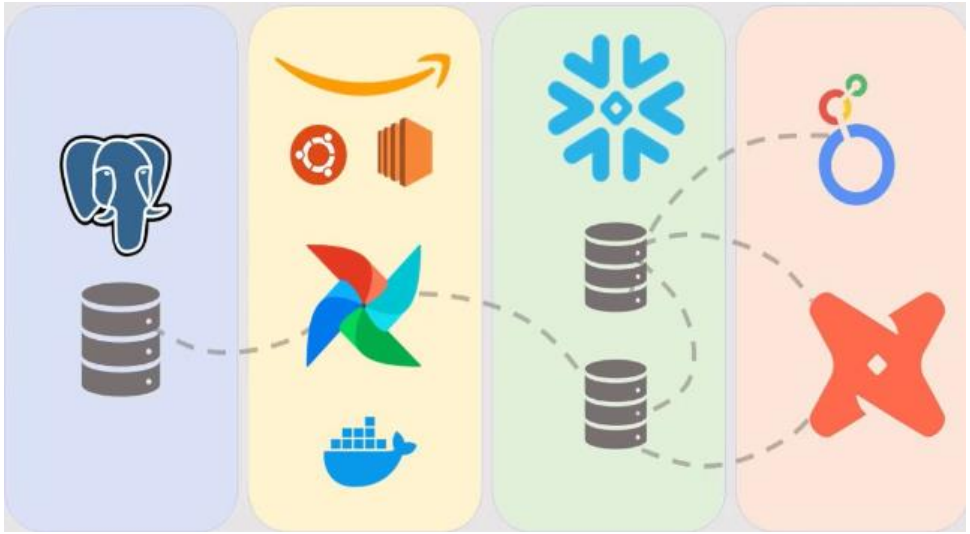
Pipeline de ELT com DBT



Projeto: Analitycs Engineer (vendas)

Linguagem: Python
BD: PostgreSQL
Cloud: AWS EC2
SO: Linux
Container: Docker
Orquestrador: Airflow
ETL: dbt
DW: Snowflake
Dataviz: Looker Studio



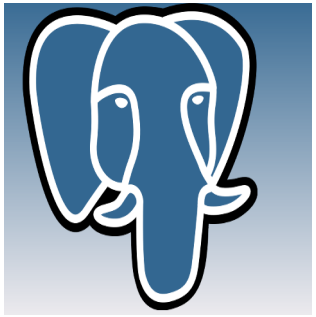
Etapas

1. Conectar e explorar o BD de vendas;
2. Criar conta AWS ;
3. Criar VM Linux (EC2);
4. Instalar e configurar o Docker;
5. Instalar e configurar o Apache Airflow (stage e teste de carga);
6. Configurar o Snowflake (criar BD, schemas, WH, Tabelas e etc)
7. Configurar DBT (criar modelos, jobs, testes);
8. Criar dashboards no Looker Studio;

Etapa 1 - Conectar e explorar o BD de vendas;

```
/home/renatopatricio/.hushlogin file.
renatopatricio@Renato:~$ psql -h 159.223.187.110 -U etlreadonly -d novadrive
Password for user etlreadonly:
psql (14.11 (Ubuntu 14.11-0ubuntu0.22.04.1), server 12.18 (Ubuntu 12.18-0ubuntu0.20.04.1))
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, bits: 256, compression: off)
Type "help" for help.

novadrive=> SELECT schema_name FROM information_schema.schemata;
    schema_name
--------------------
 pg_catalog
 public
 information_schema
(3 rows)
```

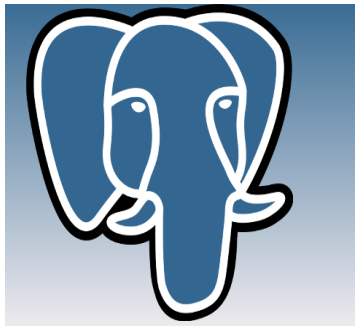

```
novadrive=> SELECT
    pg_database.datname,
    pg_size_pretty(pg_database_size(pg_database.datname)) AS size
    FROM pg_database;
    datname     |  size
----------------+---------
 postgres       | 51 MB
 template1      | 7953 kB
 template0      | 7809 kB
 novadrive      | 22 MB
 novadrivebank | 8569 kB
(5 rows)
```



