FINAL PROJECT DATA ENGINEER

END TO END PIPELINE – DE BATCH 5

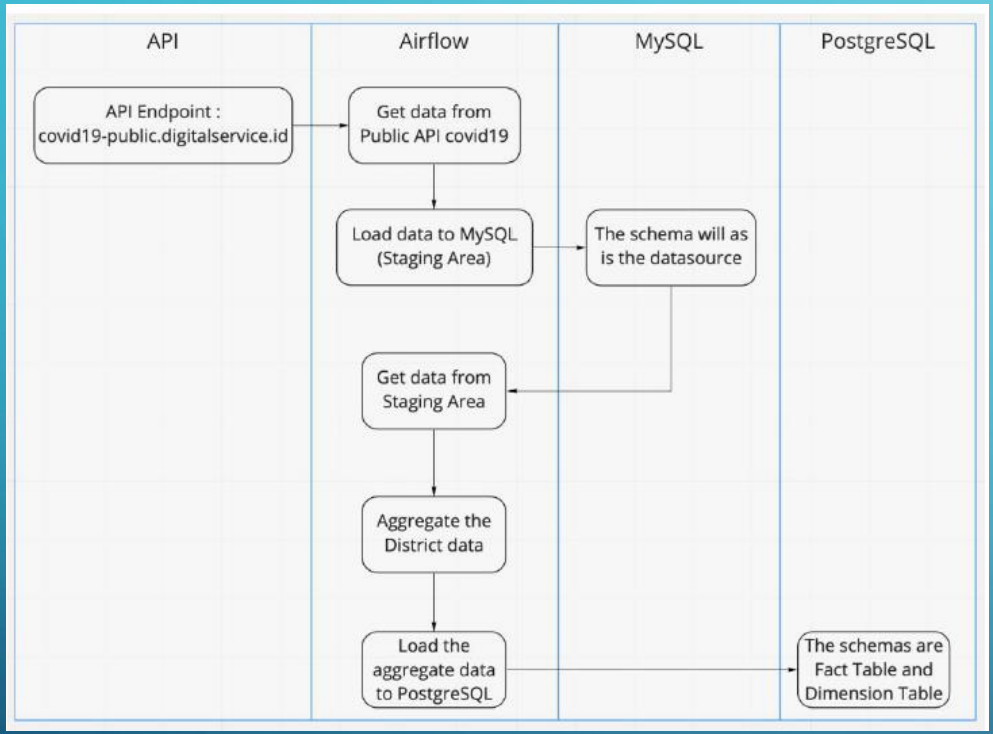
ARDHANI RAHMADIANTO

INDEX

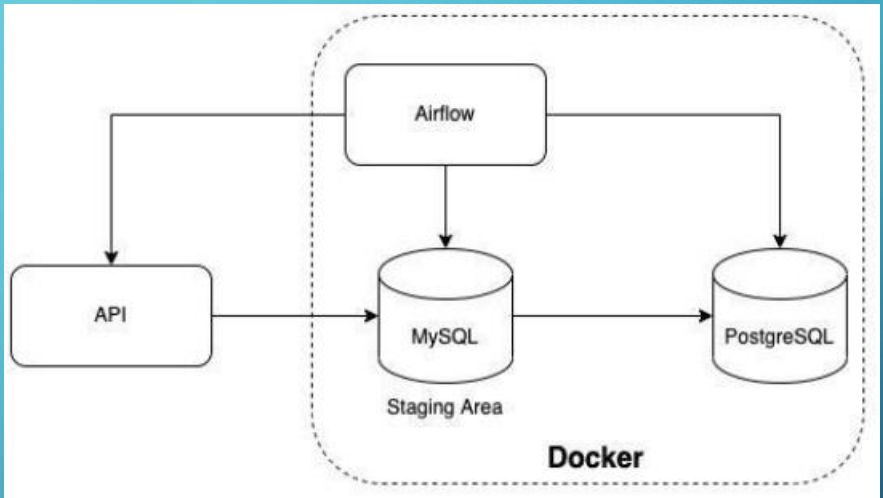
- Project Briefing
- Setup Environment (Airflow, Postgres, MySQL using Docker Compose)
- Setting Airflow Configuration
- DAG File Explanation (Python, SQL, etc.)
- Running Airflow DAG
- ERD & Table Result

PROJECT BRIEFING

Workflow



ETL Architecture



PROJECT BRIEFING

API


```
{
 "status_code": 200,
 "data": {
  "metadata": {
   "last_update": null
  },
  "content": [
   {
    "tanggal": "2020-08-05",
    "kode_prov": "32",
    "nama_prov": "Jawa Barat",
    "kode_kab": "3204",
    "nama_kab": "Kabupaten Bandung",
    "SUSPECT": 2210,
    "CLOSECONTACT": 274,
    "PROBABLE": 26,
    "suspect_diisolasi": 31,
    "suspect_discarded": 2179,
    "closecontact_dikarantina": 0,
    "closecontact_discarded": 274,
    "probable_diisolasi": 0,
    "probable_discarded": 0,
    "CONFIRMATION": 0,
    "confirmation_sembuh": 0,
    "confirmation_meninggal": 0,
    "suspect_meninggal": 0,
    "closecontact_meninggal": 0,
    "probable_meninggal": 26
   }
  ]
 }
}
```


Table Result Expectation

Dimension table

1. Province table
   a. province_id
   b. province_name
2. District table
   a. district_id
   b. province_id
   c. district_name
3. Case table
   a. Id
   b. Status name (suspect, closecontact, probable, confirmation)
   c. Status detail

Fact table

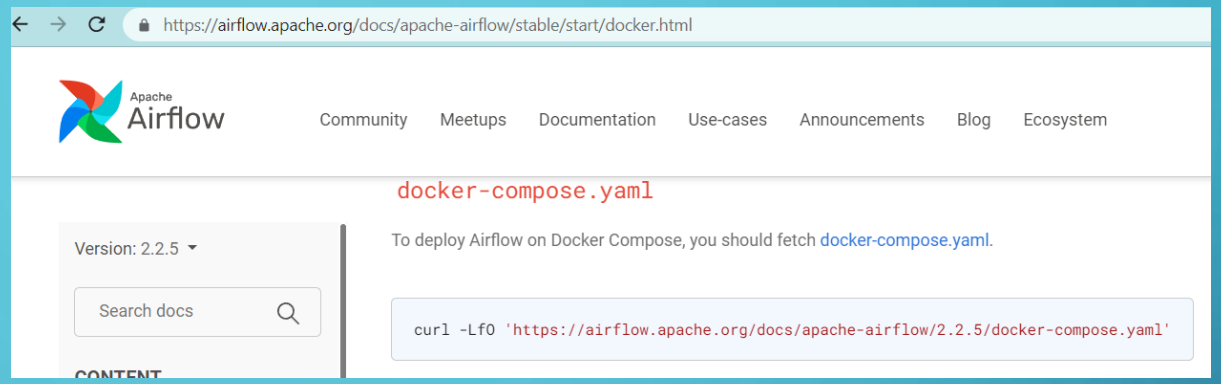
1. Province Daily Table
   a. Id (auto generate)
   b. province_id
   c. case_id
   d. date
   e. total
2. Province Monthly Table
   a. Id (auto generate)
   b. province_id
   c. case_id
   d. month
   e. total
3. Province Yearly Table
   a. Id (auto generate)
   b. province_id
   c. case_id
   d. year
   e. total
4. District Monthly Table
   a. Id (auto generate)
   b. district_id
   c. case_id
   d. month
   e. total
5. District Yearly Table
   a. Id (auto generate)
   b. district_id
   c. case_id
   d. year
   e. total

