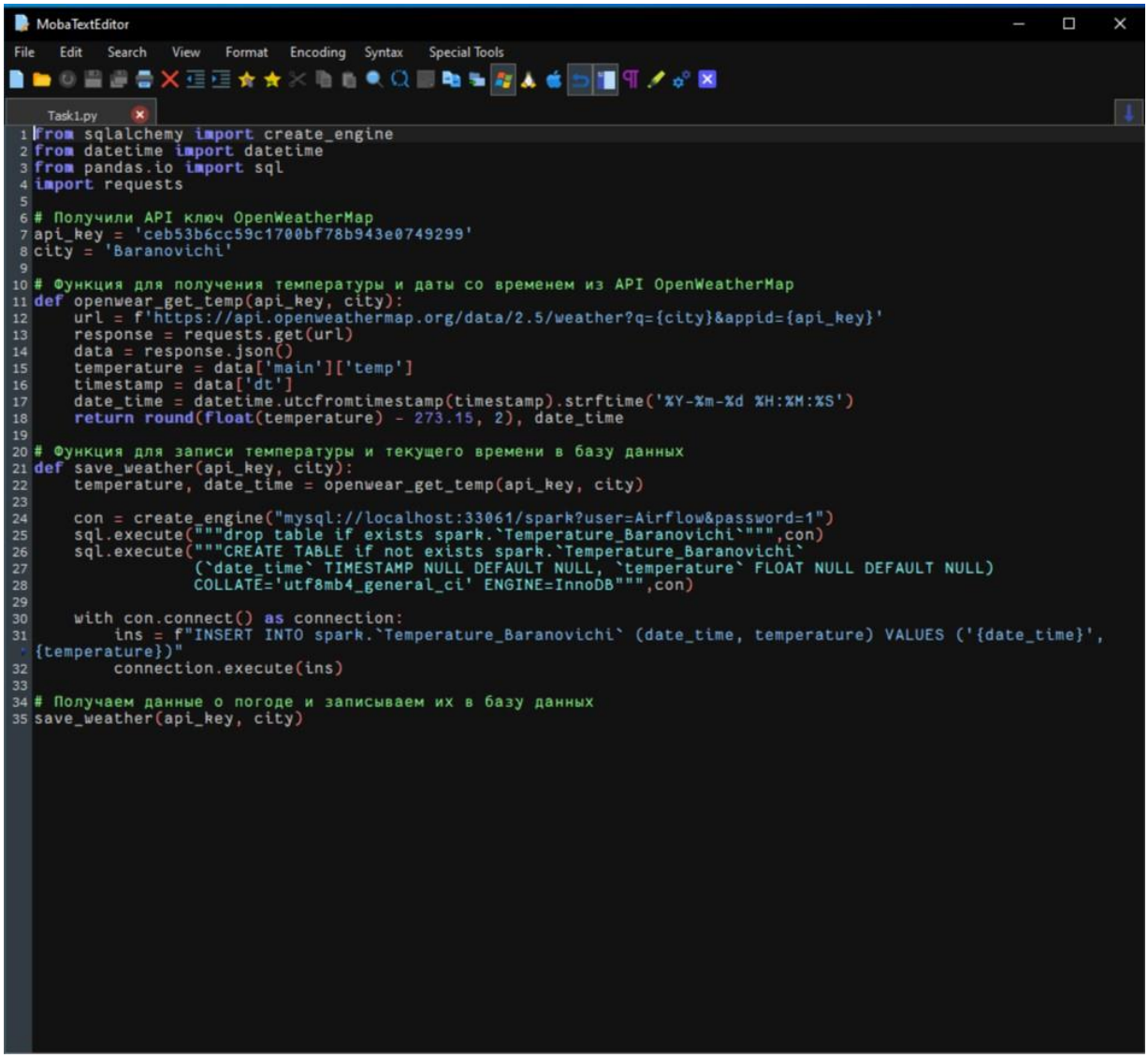
ETL: автоматизация подготовки данных

Урок 7. Построение пайплайнов