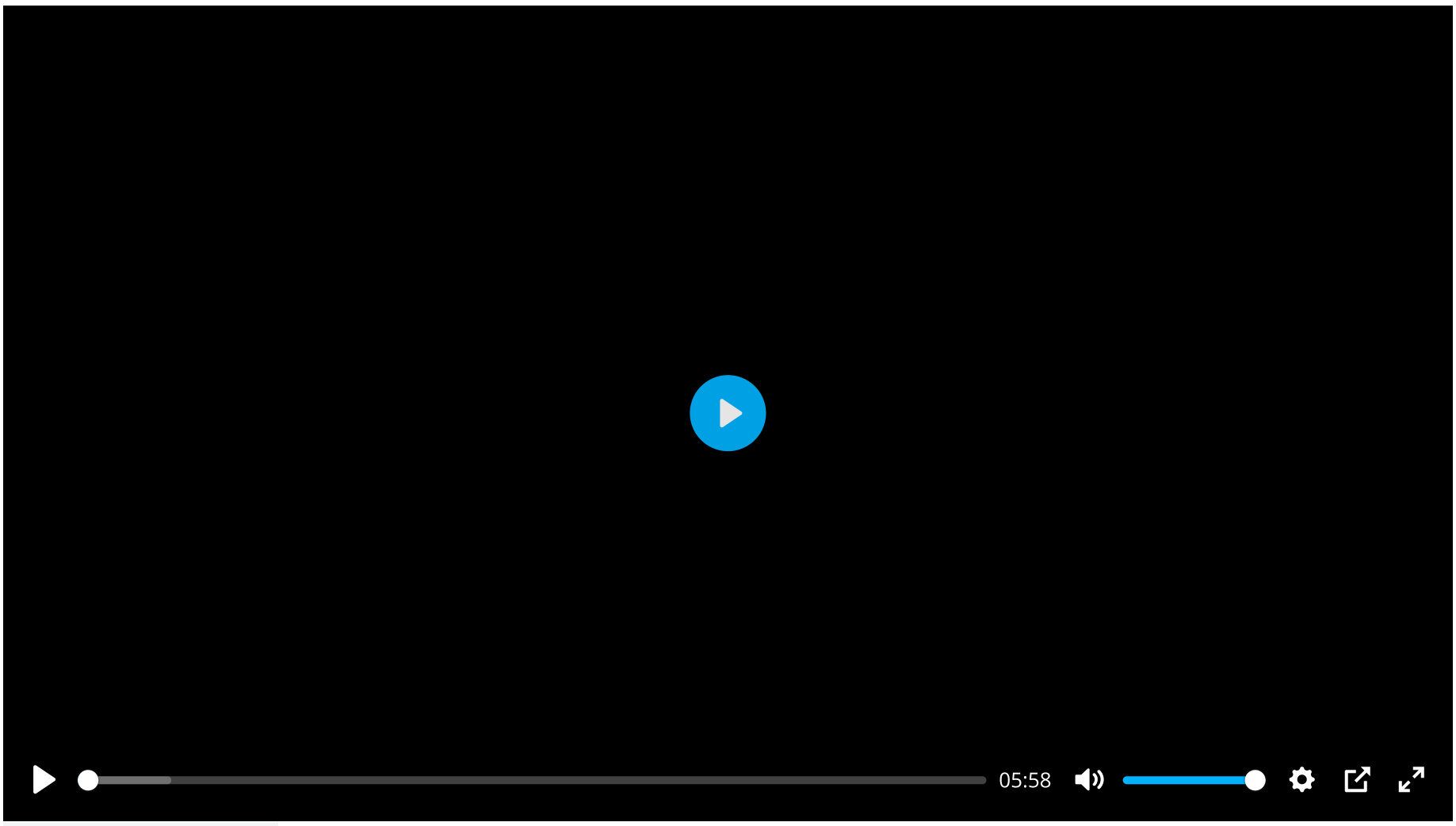
8.4 Работа с компонентами Airflow



Презентация к уроку 8.4.2

Текстовая расшифровка видео:

КОМПОНЕНТЫ AIRFLOW. METASTORE

План:

- Работа с базой, играющей роль Metastore.

