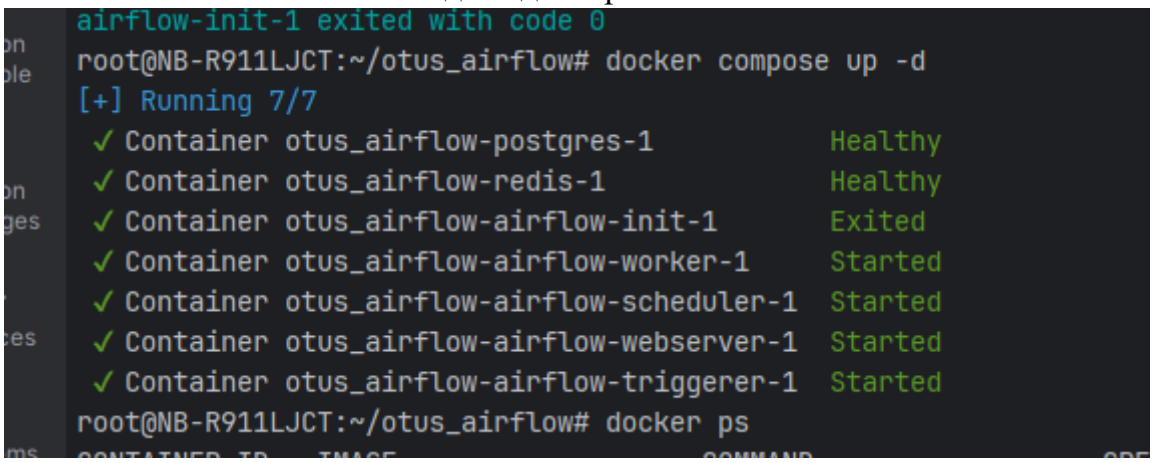
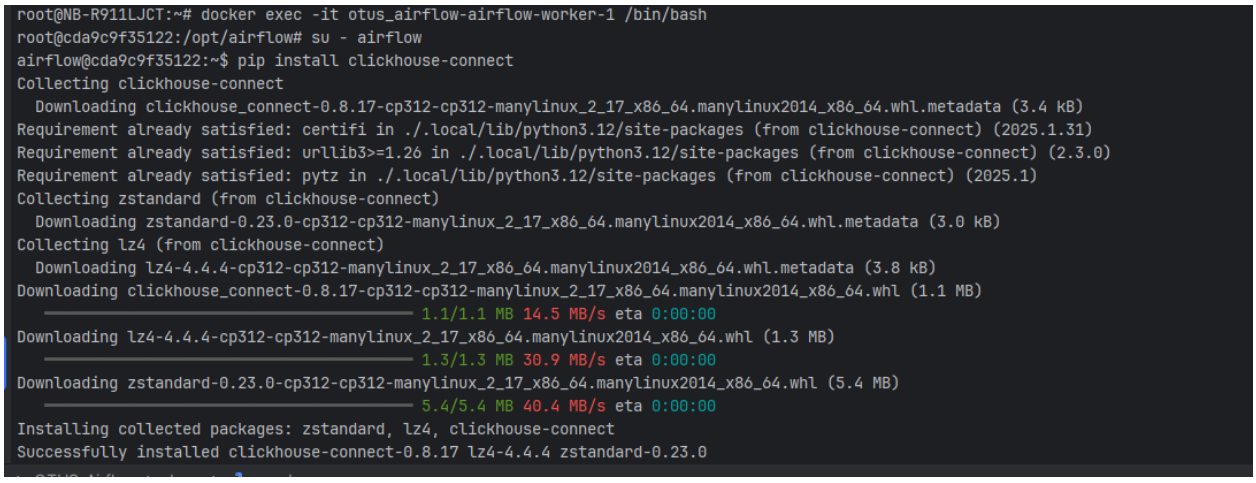
Загрузка данных в ClickHouse

1. Разверните инструмент для переливки данных (например, Airflow или Airbyte).
2. Разверните ClickHouse.
3. Выберите любой источник данных (API, сервис, БД и пр.).
4. Настройте подключение инструмента к ClickHouse.
5. Постройте пайплайн для переливки данных в ClickHouse с возможностью регулярного или одноразового выполнения.