и визуализация потоков данных в Airflow

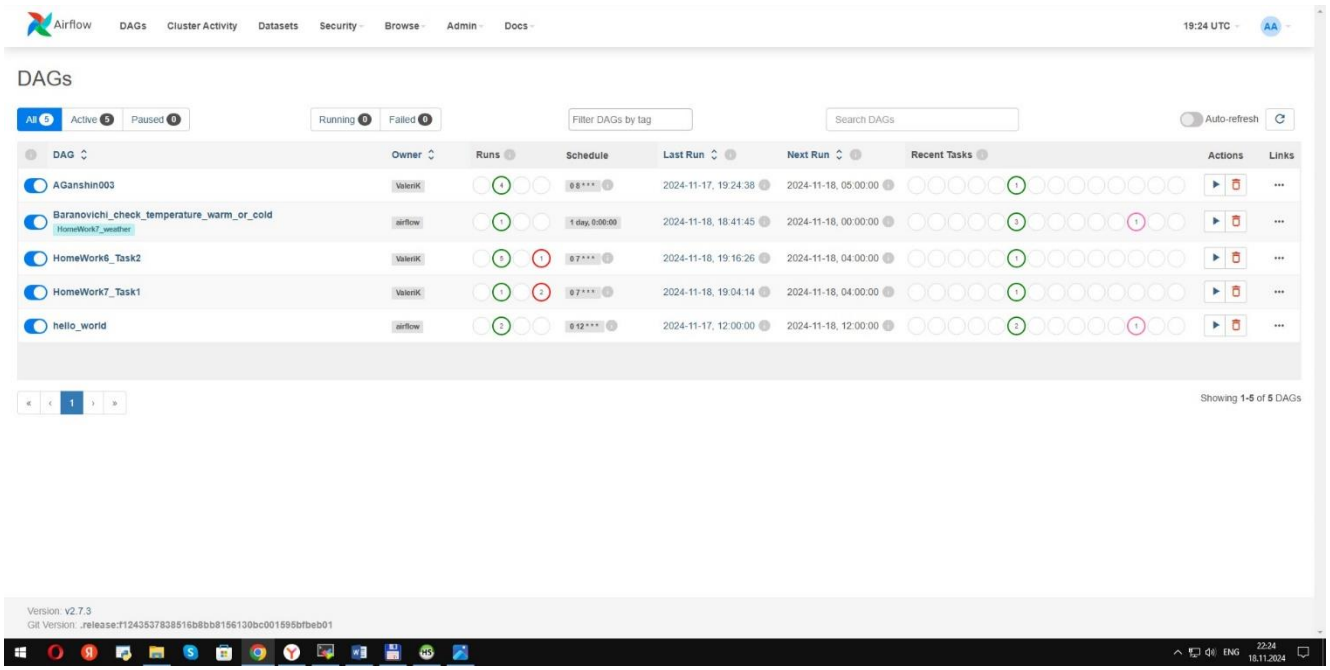
Задание