Работа с базой, играющей роль Metastore

Metastore – это база, в которой хранятся данные по всему инстансу Airflow (пользователи, xcom'ы, информация о DAG Run'ах, информация о статусах тасков и т.д.).

Чтобы заглянуть в эту базу данных в нашем примере, когда мы запустились через docker compose в нашу базу postgres, необходимо выполнить **sodo docker** exec -it **airflow_postgres_1/bin/bash**, затем зайти в саму базу данных в контейнере через команду **psql -U** (юзер) **airflow** и посмотреть, какие есть таблицы.

Параметры подключения и юзер-пароль прописаны в docker-compose-файле.

Рассмотрим на практике. В docker-compose-файле есть настройки базы postgres:

```
services:
  postgres:
    image: postgres:13
    environment:
      POSTGRES_USER: airflow
      POSTGRES_PASSWORD: airflow
      POSTGRES_DB: airflow
    volumes:
      - postgres-db-volume:/var/lib/postgresql/data
    healthcheck:
      test: ["CMD", "pg_isready", "-U", "airflow"]
      interval: 10s
      retries: 5
      start_period: 5s
    restart: always
```

Здесь есть:

- image: postgres:13;
- переменные среды, в которых прописаны данные подключения Airflow;
- объем, где хранятся данные, связанные с контейнером;

- healthcheck, который проверяет запущенный контейнер на требования здоровья, и отвечает ли он на эту команду.

Когда мы прописываем **exec -it**, можно вместо **bin/bash** сразу запускать posql или делать их внутри контейнера. Мы видим следующие файлы внутри контейнера:

```
    profiles:
      - flower
    ports:
      - "5555:5555"
    healthcheck:
      test: ["CMD", "curl", "--fail", "http://localhost:5555/"]
      interval: 30s
      timeout: 10s
      retries: 5
      start_period: 30s
    restart: always
    depends_on:
      <<: *airflow-common-depends-on
      airflow-init:
        condition: service_completed_successfully

volumes:
  postgres-db-volume:
asya@asya-Aspire-XC-886:~/airflow_edu$ docker ps
CONTAINER ID   IMAGE                  COMMAND                  CREATED       STATUS                   PORTS
                  NAMES
a3b1b1a3a67e   apache/airflow:2.6.2   "/usr/bin/dumb-init …"   6 hours ago   Up 6 hours (healthy)   8080/tcp
                  airflow_edu-airflow-triggerer-1
a2d236165387   apache/airflow:2.6.2   "/usr/bin/dumb-init …"   6 hours ago   Up 6 hours (healthy)   8080/tcp
                  airflow_edu-airflow-scheduler-1
7b8c5176d225   apache/airflow:2.6.2   "/usr/bin/dumb-init …"   6 hours ago   Up 6 hours (healthy)   8080/tcp
                  airflow_edu-airflow-worker-1
9719aaac680f   apache/airflow:2.6.2   "/usr/bin/dumb-init …"   6 hours ago   Up 6 hours (healthy)   0.0.0.0:8080->8080/tcp, ::
:8080->8080/tcp   airflow_edu-airflow-webserver-1
2f6dc4b4a135   redis:latest           "docker-entrypoint.s…"   6 hours ago   Up 6 hours (healthy)   6379/tcp
                  airflow_edu-redis-1
60886edf6c3a   postgres:13            "docker-entrypoint.s…"   6 hours ago   Up 6 hours (healthy)   5432/tcp
                  airflow_edu-postgres-1
asya@asya-Aspire-XC-886:~/airflow_edu$ sudo docker exec -it  airflow_edu-postgres-1 /bin/bash
[sudo] password for asya:
root@60886edf6c3a:/# ls
bin   dev                         etc   lib    lib64   media  opt   root  sbin  sys  usr
boot  docker-entrypoint-initdb.d  home  lib32  libx32  mnt    proc  run   srv   tmp  var
root@60886edf6c3a:/#
```