SETUP DOCKER ENVIRONMENT

Step 1

Get docker-compose.yml from

https://airflow.apache.org/docs/apache-airflow/stable/start/docker.html



Step 2

Modify the .yml file, add the postgres & mysql services and network

Makesure all the services within same networks

```yaml
postgres_data:
  image: postgres:13
  environment:
    POSTGRES_USER: ardhani
    POSTGRES_PASSWORD: ardhani
    POSTGRES_DB: tugas_akhir_de
  ports:
    - 5454:5432
  networks:
    - de_network

mysql_staging:
  image: mysql:8.0.28-debian
  environment:
    MYSQL_USER: ardhani
    MYSQL_PASSWORD: ardhani
    MYSQL_ROOT_PASSWORD: ardhani
    MYSQL_DATABASE: tugas_akhir_de
  ports:
    - 3366:3306
  networks:
    - de_network
```

Modification in services

```yaml
      condition: service_completed_successfully
    networks:
      - de_network

volumes:
  postgres-db-volume:

networks:
  de_network:
    name: de_network
```

Modification in networks

SETUP DOCKER ENVIRONMENT

Step 3

Setup Airflow Environment by execute command below

```
mkdir -p ./dags ./logs ./plugins
echo -e "AIRFLOW_UID=$(id -u)" > .env
```

Step 4

Setup Airflow Environment by execute command below

```
docker-compose up airflow-init
```

After initialization is complete, you should see a message like below.

```
airflow-init_1       | Upgrades done
airflow-init_1       | Admin user airflow created
airflow-init_1       | 2.2.5
start_airflow-init_1 exited with code 0
```

The account created has the login `airflow` and the password `airflow`.

Step 5

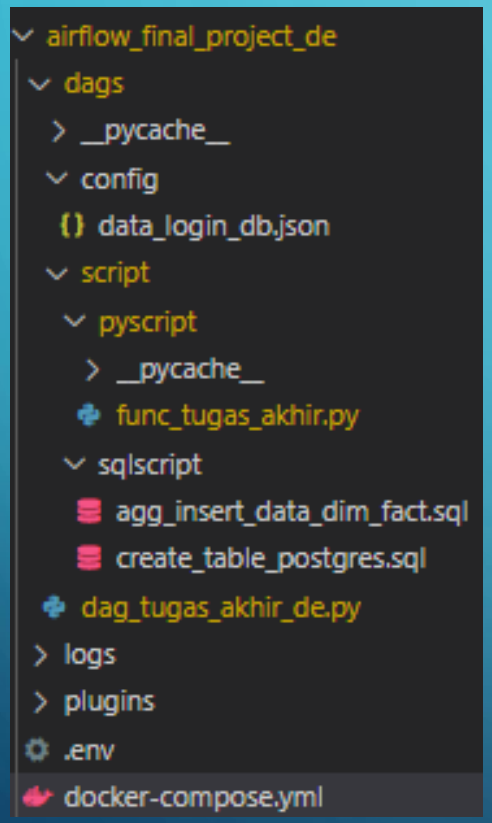
Running Aiflow

```
docker-compose up
```