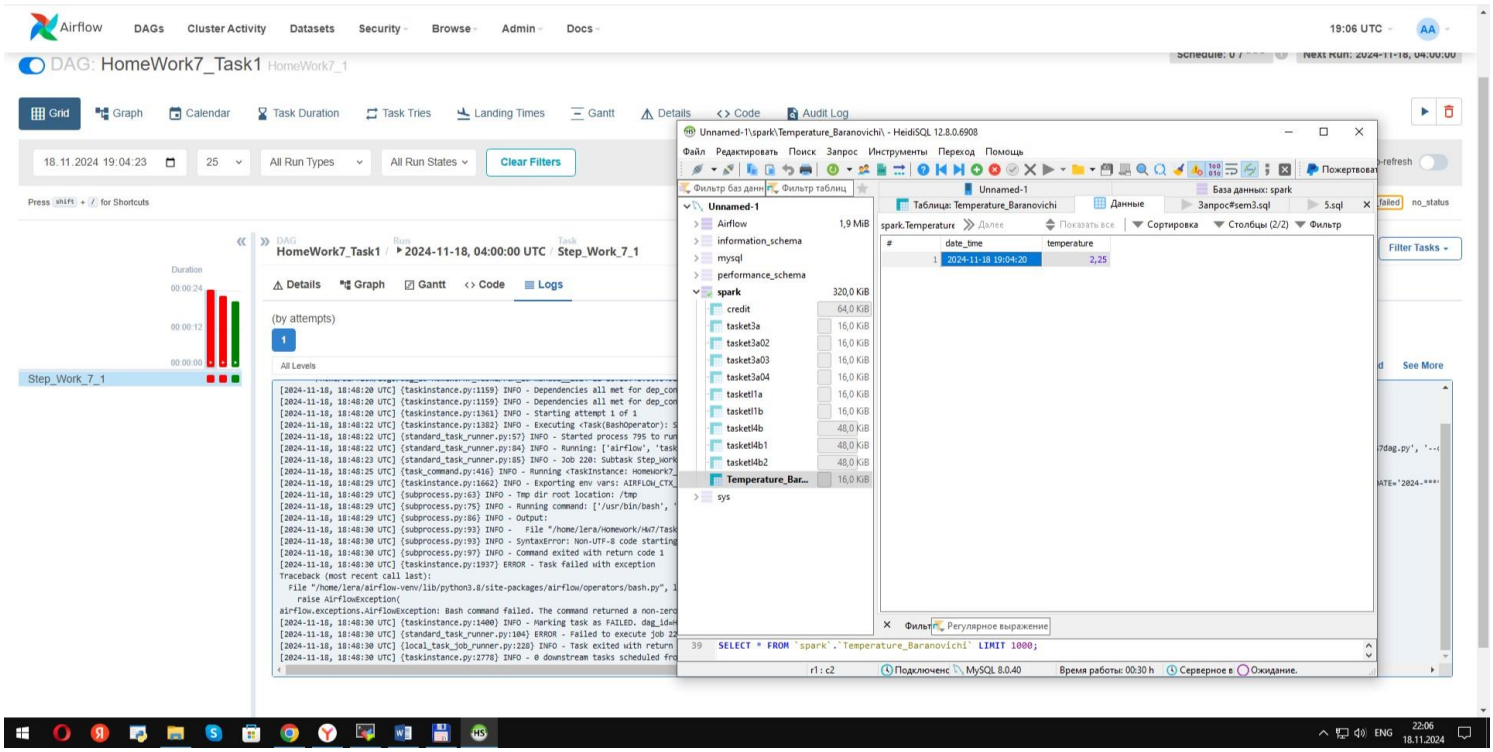
1. Ваша задача с использование пандас, записать полученную температуру в таблицу mysql.

