

[Презентация к уроку 8.2.4](#)

Текстовая расшифровка видео:

СОЗДАНИЕ ИНСТАНСА АРАСНЕ AIRFLOW

План:

- Установка через docker-compose;
- Запуск Airflow.

Установка через docker-compose

Airflow можно создавать разными способами:

- использовать сторонние библиотеки;
- использовать docker-контейнеры и собирать каждую часть отдельно;
- использовать docker-compose (данный способ работает на любой платформе и создаст для всех одинаковую среду).

Если вы создаете Airflow на своем устройстве, а не в контейнере, вам необходимо задать команду: «airflow db init». Это инициализирует базу данных, то есть создаст служебные таблицы в Metastore.

Подробнее с этим вы можете ознакомиться в [документации по установке](#) Airflow.

Шаги в установке:

- Создать директорию для инстанса Airflow;
- Поместить в нее файл docker-compose (скачать файл вы можете, перейдя по ссылке:

<https://airflow.apache.org/docs/apache-airflow/stable/howto/docker-compose/index.html#fetching-docker-compose-yaml>);

- Оставить расширение «.yaml»;

- Открыть терминал или CMD и перейти в эту директорию;
- Создать файл « .env » (если нет терминала, можно создать в другом ID);
- Внести в него переменную « AIRFLOW_UID=50000 »;
- В отдельном окне терминала выполнить команду « docker-compose up -d » (процесс будет выполняться на фоне);
- После развертывания ожидается результат healthy у всех контейнеров (ждем несколько минут) при выполнении команды « docker-compose ps »;

Healthy – это процедура, которая посылается запросом контейнера. Они прописаны в docker-compose-файле.

- Адрес локального Airflow на localhost: 8080;
- Чтобы приостановить инстанс, используйте команду « docker-compose down ».

Запуск Airflow

Для траблшутинга можно использовать команду « docker-compose ps », чтобы посмотреть все контейнеры и то, насколько они healthy.

Внутри контейнера можно заглянуть с помощью команды « docker logs имя_контейнера ».

Рассмотрим пример, перейдя в терминал:

Мы создадим директорию « airflow_edu », перейдем в нее и создадим файл « .env »:

```
File Edit View Search Terminal Help
asya@asya-Aspire-XC-886:~$ mkdir airflow_edu
asya@asya-Aspire-XC-886:~$ cd airflow_edu
asya@asya-Aspire-XC-886:~/airflow_edu$ nano .env
```


Если у вас данный файл не отображается, включите видимость всех файлов в папке.

Здесь задаем параметр « AIRFLOW_UID=50000 », сохраняем этот файл:

```
File Edit View Search Terminal Help
asya@asya-Aspire-XC-886:~$ mkdir airflow_edu
asya@asya-Aspire-XC-886:~$ cd airflow_edu
asya@asya-Aspire-XC-886:~/airflow_edu$ nano .env
asya@asya-Aspire-XC-886:~/airflow_edu$
```

Далее, скачиваем уатл-файл. Для этого можно посмотреть в документации: либо целиком скопировать данную ссылку, либо открыть docker-compose-файл, скачав содержимое (проще работать через ссылку):

← → ↻ <https://airflow.apache.org/docs/apache-airflow/stable/howto/docker-compose/index.html>

 Airflow

Fetching `docker-compose.yaml`

To deploy Airflow on Docker Compose, you should fetch [docker-compose.yaml](#).

```
curl -Lfo 'https://airflow.apache.org/docs/apache-airflow/2.6.2/docker-compose.yaml'
```

This file contains several service definitions:

- `airflow-scheduler` - The scheduler monitors all tasks and DAGs, then triggers the task instances once their dependencies are complete.
- `airflow-webserver` - The webserver is available at `http://localhost:8080`.
- `airflow-worker` - The worker that executes the tasks given by the scheduler.
- `airflow-triggerer` - The triggerer runs an event loop for deferrable tasks.
- `airflow-init` - The initialization service.
- `postgres` - The database.
- `redis` - The redis - broker that forwards messages from scheduler to worker.

Optionally, you can enable flower by adding `--profile flower` option, e.g. `docker compose --profile flower up`, or by explicitly specifying it on the command line e.g. `docker compose up flower`.

Мы видим, что файл был скачан:

```
File Edit View Search Terminal Help
asya@asya-Aspire-XC-886:~$ mkdir airflow_edu
asya@asya-Aspire-XC-886:~$ cd airflow_edu
asya@asya-Aspire-XC-886:~/airflow_edu$ nano .env
asya@asya-Aspire-XC-886:~/airflow_edu$ curl -Lfo 'https://airflow.apache.org/docs/apache-airflow/2.6.2/doc
% Total % Received % Xferd Average Speed Time Time Current
Dload Upload Total Spent Left Speed
100 11632 100 11632 0 0 19226 0 --:--:-- --:--:-- --:--:-- 19226
asya@asya-Aspire-XC-886:~/airflow_edu$
```

Запускаем инстанс.

В новых версиях «docker compose» пишется без чёрточки (-).

Ждем некоторое время. Сначала контейнеры создаются, потом airflow init инициализирует остальные компоненты airflow. После ожидания проверим их статус. Видим, что «redis» и «postgres» healthy:

```
asya@asya-Aspire-XC-886:~/airflow_edu$ docker compose up -d
[+] Building 0.0s (0/0)
[+] Running 7/8
✓ Network airflow_edu_default Created
✓ Container airflow_edu-redis-1 Healthy
✓ Container airflow_edu-postgres-1 Healthy
:: Container airflow_edu-airflow-init-1 Waiting
✓ Container airflow_edu-airflow-worker-1 Created
✓ Container airflow_edu-airflow-scheduler-1 Created
✓ Container airflow_edu-airflow-triggerer-1 Created
✓ Container airflow_edu-airflow-webserver-1 Created
```

Проверяем команды «docker ps» и «compose ps».