SETTING AIRFLOW CONFIGURATION

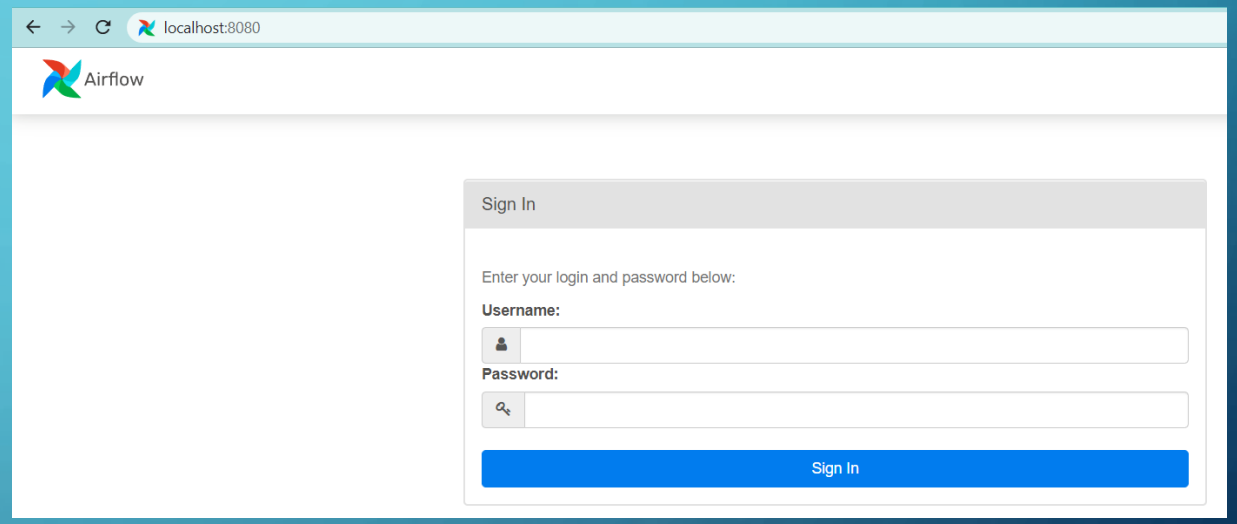
Step 1

Make sure the files and directory is like below



Step 2

Access Airflow UI using browser



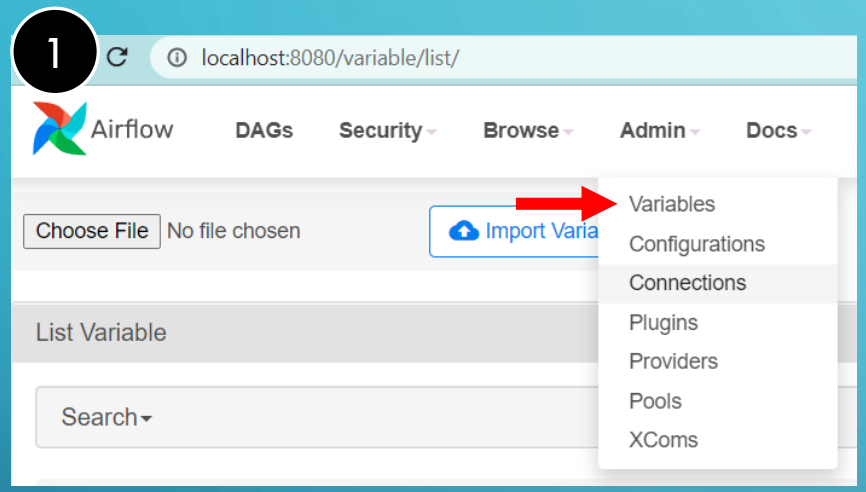
Username : airflow

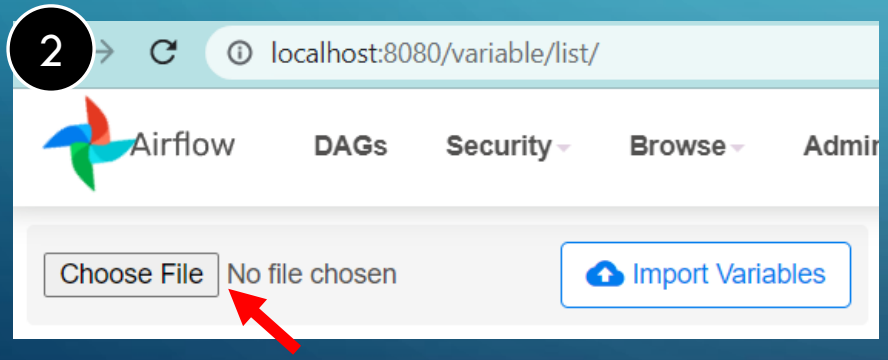
Password : airflow

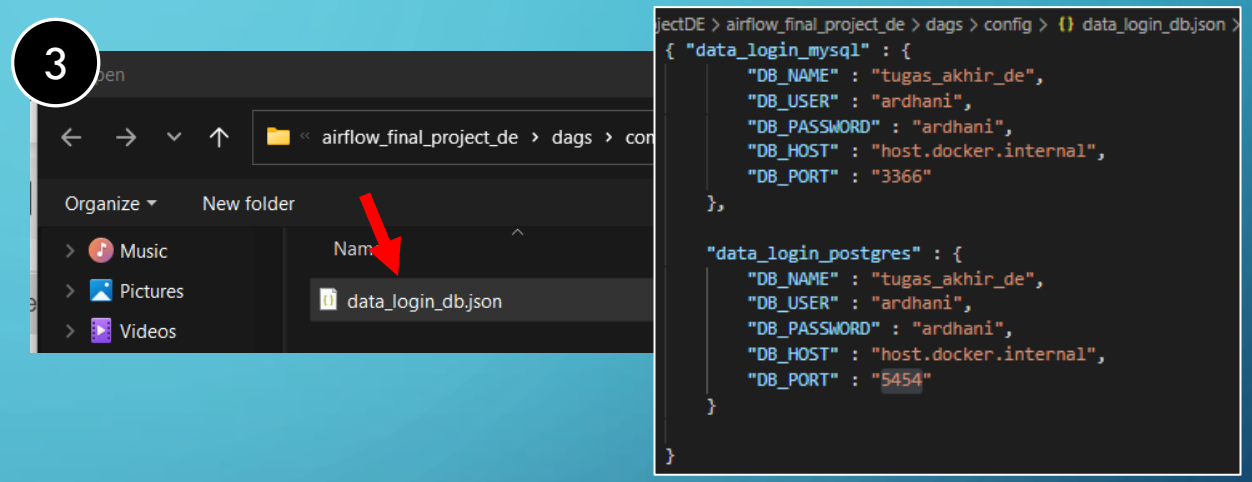
SETTING AIRFLOW CONFIGURATION

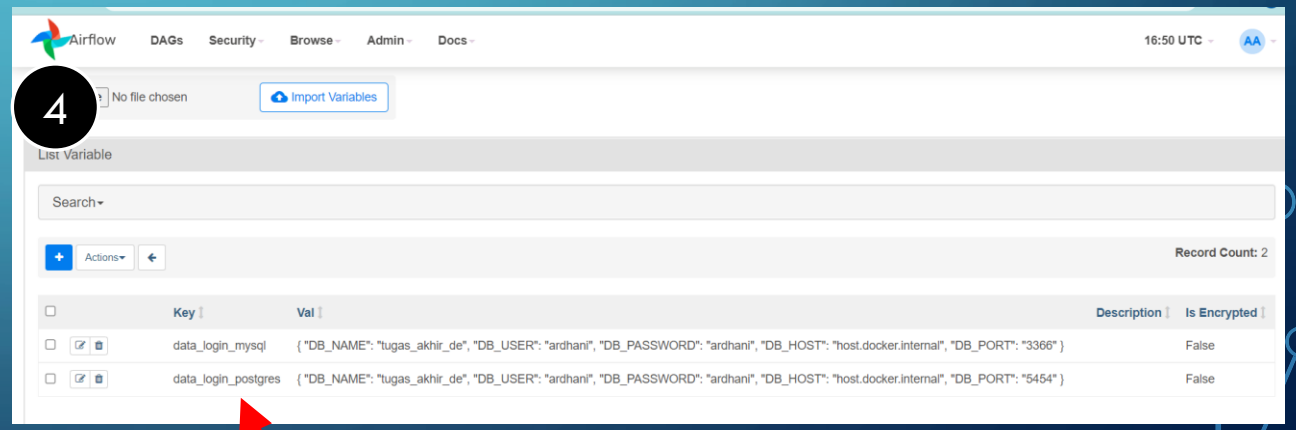
Step 3

Upload “data_login_db.json” to setup Aiflow Variables







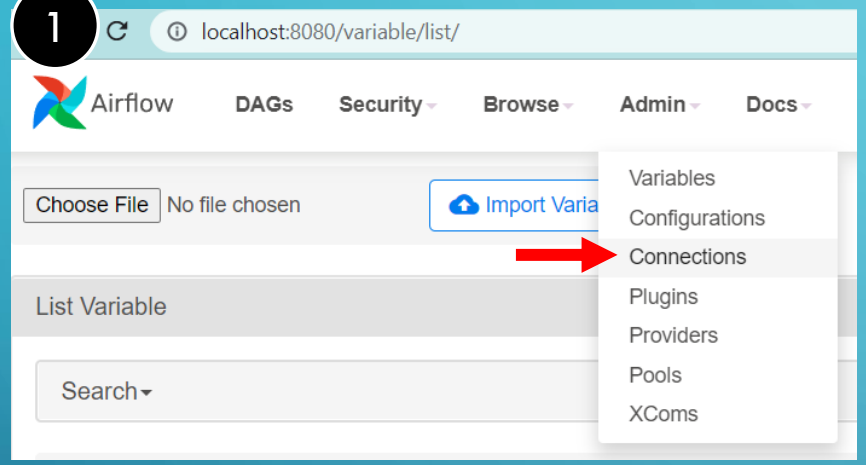


SETTING AIRFLOW CONFIGURATION

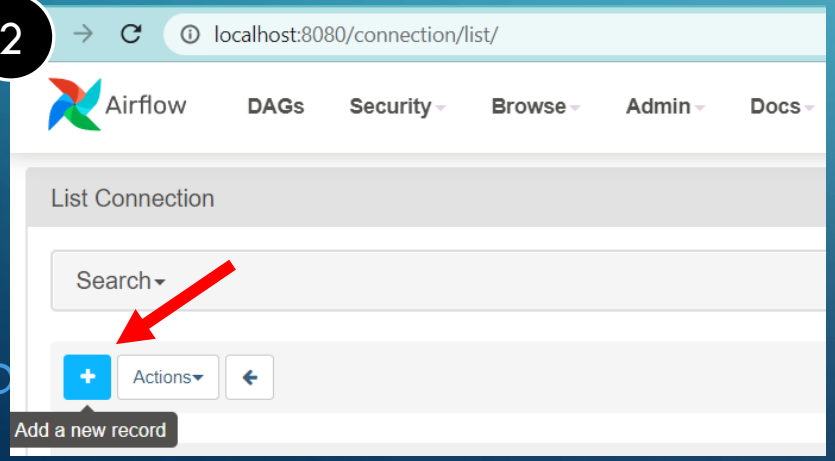
Step 3

Setup Airflow Connection to Postgres DB connection

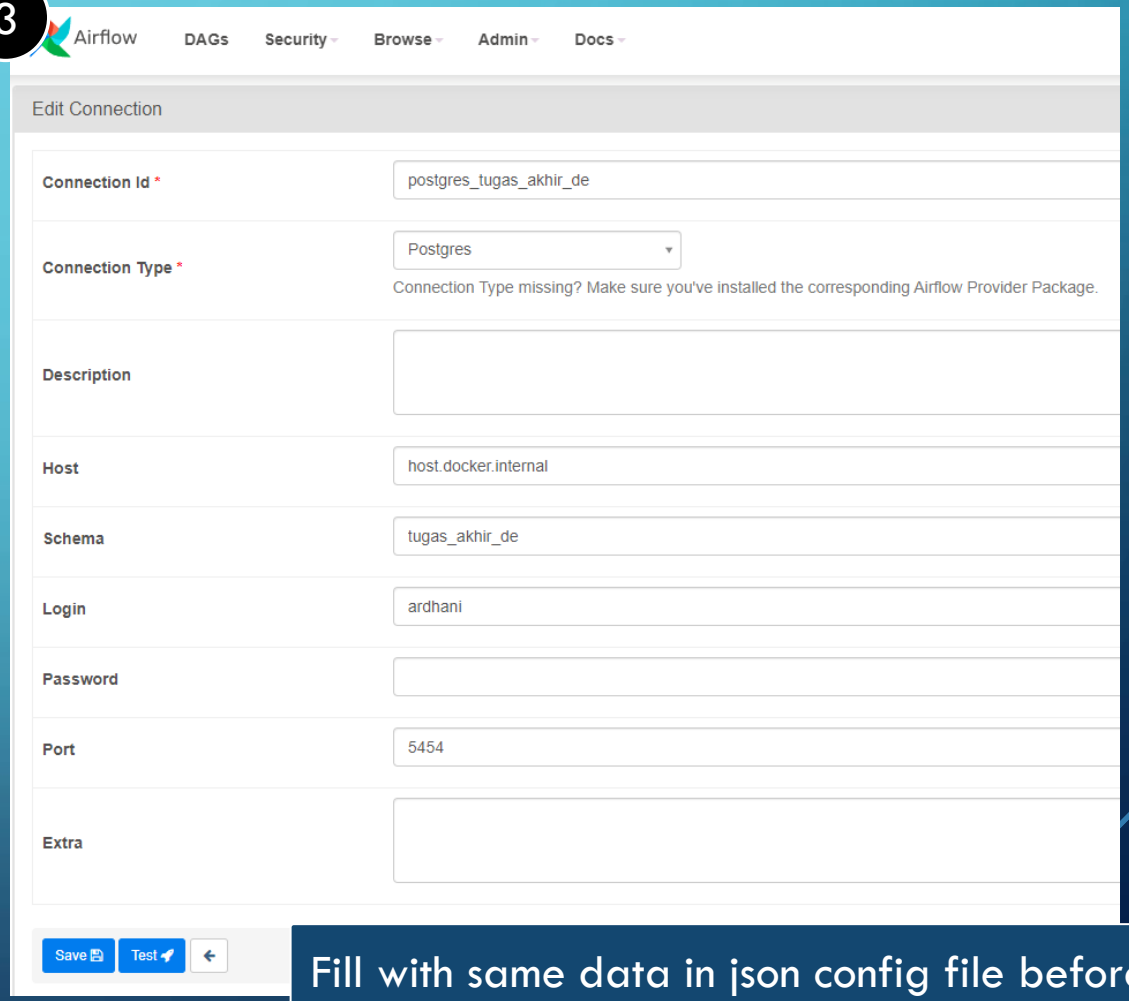
1



2



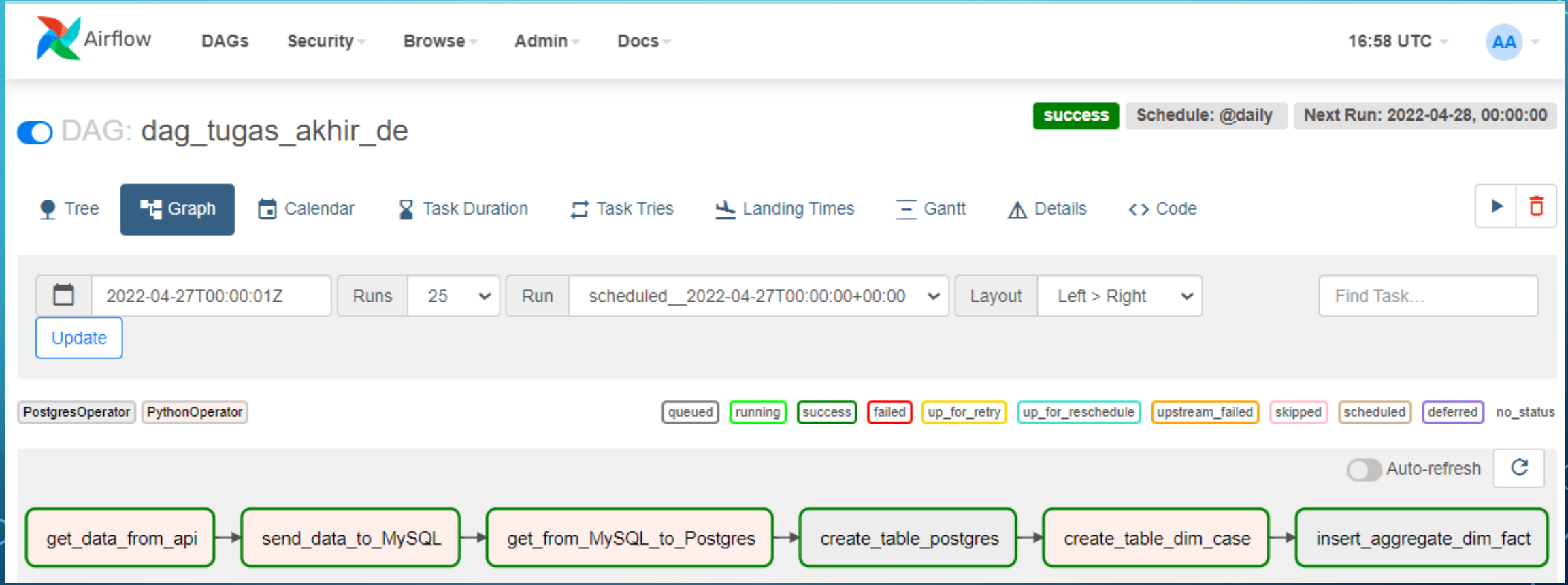
3



Fill with same data in json config file before

DAG FILE EXPLANATION

Workflow



DAG FILE EXPLANATION

Operators in DAG

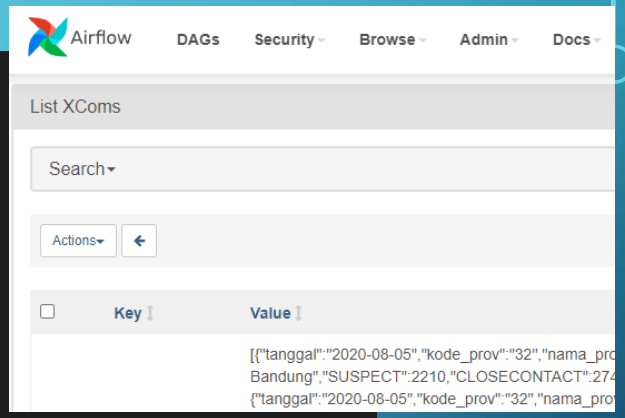
```python
with DAG(
    dag_id='dag_tugas_akhir_de',
    schedule_interval='@daily',
    start_date=datetime(2022,4,25),
    catchup= False
) as dag:
    get_data_from_api = PythonOperator(
        task_id = 'get_data_from_api',
        python_callable = func_tugas_akhir.get_data_from_api,
        op_kwargs = {"con_conf" : config_MySQL , "url_api" : 'https://covid19-public.digitalservice.id/api/v1/rekapitulasi_v2/jabar/harian?level=kab'}
    )

    send_data_to_MySQL = PythonOperator(
        task_id = 'send_data_to_MySQL',
        python_callable = func_tugas_akhir.send_data_MySQL,
        op_kwargs = {"con_conf" : config_MySQL , "url_api" : 'https://covid19-public.digitalservice.id/api/v1/rekapitulasi_v2/jabar/harian?level=kab'}
    )

    get_from_MySQL_to_Postgres = PythonOperator(
        task_id = 'get_from_MySQL_to_Postgres',
        python_callable = func_tugas_akhir.get_from_MySQL_to_Postgres,
        op_kwargs = {"con_conf_mysql" : config_MySQL,"con_conf_postgres" : config_Postgres}
    )

    create_table_postgres = PostgresOperator(