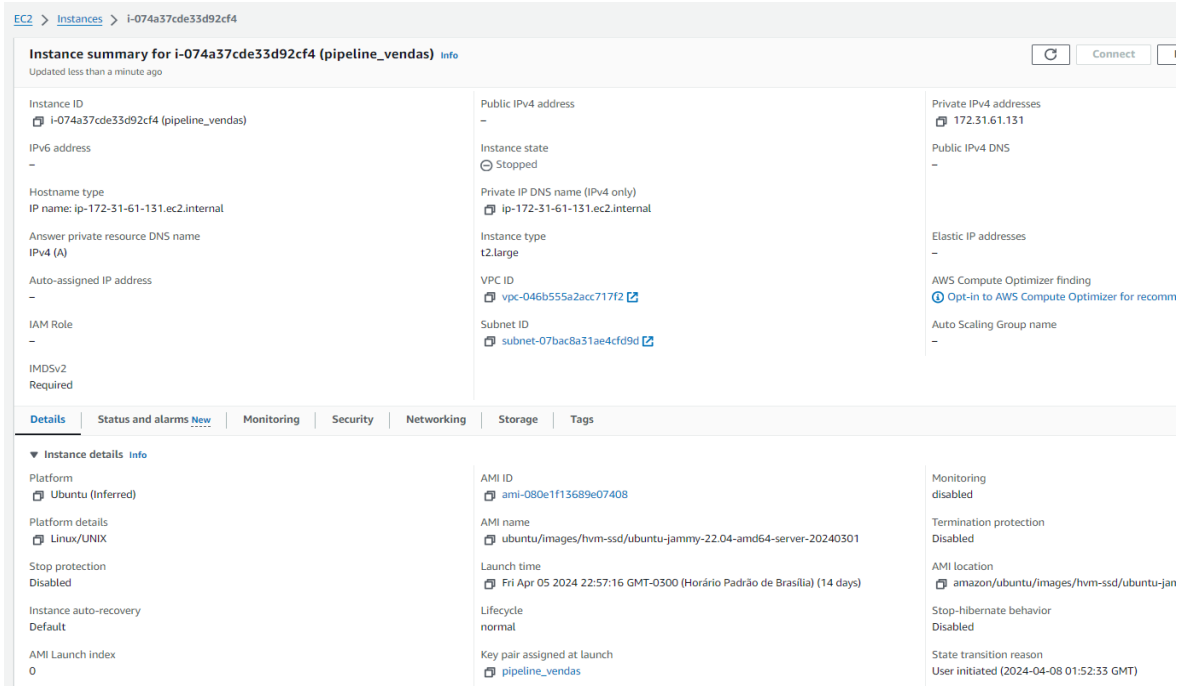
Etapa 1 - Conectar e explorar o BD de vendas;

```
novadrive=> \dt+
                                    List of relations
 Schema |      Name       | Type  |  Owner   | Persistence | Access method |    Size    | Description
--------+-----------------+-------+----------+-------------+---------------+------------+-------------
 public | cidades         | table | postgres | permanent   | heap          | 8192 bytes |
 public | clientes        | table | postgres | permanent   | heap          | 6208 kB    |
 public | concessionarias | table | postgres | permanent   | heap          | 8192 bytes |
 public | estados         | table | postgres | permanent   | heap          | 8192 bytes |
 public | veiculos        | table | postgres | permanent   | heap          | 8192 bytes |
 public | vendas          | table | postgres | permanent   | heap          | 4736 kB    |
 public | vendedores      | table | postgres | permanent   | heap          | 8192 bytes |
(7 rows)
```

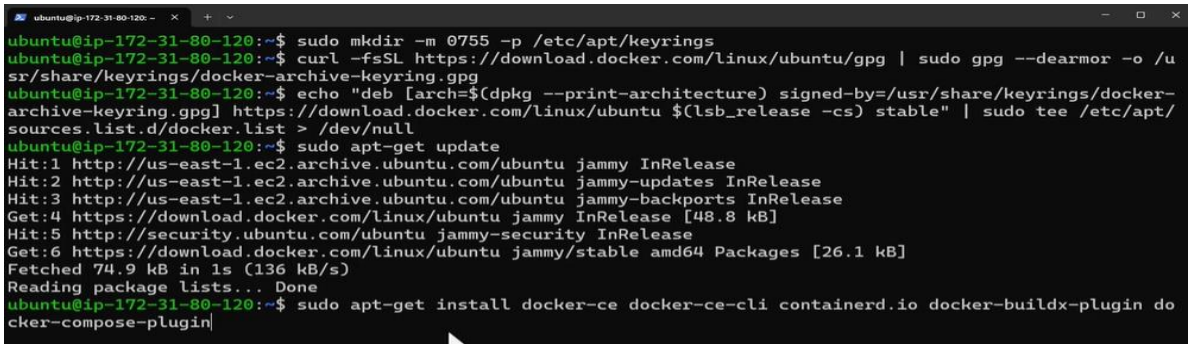
```
novadrive=> SELECT sum(pg_relation_size(c.oid)) AS total_size
FROM pg_class c
WHERE c.relname IN ('cidades', 'clientes', 'concessionarias', 'estados', 've
iculos', 'vendas', 'vendedores');
 total_size
------------
   11173888
(1 row)
```