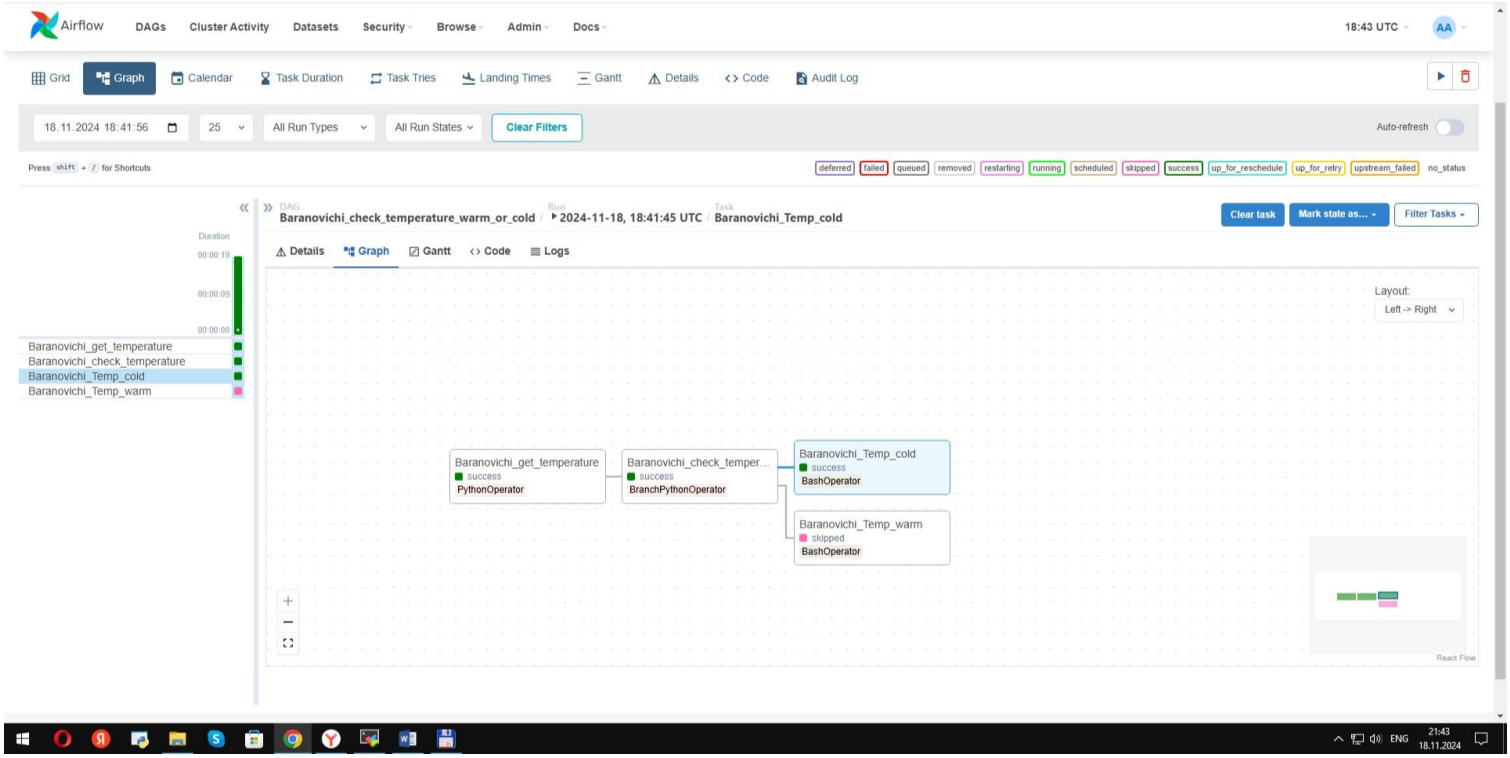
Таблица должна содержать как минимум текущее время и температуру (т.е. два поля). Таблицу не удаляем, используем append.