        task_id="create_table_postgres",
        postgres_conn_id="postgres_tugas_akhir_de",
        sql="script/sqlscript/create_table_postgres.sql"
    )

    create_table_dim_case = PythonOperator(
        task_id = 'create_table_dim_case',
        python_callable = func_tugas_akhir.create_table_dim_case,
        op_kwargs = {"con_conf_postgres" : config_Postgres}
    )

    insert_aggregate_dim_fact = PostgresOperator(
        task_id="insert_aggregate_dim_fact",
        postgres_conn_id="postgres_tugas_akhir_de",
        sql="script/sqlscript/agg_insert_data_dim_fact.sql"
    )
```

```python
from airflow.models import Variable

from script.pyscript import func_tugas_akhir

config_MySQL = Variable.get("data_login_mysql",deserialize_json = True)
config_Postgres = Variable.get("data_login_postgres",deserialize_json = True)
```

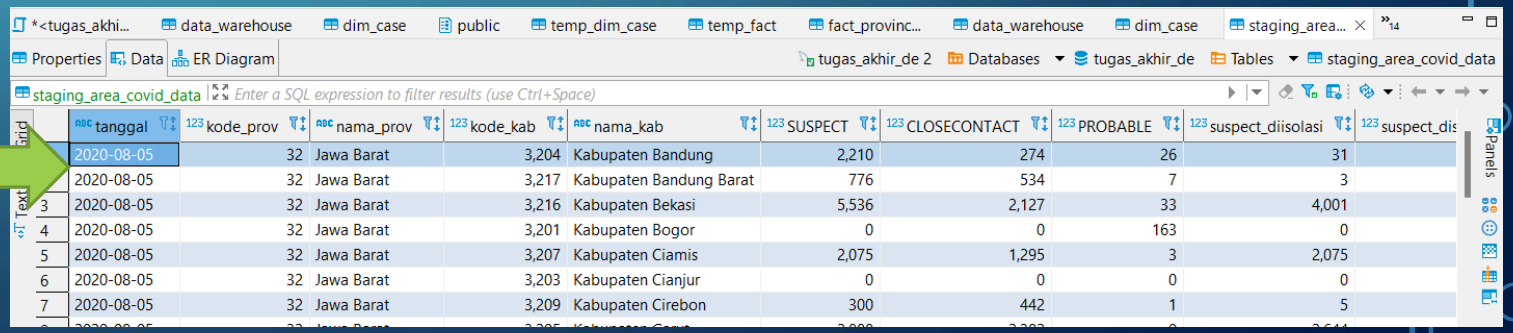
DAG FILE EXPLANATION

Python Function Used

```python
def get_data_from_api(url_api,ti):

    df_json_raw = pd.read_json(url_api)
    df_json = pd.DataFrame(df_json_raw.loc['content']['data'])
    df_for_xcom = df_json.to_json(orient = 'records')
    print("df_for_xcom type : ", type(df_for_xcom))
    ti.xcom_push(key='data_covid_jabar',value=df_for_xcom)

def send_data_MySQL(con_conf,ti):
    data_xcom = ti.xcom_pull(key='data_covid_jabar')#, task_ids = 'get_data_from_api')
    # print("data_xcom type : ", type(data_xcom))
    data_covid_jabar = pd.read_json(data_xcom, orient = 'records')
    # print("data_covid_jabar type : ", type(data_covid_jabar))

    CONNECTION_STRING = f"""mysql+mysqldb://{con_conf["DB_USER"]}:{con_conf["DB_PASSWORD"]}@{con_conf["DB_HOST"]}:{con_conf["DB_PORT"]}/{con_conf["DB_NAME"]}"""

    engine = create_engine(CONNECTION_STRING)
    if not database_exists(engine.url):
        create_database(engine.url)

    print("GET DATA API : Connection to MySQL : ",database_exists(engine.url))
    data_covid_jabar.to_sql(name = "staging_area_covid_data", con=engine, if_exists="replace", index=False)
