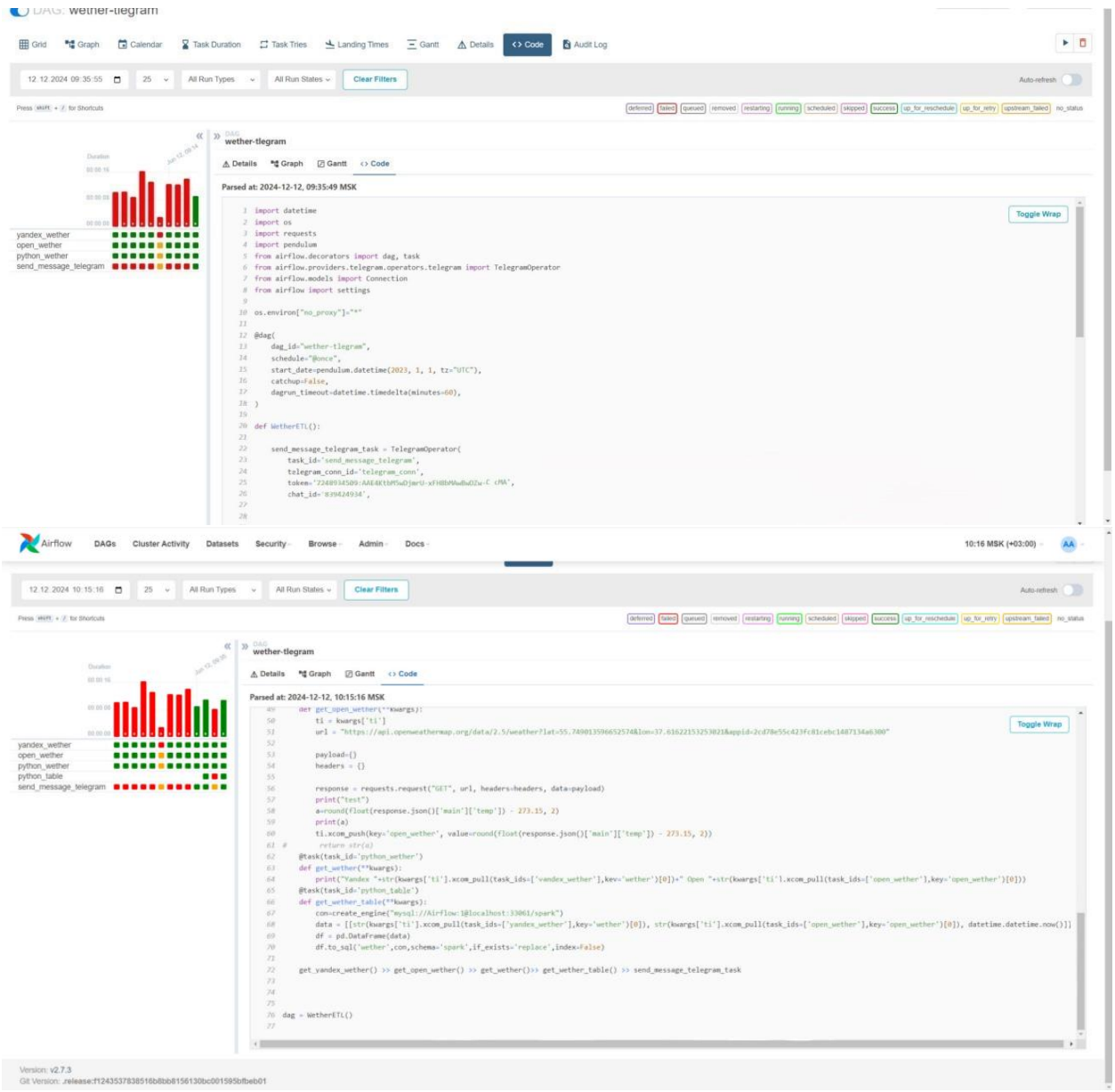
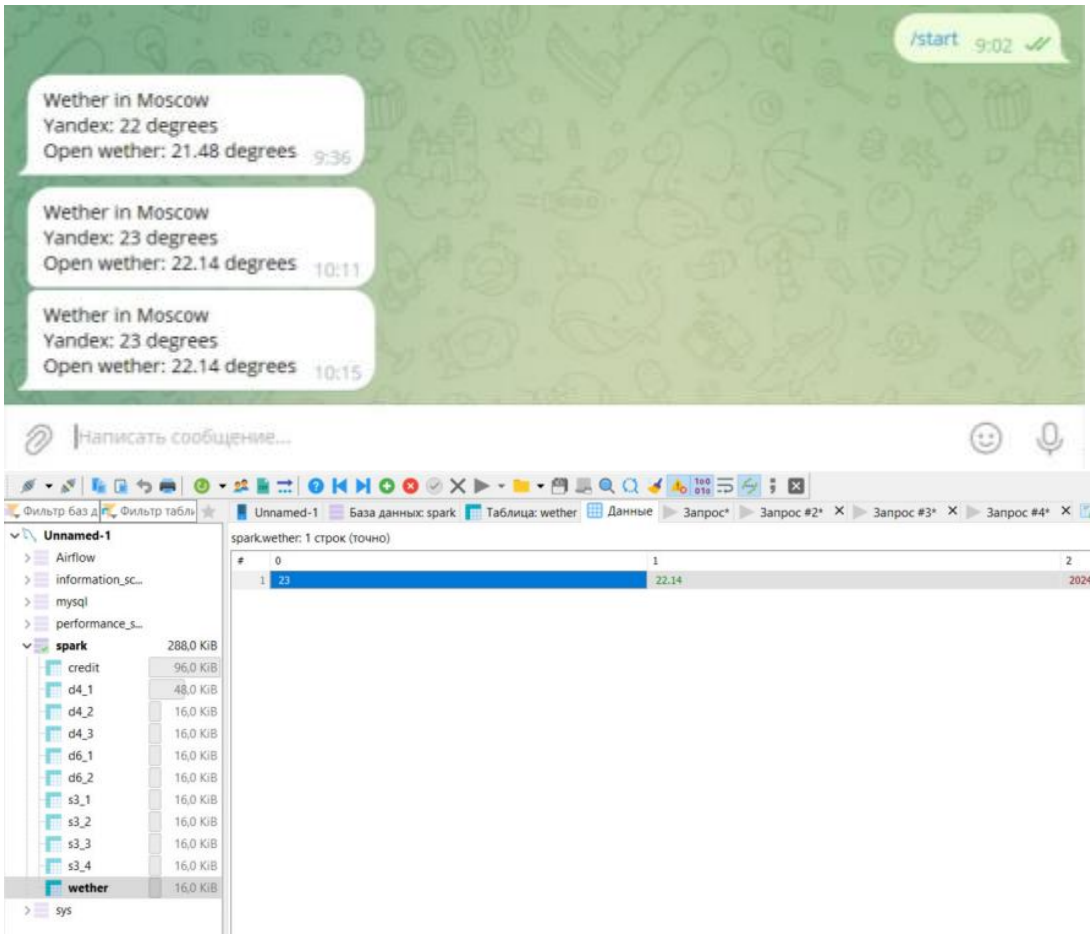

DAGs: wether-tlegram
Grid
Graph
Calendar
Task Duration
Task Tries
Landing Times
Gantt
Details
Code
Audit Log
Clear Filters
Auto-refresh
yandex_wether
open_wether
python_wether
send_message_telegram
DAG:
wether-tlegram
Details
Graph
Gantt
Code
Parsed at: 2024-12-12, 09:35:49 MSK
Toggle Wrap
import datetime
import os
import requests
import pendulum
from airflow.decorators import dag, task
from airflow.providers.telegram.operators.telegram import TelegramOperator
from airflow.models import Connection
from airflow import settings
os.environ["no_proxy"]="*"
@dag(
dag_id="wether-tlegram",
schedule="@once",
start_date=pendulum.datetime(2023, 1, 1, tz="UTC"),
catchup=False,
dagrun_timeout=datetime.timedelta(minutes=60),
)
def WetherETL():
send_message_telegram_task = TelegramOperator(
task_id='send_message_telegram',
telegram_conn_id='telegram_conn',
token='7248934509:AAE4KtbH5wDjmrU-xFHBbMwdwDZw-C cMA',
chat_id='839424934',
Airflow
DAGs
Cluster Activity
Datasets
Security
Browse
Admin
Docs
10:16 MSK (+03:00)
AA
12.12.2024 10:15:16
All Run Types
All Run States
python_table
wether-tlegram
Details
Graph
Gantt
Code
Parsed at: 2024-12-12, 10:15:16 MSK
Toggle Wrap
def get_open_wether(**kwargs):
ti = kwargs['ti']
url = "https://api.openweathermap.org/data/2.5/weather?lat=55.749013596652574&lon=37.616221532503021&appid=2cd78e55c423fc81cebc1487134a6300"
payload={}
headers = {}
response = requests.request("GET", url, headers=headers, data=payload)
print("test")
a=round(float(response.json()['main']['temp']) - 273.15, 2)
print(a)