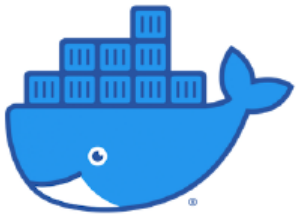


Criação e configuração da VM/Linux

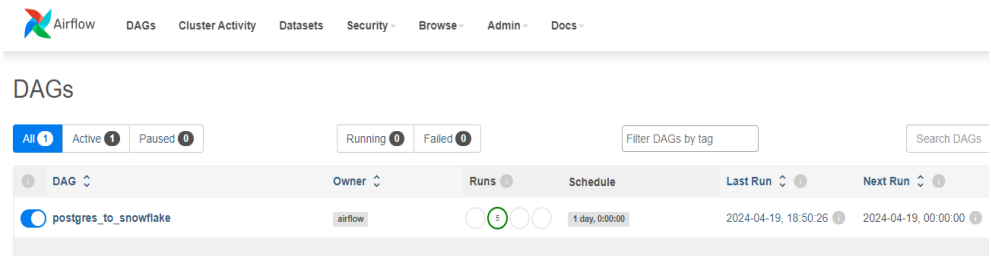


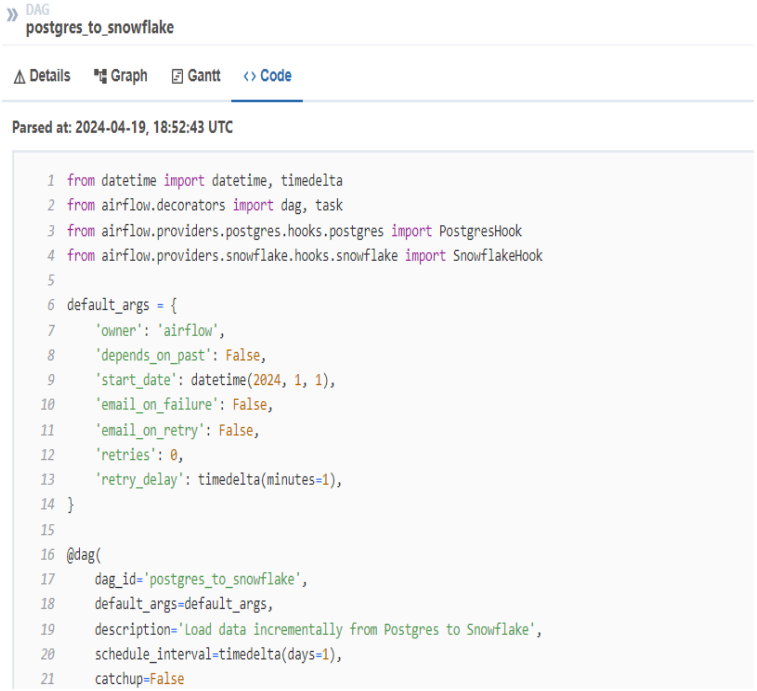
Instalação e configuração Docker/Airflow





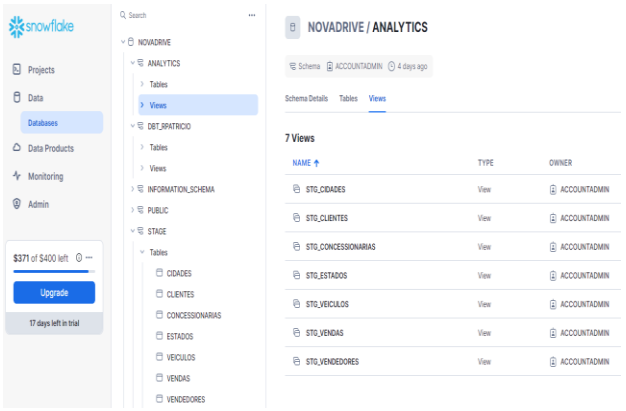
Criação da DAG(Airflow)