```



	tanggal	kode_prov	nama_prov	kode_kab	nama_kab	SUSPECT	CLOSECONTACT	PROBA
0	2020-08-05	32	Jawa Barat	3204	Kabupaten Bandung	2210	274	
1	2020-08-05	32	Jawa Barat	3217	Kabupaten Bandung Barat	776		
2	2020-08-05	32	Jawa Barat	3216	Kabupaten Bekasi	5536	2127	
3	2020-	32	Jawa Barat	3201	Kabupaten	0	0	



	tanggal	kode_prov	nama_prov	kode_kab	nama_kab	SUSPECT	CLOSECONTACT	PROBABLE	suspect_diisolasi	suspect_dis
	2020-08-05	32	Jawa Barat	3,204	Kabupaten Bandung	2,210	274	26	31	
	2020-08-05	32	Jawa Barat	3,217	Kabupaten Bandung Barat	776	534	7	3	
3	2020-08-05	32	Jawa Barat	3,216	Kabupaten Bekasi	5,536	2,127	33	4,001	
4	2020-08-05	32	Jawa Barat	3,201	Kabupaten Bogor	0	0	163	0	
5	2020-08-05	32	Jawa Barat	3,207	Kabupaten Ciamis	2,075	1,295	3	2,075	
6	2020-08-05	32	Jawa Barat	3,203	Kabupaten Cianjur	0	0	0	0	
7	2020-08-05	32	Jawa Barat	3,209	Kabupaten Cirebon	300	442	1	5	

DAG FILE EXPLANATION

SQL Used

create_table_postgres.sql

FinalProjectDE > airflow_final_project_de > dags > script > sqlscript > create_table_postgres.sql

```sql
--Table Creation DDL
--DIM Table
-- 1.Province Table
create table if not exists dim_province(
    province_id text
    , province_name text
    , primary key(province_id)
);
-- 2.District Table
create table if not exists dim_district(
    district_id text
    , province_id text
    , district_name text
    , primary key(district_id)
    , foreign key(province_id) references dim_province(province_id)
);
-- 3.Case Table
create table if not exists dim_case(
    case_id serial
    , status_name text -- SUSPECT, CLOSECONTACT, PROBABLE, CONFIRMATION
    , status_detail text -- suspect_diisolasi, suspect_discarded, closecontact_dikarantina, closec
    , primary key(case_id)
);
--Fact Table
--1. Province Daily table
create table if not exists fact_province_daily(
    id serial
    , province_id text
    , case_id int
```

agg_insert_data_dim_fact.sql | data_login_db.json | docker-compose.yml | func_tugas_akhir.py 1 | testPyt