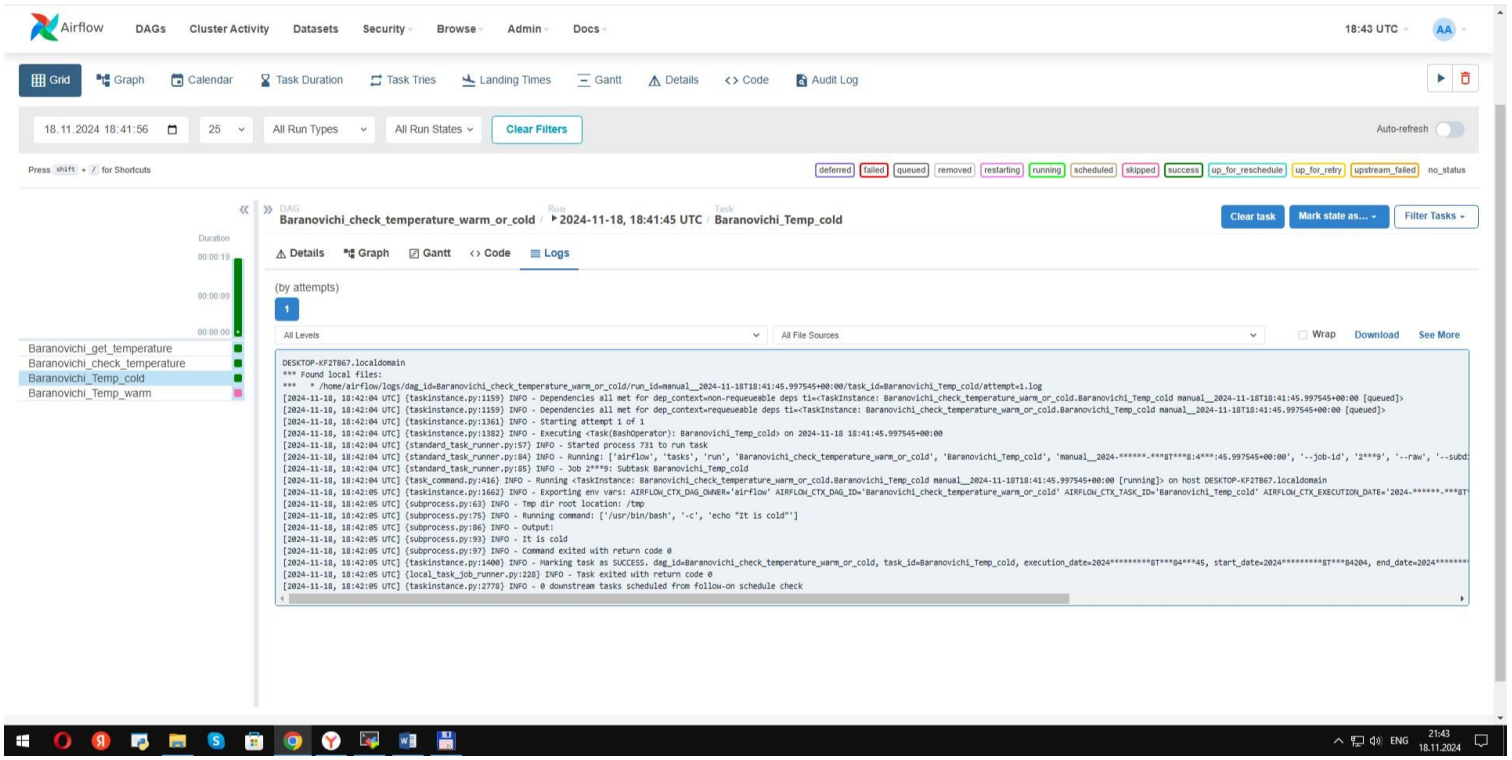


```python
#DAG5
dag3 = DAG('HomeWork7_Task1',
default_args=default_args,
description="HomeWork7_1",
catchup=False,
schedule_interval='0 7 * * *')
task51 = BashOperator(
task_id='Step_Work_7_1',
bash_command='export SPARK_HOME=/home/spark && export PATH=$PATH:$SPARK_HOME/bin:$SPARK_HOME/sbin && python3 /home/lera/Homework/HW7/Task1.py',
dag=dag3)
#DAG6
from datetime import datetime
from airflow import DAG
from airflow.operators.bash import BashOperator
from airflow.operators.python import PythonOperator, BranchPythonOperator
from datetime import datetime, timedelta
from airflow.operators.python import PythonOperator
from datetime import datetime
import requests
import os

def openwear_get_temp(**kwargs):

    ti = kwargs['ti']
    city = "Baranovichi"
    api_key = "ceb53b6cc59c1700bf78b943e0749299"
    url = f"https://api.openweathermap.org/data/2.5/weather?q={city}&appid={api_key}"
    payload = {}
    headers = {}
    response = requests.request("GET", url, headers=headers, data=payload)
    return round(float(response.json()['main']['temp'])-273.15, 2)


def openwear_check_temp(ti):
    temp = int(ti.xcom_pull(task_ids='Baranovichi_get_temperature'))
    print(f'Temperature now is {temp}')
    if temp >= 15:
        return 'Baranovichi_Temp_warm'
    else:
        return 'Baranovichi_Temp_cold'

with DAG(
        'Baranovichi_check_temperature_warm_or_cold',
        start_date=datetime(2024, 11, 18),
        catchup=False,
        tags=['HomeWork7_weather'],
) as dag:
    Baranovichi_get_temperature = PythonOperator(
        task_id='Baranovichi_get_temperature',
        python_callable=openwear_get_temp,
    )

    Baranovichi_check_temperature = BranchPythonOperator(
        task_id='Baranovichi_check_temperature',
        python_callable=openwear_check_temp,
    )

    Baranovichi_Temp_warm = BashOperator(
        task_id='Baranovichi_Temp_warm',
        bash_command='echo "It is warm"',