Подключимся к базе данных:

```
asya@asya-Aspire-XC-886:~/airflow_edu$ sudo docker exec -it  airflow_edu-postgres-1 /bin/bash
[sudo] password for asya:
root@60886edf6c3a:/# ls
bin   dev                         etc   lib    lib64   media  opt   root  sbin  sys  usr
boot  docker-entrypoint-initdb.d  home  lib32  libx32  mnt    proc  run   srv   tmp  var
root@60886edf6c3a:/# psql -U airflow
psql (13.11 (Debian 13.11-1.pgdg120+1))
Type "help" for help.

airflow=#
```

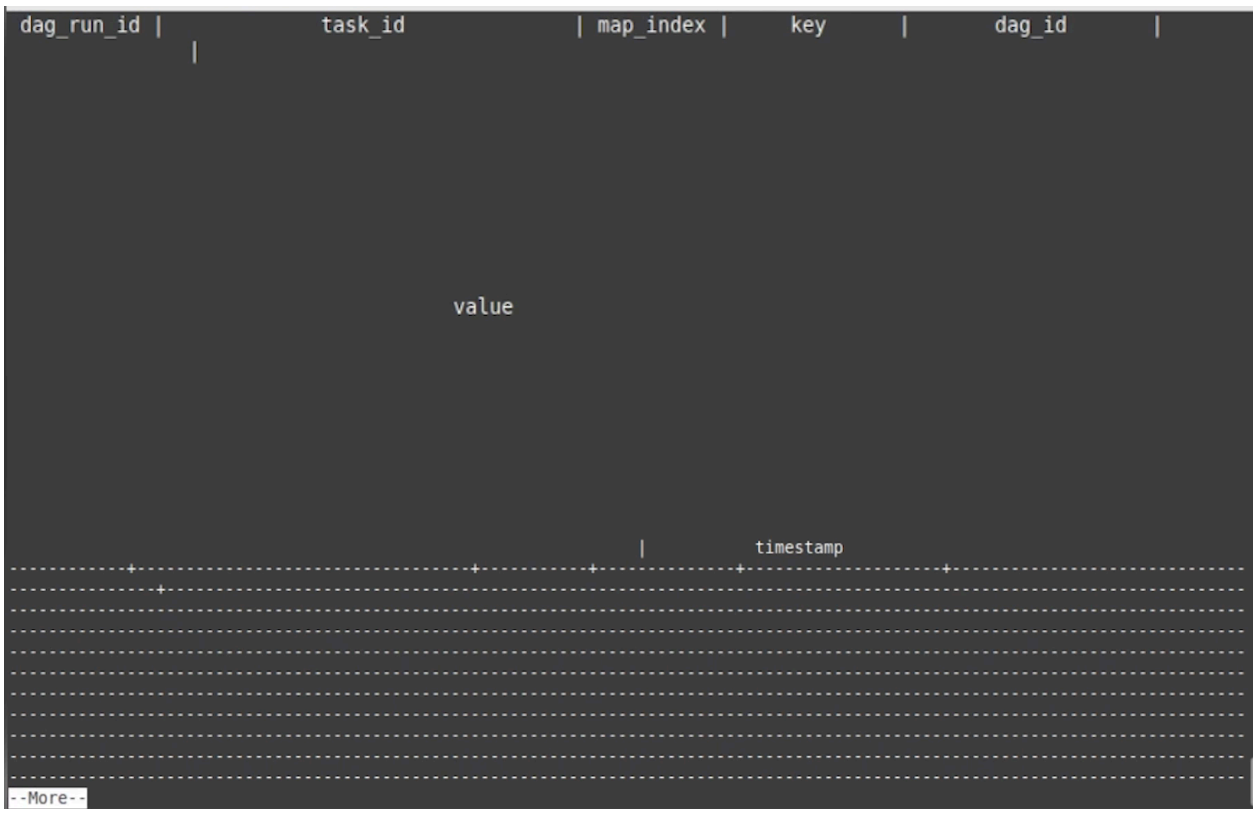
Смотрим все таблицы, с помощью пробела проматываем вниз.