FinalProjectDE > airflow_final_project_de > dags > script > sqlscript > agg_insert_data_dim_fact.sql

```sql
-- clear the table data before inserting whole updated data
truncate dim_province restart identity cascade;
truncate dim_district restart identity cascade;
truncate dim_case restart identity cascade;
truncate fact_province_daily  restart identity cascade;
truncate fact_province_monthly  restart identity cascade;
truncate fact_province_yearly  restart identity cascade;
truncate fact_district_monthly  restart identity cascade;
truncate fact_district_yearly  restart identity cascade;

-- Dim Table data insert
-- insert data for dim case
INSERT INTO dim_case (status_name,status_detail)
    SELECT * FROM temp_dim_case;

-- insert data for dim province
insert into dim_province
    select distinct kode_prov, nama_prov
    from data_warehouse;

-- insert data for dim district
insert into dim_district
    select distinct kode_kab,kode_prov,nama_kab
    from data_warehouse;

-- insert data to temp_fact for pre processed data before insert to fact table
insert into temp_fact
    select kode_prov,kode_kab, tanggal::date,
        unnest(array ['suspect_diisolasi','suspect_discarded','suspect_meninggal','closecontact_di
```

DAG FILE EXPLANATION

Dim Table Creation

```python
def create_table_dim_case(con_conf_postgres,ti):
    data_xcom = ti.xcom_pull(key='data_covid_jabar')#, task_ids = 'get_data_
    # print("data_xcom type : ", type(data_xcom))
    data_covid_jabar = pd.read_json(data_xcom, orient = 'records')
    # algorithm for splitting the status name & details

    temp = data_covid_jabar.columns
    status_name = []
    status_detail = []

    #check for every colum name in dataframe
    for nama_kolom in temp:
        # column name with uppercase letter is the name of status
        if nama_kolom.isupper():
            status_name.append(nama_kolom)
        # detail of the status
        else:
            status_detail.append(nama_kolom)

    # splitting the 2 words of status name and detail the save into list
    merge = []
    for word in status_name:
        for sentence in status_detail:
            split = sentence.split("_")
            if word.lower() in split:
                merge.append([split[0].lower(), split[1]])
    dim_case = pd.DataFrame(merge, columns=['status_name','status_detail'])
```

SUSPECT	CLOSECONTACT	PROBABLE	suspect_diisolasi	suspect_discarded	closeco
2,210	274	26	31	2,179	
776	534	7	3	773	
5,536	2,127	33	4,001	1,535	

	status_name	status_detail
0	suspect	diisolasi
1	suspect	discarded
2	suspect	meninggal
3	closecontact	dikarantina
4	closecontact	discarded
5	closecontact	meninggal
6	probable	diisolasi
7	probable	discarded
8	probable	meninggal
9	confirmation	sembuh
10	confirmation	meninggal

Properties | Data | ER Diagram

temp_dim_case | Enter a SQL expression to filter res

	status_name	status_detail
1	suspect	diisolasi
2	suspect	discarded
3	suspect	meninggal
4	closecontact	dikarantina
5	closecontact	discarded
6	closecontact	meninggal
7	probable	diisolasi
8	probable	discarded
9	probable	meninggal
10	confirmation	sembuh
11	confirmation	meninggal

```python
dim_case.to_sql(name = "temp_dim_case", con=engine_postgres, if_exists="replace", index=False)
```

DAG FILE EXPLANATION

Aggregate Function

```
-- insert data to temp_fact for pre processed data before insert to fact table
insert into temp_fact
    select kode_prov,kode_kab, tanggal::date,
        unnest(array ['suspect_diisolasi','suspect_discarded','suspect_meninggal
        , unnest(array [suspect_diisolasi,suspect_discarded,suspect_meninggal,cl
from data_warehouse;
```

Properties | Data | ER Diagram

temp_fact | Enter a SQL expression to filter results (use Ctrl+Space)

	province_id	district_id	date	case	total
1	32	3204	2020-08-05	suspect_diisolasi	31
2	32	3204	2020-08-05	suspect_discarded	2,179
3	32	3204	2020-08-05	suspect_meninggal	0
4	32	3204	2020-08-05	closecontact_dikarantina	0
5	32	3204	2020-08-05	closecontact_discarded	274
6	32	3204	2020-08-05	closecontact_meninggal	0
7	32	3204	2020-08-05	probable_diisolasi	0
8	32	3204	2020-08-05	probable_discarded	0

```
-- insert data to fact_province_monthly
insert into fact_province_monthly(province_id,case_id, "month", total)
    select province_id, dc.case_id, to_char(date,'YYYY-MM') as "month", sum(total) as total
    from temp_fact tf inner join dim_case dc on concat(dc.status_name,'_',dc.status_detail) = tf."case"
    group by province_id, case_id, "month"
    order by province_id, case_id, "month" asc;
```

dim_case | Enter a SQL expression to filter results (use Ctrl+Spac

	case_id	status_name	status_detail
1	1	suspect	diisolasi
2	2	suspect	discarded
3	3	suspect	meninggal
4	4	closecontact	dikarantina
5	5	closecontact	discarded
6	6	closecontact	meninggal
7	7	probable	diisolasi
8	8	probable	discarded
9	9	probable	meninggal
10	10	confirmation	sembuh
11	11	confirmation	meninggal

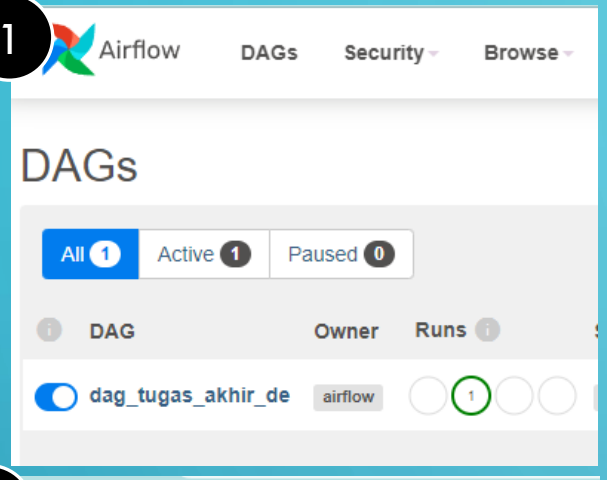
Properties | Data | ER Diagram

fact_province_monthly | Enter a SQL expression to filter results (use Ctrl+Space)