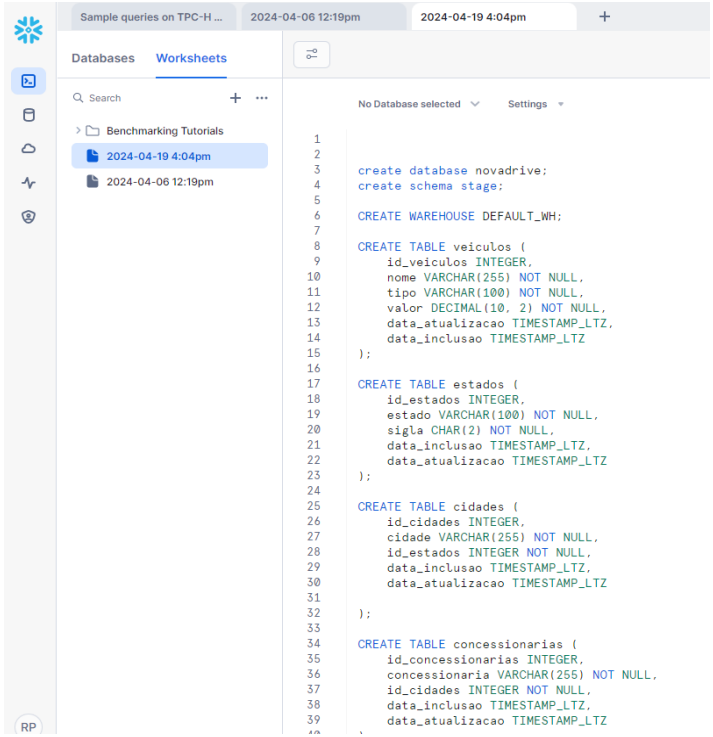




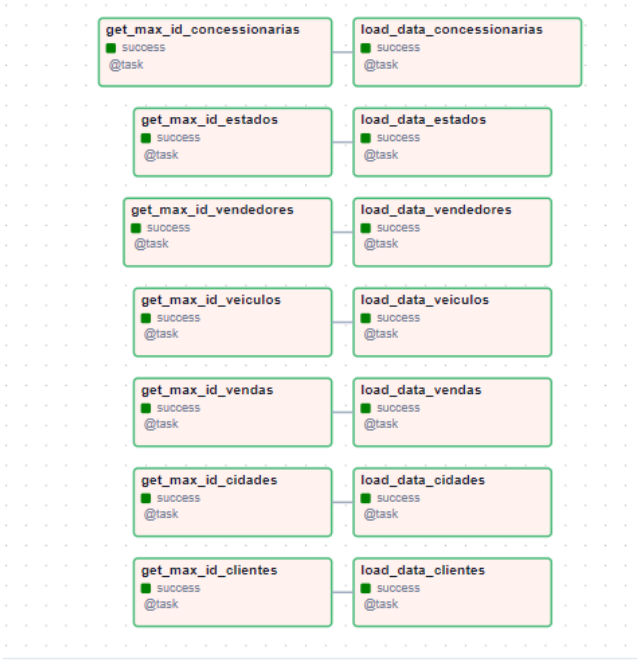

```python
from datetime import datetime, timedelta
from airflow.decorators import dag, task
from airflow.providers.postgres.hooks.postgres import PostgresHook
from airflow.providers.snowflake.hooks.snowflake import SnowflakeHook

default_args = {
    'owner': 'airflow',
    'depends_on_past': False,
    'start_date': datetime(2024, 1, 1),
    'email_on_failure': False,
    'email_on_retry': False,
    'retries': 0,
    'retry_delay': timedelta(minutes=1),
}

@dag(
    dag_id='postgres_to_snowflake',
    default_args=default_args,
    description='Load data incrementally from Postgres to Snowflake',
    schedule_interval=timedelta(days=1),
    catchup=False
```

Configuração do Snowflake





Testando a DAG (Airlfow)



Testando a carga incremental (Snowflake)

ACCOUNTADMIN

NOVADRIVE.STAGE Settings

```
1  SHOW TABLES IN stage;
2
3
4
5
```

Results Chart

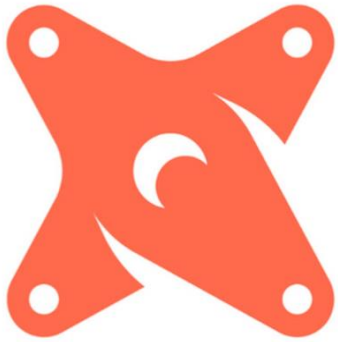
	created_on	name	database_name	schema_name	kind	comment	cluster_by	rows	bytes	owner
1	2024-04-06 08:29:09.768 -0700	CIDADES	NOVADRIVE	STAGE	TABLE			54	4608	ACCOUNT
2	2024-04-06 08:29:10.992 -0700	CLIENTES	NOVADRIVE	STAGE	TABLE			10862	317440	ACCOUNT
3	2024-04-06 08:29:10.138 -0700	CONCESSIONARIAS	NOVADRIVE	STAGE	TABLE			58	4608	ACCOUNT
4	2024-04-06 08:29:09.325 -0700	ESTADOS	NOVADRIVE	STAGE	TABLE			54	4608	ACCOUNT
5	2024-04-06 08:29:08.851 -0700	VEICULOS	NOVADRIVE	STAGE	TABLE			16	4608	ACCOUNT
6	2024-04-06 08:29:11.446 -0700	VENDAS	NOVADRIVE	STAGE	TABLE			10897	262144	ACCOUNT
7	2024-04-06 08:29:10.564 -0700	VENDEDORES	NOVADRIVE	STAGE	TABLE			106	3072	ACCOUNT

