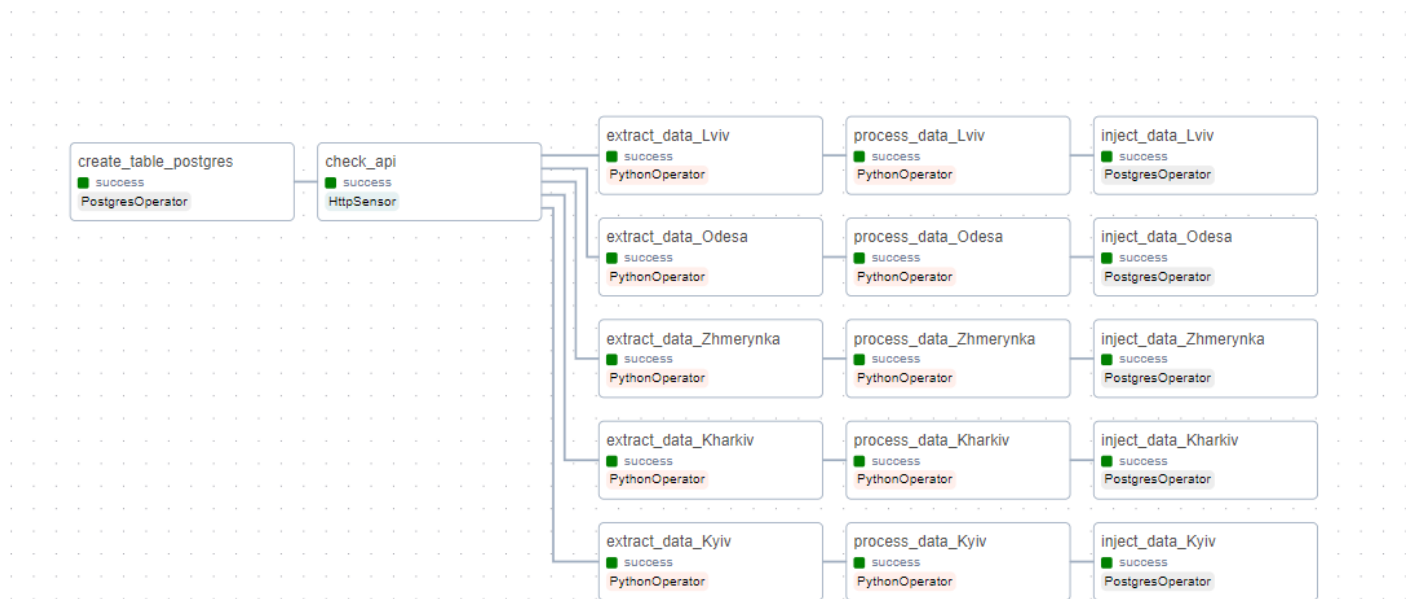I use PostgreSQL for airflow db and also use PostgreSQL db for saving weather data.
Here is how DAG looks:

Jobs:

create_table_postgres
This job creates table in postgres with columns city, timestamp, date, temp, humidity, clouds, wind_speed