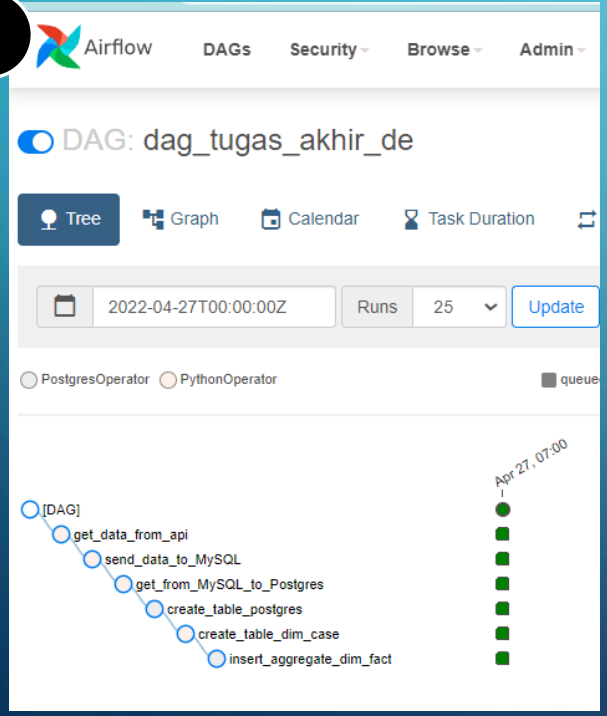
	id	province_id	case_id	month	total
1	1	32	1	2020-08	22,951
2	2	32	1	2020-09	2,043
3	3	32	1	2020-10	2,311
4	4	32	1	2020-11	3,481
5	5	32	1	2020-12	4,677
6	6	32	1	2021-01	5,337
7	7	32	1	2021-02	2,656
8	8	32	1	2021-03	3,294
9	9	32	1	2021-04	1,813
10	10	32	1	2021-05	1,613
11	11	32	1	2021-06	5,168