1. На локальной машине был поднят доккер с AF.



В него были добавлены необходимые зависимости – clickhouse-connect



2. Развернул clickhouse на той же машине, создал в нем базу данных otus и две демонстрационных таблицы «otus.read_from» - таблица ИЗ которой в даге будем читать данные и «otus.write_in» - таблица куда будем данные записывать. «otus.read_from» наполнил случайными данными (150 строк).
3. Для демонстрации решил, что читать буду из того же клика, куда и писать.

4. Подключение было настроено через clickhouse-connect, креды положил в connections AF.

```python
# Функция для получения данных из соединения
def write_data_to_clickhouse(**kwargs) -> dict:  1 usage
    # Получаем ID соединения из параметров
    conn_id = kwargs.get('conn_id')

    # Извлекаем соединение
    connection = BaseHook.get_connection(conn_id)

    # Получаем данные соединения
    conn_data = {
        'host': connection.host,
        'port': connection.port,
        'user': connection.login,
        'password': connection.password,
        'database': connection.schema,
    }

    client = get_client(**conn_data)
```

5. Был составлен DAG на чтение из клика из одной таблицы и вставка в другую.

```
cda9c9f35122
 ▶ Log message source details
[2025-04-20, 10:23:37 UTC] {local_task_job_runner.py:123} ▶ Pre task execution logs
[2025-04-20, 10:23:37 UTC] {base.py:84} INFO - Retrieving connection 'clickhouse_local'
[2025-04-20, 10:23:37 UTC] {logging_mixin.py:190} INFO - Таблица из которой мы читаем данные имеет размерность:  (150, 2)
[2025-04-20, 10:23:37 UTC] {logging_mixin.py:190} INFO - Таблица в которую данные будут записаны ДО записи имеет размерность:  (0, 0)
[2025-04-20, 10:23:37 UTC] {logging_mixin.py:190} INFO - Таблица в которую записались данные ПОСЛЕ записи имеет размерность:  (150, 2)
[2025-04-20, 10:23:37 UTC] {python.py:240} INFO - Done. Returned value was: None
[2025-04-20, 10:23:37 UTC] {taskinstance.py:341} ▶ Post task execution logs
```

Из логов AF видно, что чначала мы читаем данные из otus.read_from, которая имеет размерность 150 строк и 2 столбца. Читаем данные из otus.write_in и убеждаемся что она пустая, затем записываем в нее то что прочли из otus.read_from и убеждаемся, что данные записались (размерность стала 150,2)

Код ДАГА:

```python
from airflow import DAG
from airflow.operators.python import PythonOperator
from airflow.hooks.base_hook import BaseHook
from datetime import datetime
from clickhouse_connect import get_client

# Функция для получения данных из соединения
def write_data_to_clickhouse(**kwargs) -> dict:  1 usage
    # Получаем ID соединения из параметров
    conn_id = kwargs.get('conn_id')

    # Извлекаем соединение
    connection = BaseHook.get_connection(conn_id)

    # Получаем данные соединения
    conn_data = {
        'host': connection.host,
        'port': connection.port,
        'user': connection.login,
        'password': connection.password,
        'database': connection.schema,
    }

    client = get_client(**conn_data)
    df_from = client.query_df("""SELECT * FROM otus.read_from""")
    print("Таблица из которой мы читаем данные имеет размерность: ", df_from.shape)

    df_in = client.query_df("""SELECT * FROM otus.write_in""")
    print("Таблица в которую данные будут записаны ДО записи имеет размерность: ", df_in.shape)

    client.insert_df(
        table="otus.write_in",
        df=df_from
    )

    df_in_after = client.query_df("""SELECT * FROM otus.write_in""")
    print("Таблица в которую записались данные ПОСЛЕ записи имеет размерность: ", df_in_after.shape)


# Определяем DAG
with DAG(
        dag_id='write_data_to_clickhouse',
        schedule_interval='@daily',
        start_date=datetime( year: 2023,  month: 1,  day: 1),
        catchup=False,
) as dag:
    # Определяем задачу
    task = PythonOperator(
        task_id='write_data_to_clickhouse_task',
        python_callable=write_data_to_clickhouse,
        op_kwargs={'conn_id': 'clickhouse_local'},  # Замените на ваш ID соединения
    )

    task
```