check_api
This job checks whether OpenWeather API is working

extract_data_{city}
This job extracts data from OpenWeather for job DAG start date and for given city
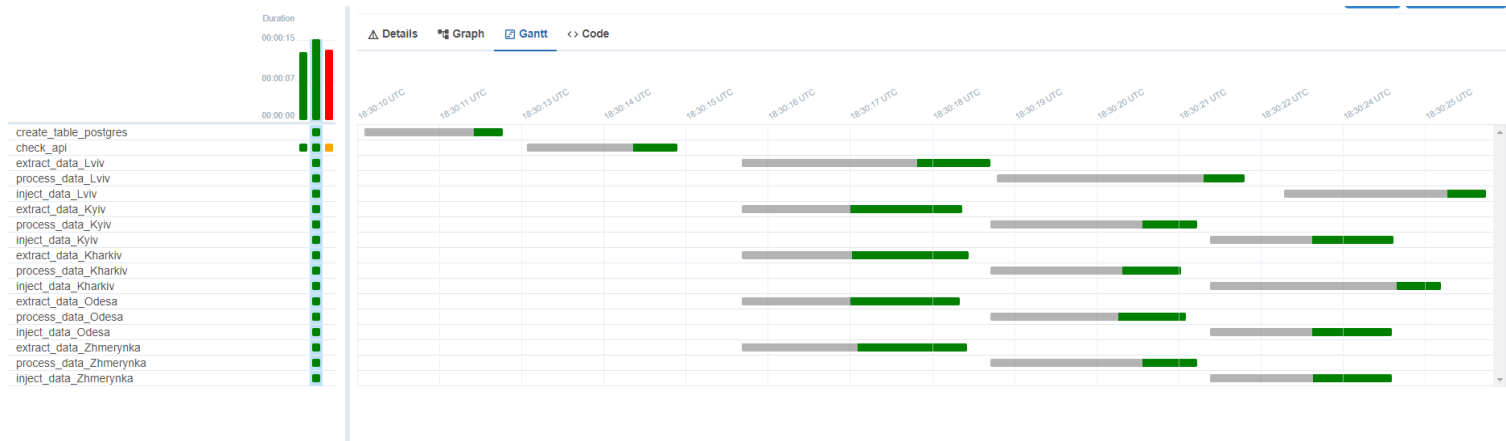
process_data_{city}
This jobs processes data for given city and returns only values that need to be injected in table

inject_data_{city}
This job injects data into previously created PostgreSQL table

Jobs extract_data_{city}, process_data_{city} and inject_data_{city} are running in parallel
It can be seen on Gantt chart:

Duration
00:00:15
00:00:07
00:00:00

⚠ Details   🔊 Graph   ⊘ Gantt   <> Code

18:30:10 UTC 18:30:11 UTC 18:30:13 UTC 18:30:14 UTC 18:30:15 UTC 18:30:16 UTC 18:30:17 UTC 18:30:18 UTC 18:30:19 UTC 18:30:20 UTC 18:30:21 UTC 18:30:22 UTC 18:30:24 UTC 18:30:25 UTC

create_table_postgres
check_api
extract_data_Lviv
process_data_Lviv
inject_data_Lviv
extract_data_Kyiv
process_data_Kyiv
inject_data_Kyiv
extract_data_Kharkiv
process_data_Kharkiv
inject_data_Kharkiv
extract_data_Odesa
process_data_Odesa
inject_data_Odesa
extract_data_Zhmerynka
process_data_Zhmerynka
inject_data_Zhmerynka

Here is the results of running pipeline for two different dates:

```
weather_db=# select * from measures;
   city     |      timestamp      |        date         | temp  | humidity | clouds | wind_speed
-----------+---------------------+---------------------+-------+----------+--------+------------
 Kharkiv   | 2023-12-01 20:29:09 | 2023-12-01 18:29:09 |  1.85 |       98 |    100 |        3.4
 Zhmerynka | 2023-12-01 20:29:09 | 2023-12-01 18:29:09 | -0.33 |       98 |    100 |       2.95
 Odesa     | 2023-12-01 20:29:09 | 2023-12-01 18:29:09 |  9.96 |       77 |     96 |       3.21
 Lviv      | 2023-12-01 20:29:09 | 2023-12-01 18:29:09 | -0.64 |       99 |    100 |       1.91
 Kyiv      | 2023-12-01 20:29:09 | 2023-12-01 18:29:09 |  0.19 |       92 |    100 |       1.86
 Odesa     | 2023-11-11 02:00:00 | 2023-11-11 00:00:00 | 14.55 |       81 |    100 |      14.23
 Zhmerynka | 2023-11-11 02:00:00 | 2023-11-11 00:00:00 |  9.31 |       71 |    100 |       8.46
 Kyiv      | 2023-11-11 02:00:00 | 2023-11-11 00:00:00 |  7.53 |       81 |    100 |       0.89
 Kharkiv   | 2023-11-11 02:00:00 | 2023-11-11 00:00:00 |  4.55 |       85 |    100 |       3.55
 Lviv      | 2023-11-11 02:00:00 | 2023-11-11 00:00:00 |  5.71 |       77 |    100 |       2.69
(10 rows)
```