    )

    Baranovichi_Temp_cold = BashOperator(
        task_id='Baranovichi_Temp_cold',
        bash_command='echo "It is cold"',
    )

Baranovichi_get_temperature >> Baranovichi_check_temperature >> [Baranovichi_Temp_warm, Baranovichi_Temp_cold]

```




Airflow
DAGs
Cluster Activity
Datasets
Security
Browse
Admin
Docs
19:06 UTC
DAG: HomeWork7_Task1
Grid
Graph
Calendar
Task Duration
Task Tries
Landing Times
Gantt
Details
Code
Audit Log
18.11.2024 19:04:23
All Run Types
All Run States
Clear Filters
Step_Work_7_1
Logs
Unnamed-1\spark\Temperature_Baranovichi\ - HeidiSQL 12.8.0.6908
Airflow
information_schema
mysql
performance_schema
spark
Temperature_Bar...
sys
SELECT * FROM `spark`.`Temperature_Baranovichi` LIMIT 1000;


Airflow
DAGs
Cluster Activity
Datasets
Security
Browse
Admin
Docs
18:43 UTC
Grid
Graph
Calendar
Task Duration
Task Tries
Landing Times
Gantt
Details
Code
Audit Log
18.11.2024 18:41:56
All Run Types
All Run States
Clear Filters
Baranovichi_check_temperature_warm_or_cold
2024-11-18, 18:41:45 UTC
Baranovichi_Temp_cold
Clear task
Mark state as...
Filter Tasks
Baranovichi_get_temperature
Baranovichi_check_temperature
Baranovichi_Temp_cold
Baranovichi_Temp_warm
PythonOperator
BranchPythonOperator
BashOperator
skipped
Layout:
Left -> Right
React Flow


Airflow
DAGs
Cluster Activity
Datasets
Security
Browse
Admin
Docs
18:43 UTC
AA
Grid
Graph
Calendar
Task Duration
Task Tries
Landing Times
Gantt
Details
Code
Audit Log
18.11.2024 18:41:56
25
All Run Types
All Run States
Clear Filters
Auto-refresh
Press shift + / for Shortcuts
deferred
failed
queued
removed
restarting
running
scheduled
skipped
success
up_for_reschedule
up_for_retry
upstream_failed
no_status
Duration
00:00:19
00:00:09
00:00:00
Baranovichi_get_temperature
Baranovichi_check_temperature
Baranovichi_Temp_cold
Baranovichi_Temp_warm
DAG
Baranovichi_check_temperature_warm_or_cold
Run
2024-11-18, 18:41:45 UTC
Task
Baranovichi_Temp_cold
Clear task
Mark state as...