ti.xcom_push(key='open_wether', value=round(float(response.json()['main']['temp']) - 273.15, 2))
# return str(a)
@task(task_id='python_wether')
def get_wether(**kwargs):
print("Yandex "+str(kwargs['ti'].xcom_pull(task_ids=['yandex_wether'],key='wether')[0])+" Open "+str(kwargs['ti'].xcom_pull(task_ids=['open_wether'],key='open_wether')[0]))
@task(task_id='python_table')
def get_wether_table(**kwargs):
con=create_engine("mysql://Air-flow:1@localhost:33061/spark")
data = [[str(kwargs['ti'].xcom_pull(task_ids=['yandex_wether'],key='wether')[0]), str(kwargs['ti'].xcom_pull(task_ids=['open_wether'],key='open_wether')[0]), datetime.datetime.now()]]
df = pd.DataFrame(data)
df.to_sql('wether',con,schema='spark',if_exists='replace',index=False)
get_yandex_wether() >> get_open_wether() >> get_wether()>> get_wether_table() >> send_message_telegram_task
dag = WetherETL()
Version: v2.7.3
Git Version: .release:f1243537838516b8bb8156130bc00159bfbeb01


/start 9:02
Wether in Moscow
Yandex: 22 degrees
Open wether: 21.48 degrees 9:36
Wether in Moscow
Yandex: 23 degrees
Open wether: 22.14 degrees 10:11
Wether in Moscow
Yandex: 23 degrees
Open wether: 22.14 degrees 10:15
Написать сообщение...
Фильтр баз д
Фильтр табл
Unnamed-1
Airflow
information_sc...
mysql
performance_s...
spark 288,0 KiB
credit 96,0 KiB
d4_1 48,0 KiB
d4_2 16,0 KiB
d4_3 16,0 KiB
d6_1 16,0 KiB
d6_2 16,0 KiB
s3_1 16,0 KiB
s3_2 16,0 KiB
s3_3 16,0 KiB
s3_4 16,0 KiB
wether 16,0 KiB
sys
Unnamed-1
База данных: spark
Таблица: wether
Данные
Запрос*
Запрос #2*
Запрос #3*
Запрос #4*
spark.wether: 1 строк (точно)
0 1 2
1 23 22.14 2024-12-12 10:15:24