Criação do arquivo source.yml (dbt)

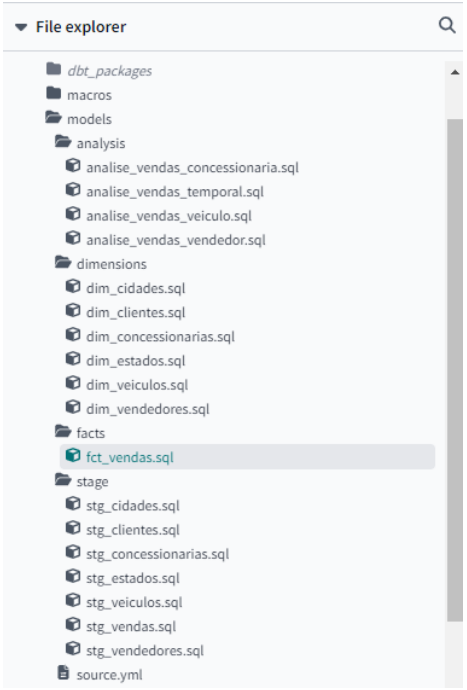
source.yml

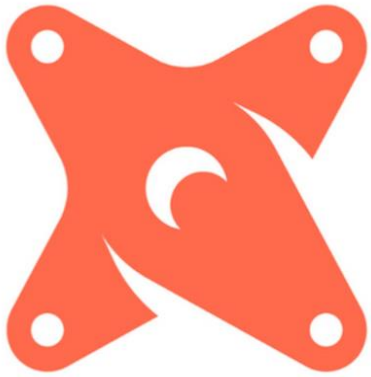
models > source.yml

```yaml
version: 2

sources:
  - name: sources
    database: NOVADRIVE
    schema: STAGE
    tables:
      Generate model
      - name: cidades
      Generate model
      - name: clientes
      Generate model
      - name: concessionarias
      Generate model
      - name: estados
      Generate model
      - name: veiculos
      Generate model
      - name: vendas
      Generate model
      - name: vendedores
```

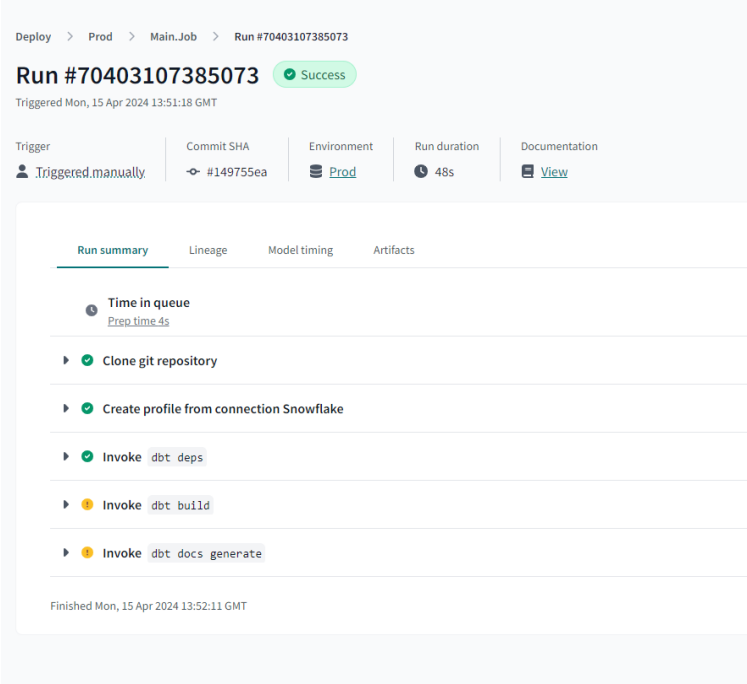


Criação dos modelos de consulta (dbt)





Deploy do job de produção (dbt)



Dashboard das análises (Looker Studio)