Это все административные таблицы, связанные с инстансом Airflow:

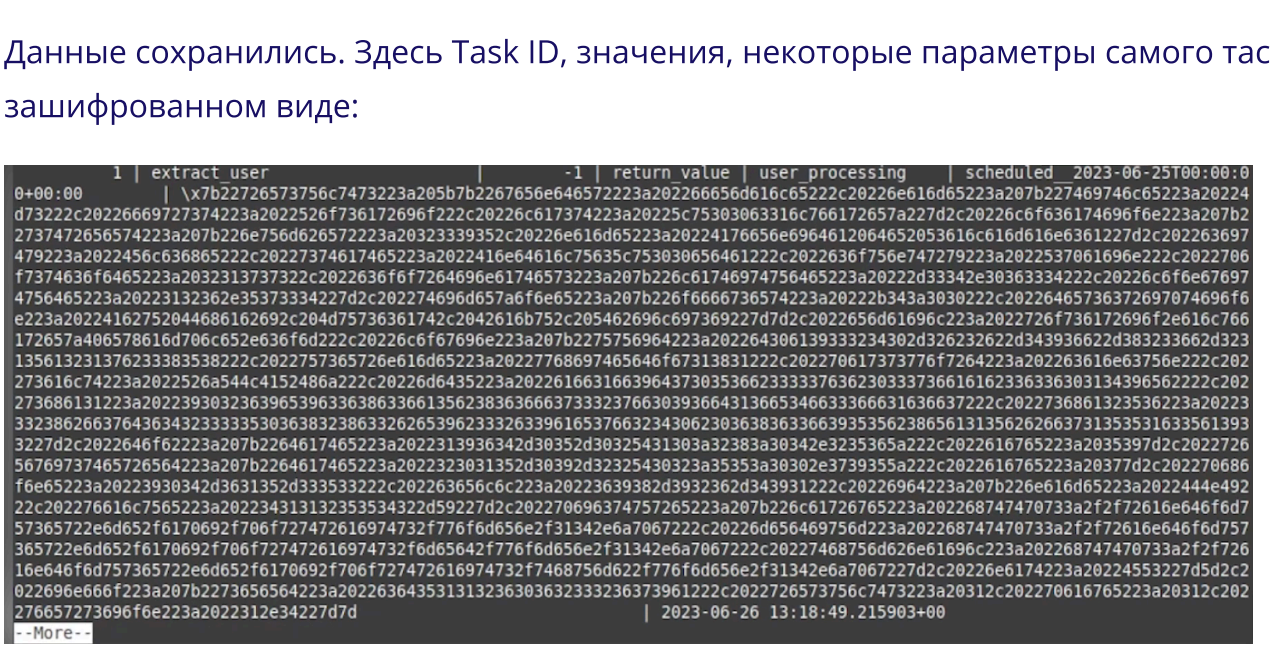
```
File  Edit  View  Search  Terminal  Help
                  List of relations
 Schema |              Name               | Type  |  Owner
--------+---------------------------------+-------+---------
 public | ab_permission                   | table | airflow
 public | ab_permission_view              | table | airflow
 public | ab_permission_view_role         | table | airflow
 public | ab_register_user                | table | airflow
 public | ab_role                         | table | airflow
 public | ab_user                         | table | airflow
 public | ab_user_role                    | table | airflow
 public | ab_view_menu                    | table | airflow
 public | alembic_version                 | table | airflow
 public | callback_request                | table | airflow
 public | celery_taskmeta                 | table | airflow
 public | celery_tasksetmeta              | table | airflow
 public | connection                      | table | airflow
 public | dag                             | table | airflow
 public | dag_code                        | table | airflow
 public | dag_owner_attributes            | table | airflow
 public | dag_pickle                      | table | airflow
 public | dag_run                         | table | airflow
 public | dag_run_note                    | table | airflow
 public | dag_schedule_dataset_reference  | table | airflow
 public | dag_tag                         | table | airflow
 public | dag_warning                     | table | airflow
 public | dagrun_dataset_event            | table | airflow
 public | dataset                         | table | airflow
 public | dataset_dag_run_queue           | table | airflow
 public | dataset_event                   | table | airflow
 public | import_error                    | table | airflow
 public | job                             | table | airflow
 public | log                             | table | airflow
 public | log_template                    | table | airflow
--More--
```

```
File  Edit  View  Search  Terminal  Help
 public | rendered_task_instance_fields   | table | airflow
 public | serialized_dag                  | table | airflow
 public | session                         | table | airflow
 public | sla_miss                        | table | airflow
 public | slot_pool                       | table | airflow
 public | task_fail                       | table | airflow
 public | task_instance                   | table | airflow
 public | task_instance_note              | table | airflow
 public | task_map                        | table | airflow
 public | task_outlet_dataset_reference   | table | airflow
 public | task_reschedule                 | table | airflow
 public | trigger                         | table | airflow
 public | users                           | table | airflow
 public | variable                        | table | airflow
 public | xcom                            | table | airflow
(45 rows)
```