Filter Tasks
Details
Graph
Gantt
Code
Logs
(by attempts)
1
All Levels
All File Sources
Wrap
Download
See More
DESKTOP-KF2TB67.localdomain
*** Found local files:
*** * /home/airflow/logs/dag_id=Baranovichi_check_temperature_warm_or_cold/run_id=manual__2024-11-18T18:41:45.997545+00:00/task_id=Baranovichi_Temp_cold/attempt=1.log
[2024-11-18, 18:42:04 UTC] {taskinstance.py:1159} INFO - Dependencies all met for dep_context=non-requeueable deps ti=<TaskInstance: Baranovichi_check_temperature_warm_or_cold.Baranovichi_Temp_cold manual__2024-11-18T18:41:45.997545+00:00 [queued]>
[2024-11-18, 18:42:04 UTC] {taskinstance.py:1159} INFO - Dependencies all met for dep_context=requeueable deps ti=<TaskInstance: Baranovichi_check_temperature_warm_or_cold.Baranovichi_Temp_cold manual__2024-11-18T18:41:45.997545+00:00 [queued]>
[2024-11-18, 18:42:04 UTC] {taskinstance.py:1361} INFO - Starting attempt 1 of 1
[2024-11-18, 18:42:04 UTC] {taskinstance.py:1382} INFO - Executing <Task(BashOperator): Baranovichi_Temp_cold> on 2024-11-18 18:41:45.997545+00:00
[2024-11-18, 18:42:04 UTC] {standard_task_runner.py:57} INFO - Started process 731 to run task
[2024-11-18, 18:42:04 UTC] {standard_task_runner.py:84} INFO - Running: ['airflow', 'tasks', 'run', 'Baranovichi_check_temperature_warm_or_cold', 'Baranovichi_Temp_cold', 'manual__2024-******-***8T***8:4***:45.997545+00:00', '--job-id', '2***9', '--raw', '--subd
[2024-11-18, 18:42:04 UTC] {standard_task_runner.py:85} INFO - Job 2***9: Subtask Baranovichi_Temp_cold
[2024-11-18, 18:42:04 UTC] {task_command.py:416} INFO - Running <TaskInstance: Baranovichi_check_temperature_warm_or_cold.Baranovichi_Temp_cold manual__2024-11-18T18:41:45.997545+00:00 [running]> on host DESKTOP-KF2TB67.localdomain
[2024-11-18, 18:42:05 UTC] {taskinstance.py:1662} INFO - Exporting env vars: AIRFLOW_CTX_DAG_OWNER='airflow' AIRFLOW_CTX_DAG_ID='Baranovichi_check_temperature_warm_or_cold' AIRFLOW_CTX_TASK_ID='Baranovichi_Temp_cold' AIRFLOW_CTX_EXECUTION_DATE='2024-******-***8T
[2024-11-18, 18:42:05 UTC] {subprocess.py:63} INFO - Tmp dir root location: /tmp
[2024-11-18, 18:42:05 UTC] {subprocess.py:75} INFO - Running command: ['/usr/bin/bash', '-c', 'echo "It is cold"']
[2024-11-18, 18:42:05 UTC] {subprocess.py:86} INFO - Output:
[2024-11-18, 18:42:05 UTC] {subprocess.py:93} INFO - It is cold
[2024-11-18, 18:42:05 UTC] {subprocess.py:97} INFO - Command exited with return code 0
[2024-11-18, 18:42:05 UTC] {taskinstance.py:1400} INFO - Marking task as SUCCESS. dag_id=Baranovichi_check_temperature_warm_or_cold, task_id=Baranovichi_Temp_cold, execution_date=2024********8T***84***45, start_date=2024********8T***84204, end_date=2024********
[2024-11-18, 18:42:05 UTC] {local_task_job_runner.py:228} INFO - Task exited with return code 0
[2024-11-18, 18:42:05 UTC] {taskinstance.py:2778} INFO - 0 downstream tasks scheduled from follow-on schedule check
ENG
21:43
18.11.2024