RUNNING AIRFLOW DAG

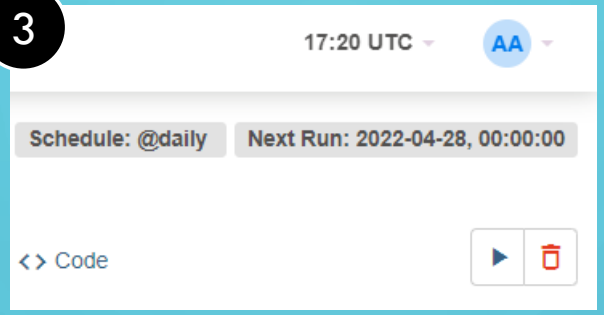
1



2

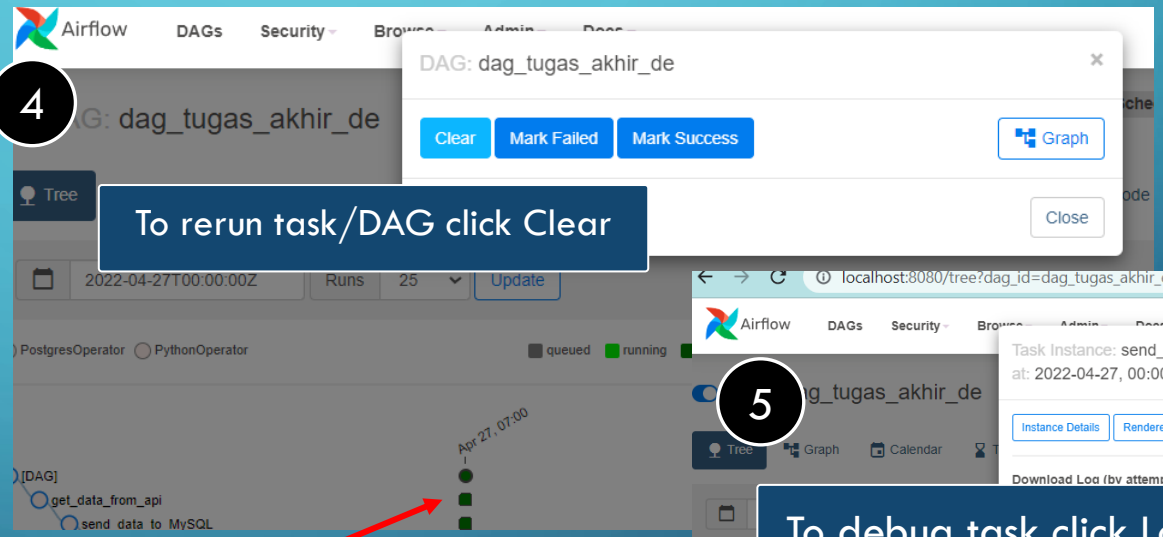


3



Play button to run DAG in new timeline

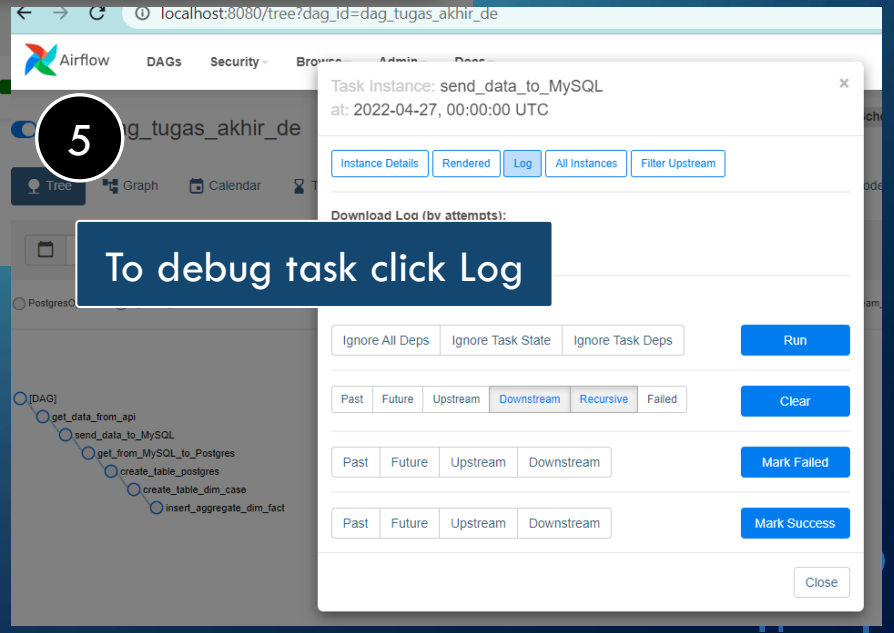
4



To rerun task/DAG click Clear

Select DAG or Task

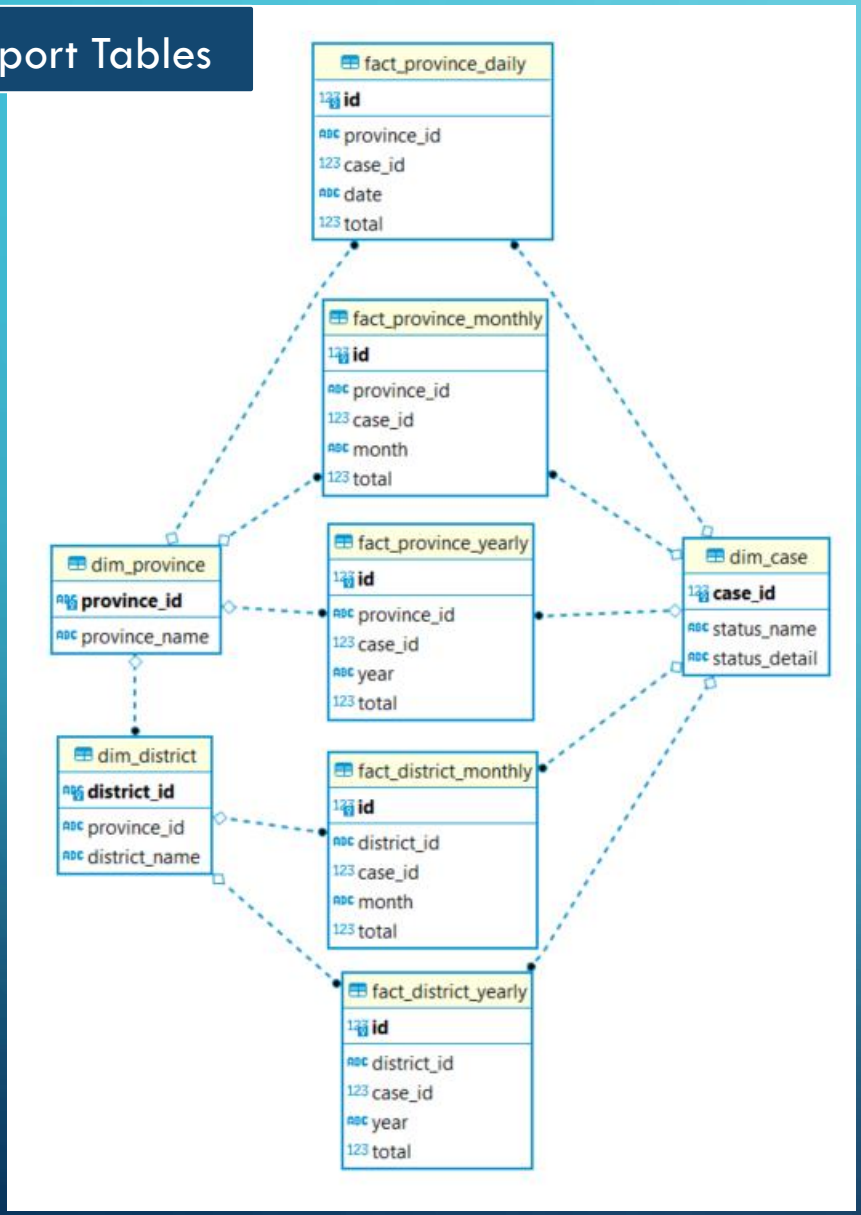
5



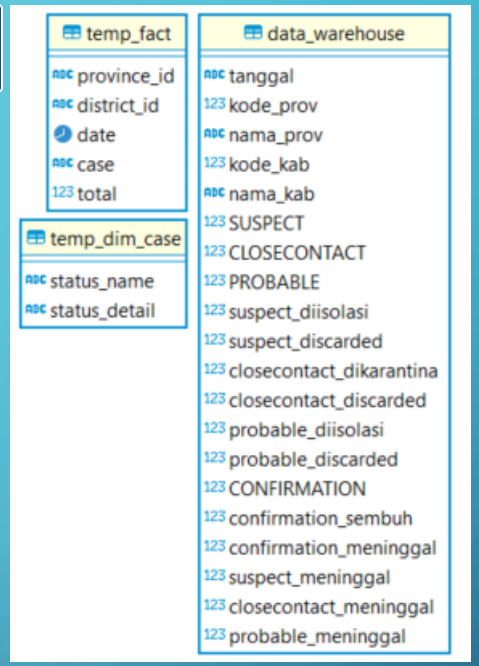
To debug task click Log

ERD RESULT

Report Tables



Other Tables



TABLES RESULT

Dimension Tables

dim_province | Enter a SQL expression to filter result

	province_id	province_name
1	32	Jawa Barat

dim_case | Enter a SQL expression to filter results (use Ctrl+Sp

	case_id	status_name	status_detail
1	1	suspect	diisolasi
2	2	suspect	discarded
3	3	suspect	meninggal
4	4	closecontact	dikarantina
5	5	closecontact	discarded
6	6	closecontact	meninggal
7	7	probable	diisolasi
8	8	probable	discarded
9	9	probable	meninggal
10	10	confirmation	sembuh
11	11	confirmation	meninggal

dim_district | Enter a SQL expression to filter results (use Ctrl+Space)

	district_id	province_id	district_name
1	3205	32	Kabupaten Garut
2	3279	32	Kota Banjar
3	3277	32	Kota Cimahi
4	3271	32	Kota Bogor
5	3203	32	Kabupaten Cianjur
6	3215	32	Kabupaten Karawang
7	3212	32	Kabupaten Indramayu
8	3210	32	Kabupaten Majalengka
9	3213	32	Kabupaten Subang
10	3273	32	Kota Bandung
11	3206	32	Kabupaten Tasikmalaya
12	3214	32	Kabupaten Purwakarta

TABLES RESULT

Fact Tables

fact_province_daily | Enter a SQL expression to filter results (use Ctrl+Space)

	id	province_id	case_id	date	total
1	1	32	1	2020-08-05	42,540
2	2	32	1	2020-08-06	436
3	3	32	1	2020-08-07	42
4	4	32	1	2020-08-08	120
5	5	32	1	2020-08-09	498
6	6	32	1	2020-08-10	264
7	7	32	1	2020-08-11	40
8	8	32	1	2020-08-12	136
9	9	32	1	2020-08-13	170
10	10	32	1	2020-08-14	204
11	11	32	1	2020-08-15	102
12	12	32	1	2020-08-16	100

fact_province_monthly | Enter a SQL expression to filter results (use Ctrl+Space)

	id	province_id	case_id	month	total
1	1	32	1	2020-08	45,902
2	2	32	1	2020-09	4,086
3	3	32	1	2020-10	4,622
4	4	32	1	2020-11	6,962
5	5	32	1	2020-12	9,354
6	6	32	1	2021-01	10,674
7	7	32	1	2021-02	5,312
8	8	32	1	2021-03	6,588
9	9	32	1	2021-04	3,626
10	10	32	1	2021-05	3,226
11	11	32	1	2021-06	10,336
12	12	32	1	2021-07	5,313

fact_province_yearly | Enter a SQL expression to filter results (use Ctrl+Space)

	id	province_id	case_id	year	total
1	1	32	1	2020	35,463
2	2	32	1	2021	23,972
3	3	32	1	2022	0
4	4	32	2	2020	100,562
5	5	32	2	2021	107,756
6	6	32	2	2022	0
7	7	32	3	2020	3,125
8	8	32	3	2021	0
9	9	32	3	2022	0
10	10	32	4	2020	55,133
11	11	32	4	2021	72,689
12	12	32	4	2022	0

fact_district_monthly | Enter a SQL expression to filter results (use Ctrl+Space)

	id	district_id	case_id	month	total
1	1	3201	1	2020-08	1,004
2	2	3201	1	2020-09	588
3	3	3201	1	2020-10	1,034
4	4	3201	1	2020-11	948
5	5	3201	1	2020-12	964
6	6	3201	1	2021-01	658
7	7	3201	1	2021-02	796
8	8	3201	1	2021-03	448
9	9	3201	1	2021-04	148
10	10	3201	1	2021-05	244
11	11	3201	1	2021-06	142
12	12	3201	1	2021-07	6

fact_district_yearly | Enter a SQL expression to filter results (use Ctrl+Space)

	id	district_id	case_id	year	total
1	1	3201	1	2020	2,269
2	2	3201	1	2021	1,221
3	3	3201	1	2022	0
4	4	3201	2	2020	6,125
5	5	3201	2	2021	1,347
6	6	3201	2	2022	0
7	7	3201	3	2020	702
8	8	3201	3	2021	0
9	9	3201	3	2022	0
10	10	3201	4	2020	0
11	11	3201	4	2021	0
12	12	3201	4	2022	0