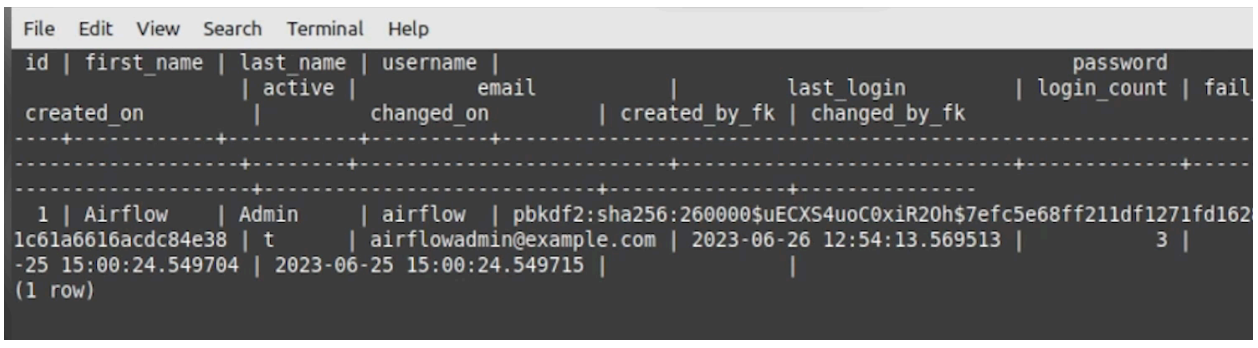
Посмотрим содержимое таблицы, например, xcom. После прошлого урока в xcom загрузились некоторые данные, проверим, сохранились ли они:



Данные сохранились. Здесь Task ID, значения, некоторые параметры самого таска. Значения хранятся в зашифрованном виде:

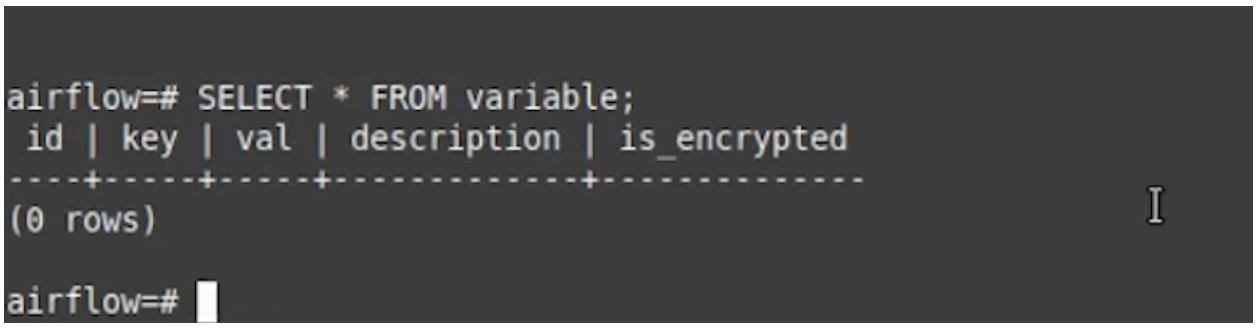


Посмотрим содержимое users:



Users – это та таблица, которую создавали мы, а **AB users** – таблица административной ID Airflow, здесь находятся данные юзеров, пользователей Airflow.

Рассмотрим таблицу с переменными. Переменные тоже хранятся в зашифрованном виде. Если вы указали, что они должны сохраняться **is_encrypted true**, то переменные должны шифроваться:



Чтобы выйти используем «**\q**» и выходим из контейнера.

Мы узнали, что такое Metastore-база и как в нее заглянуть, а также, какие данные в ней хранятся.