Видим, что не все контейнеры healthy. Программа пишет «health: starting», соответственно, нужно подождать:

```
asya@asya-Aspire-XC-886:~/airflow_edu$ docker compose ps
NAME IMAGE SERVICE CREATED STAT
airflow_edu-airflow-scheduler-1 apache/airflow:2.6.2 airflow-scheduler 54 seconds ago Up 2
5 seconds (health: starting) 8080/tcp
airflow_edu-airflow-triggerer-1 apache/airflow:2.6.2 airflow-triggerer 54 seconds ago Up 2
5 seconds (health: starting) 8080/tcp
airflow_edu-airflow-webserver-1 apache/airflow:2.6.2 airflow-webserver 54 seconds ago Up 2
4 seconds (health: starting) 0.0.0.0:8080->8080/tcp, :::8080->8080/tcp
airflow_edu-airflow-worker-1 apache/airflow:2.6.2 airflow-worker 54 seconds ago Up 2
5 seconds (health: starting) 8080/tcp
airflow_edu-postgres-1 postgres:13 postgres 54 seconds ago Up 5
2 seconds (healthy) 5432/tcp
airflow_edu-redis-1 redis:latest redis 54 seconds ago Up 5
2 seconds (healthy) 6379/tcp
asya@asya-Aspire-XC-886:~/airflow_edu$
```

Теперь мы видим, что все контейнеры healthy:

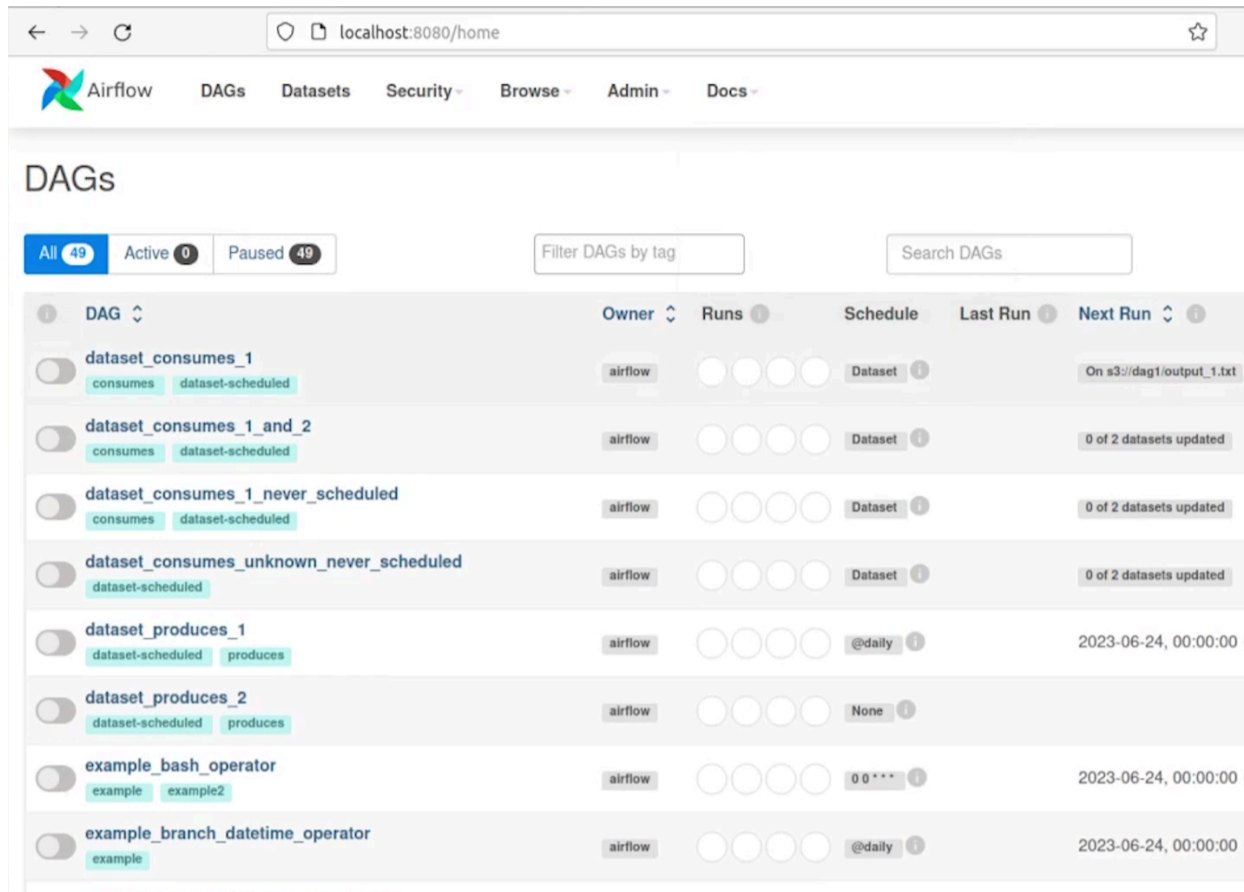
```

asya@asya-Aspire-XC-886:~/airflow_edu$ docker compose ps
NAME                                IMAGE                                COMMAND                                SERVICE    CREATED         STA
TUS
airflow_edu-airflow-scheduler-1    apache/airflow:2.6.2              "/usr/bin/dumb-init ..."         airflow-scheduler    About a minute ago    Up
43 seconds (healthy)              8080/tcp
airflow_edu-airflow-triggerer-1    apache/airflow:2.6.2              "/usr/bin/dumb-init ..."         airflow-triggerer    About a minute ago    Up
42 seconds (healthy)              8080/tcp
airflow_edu-airflow-webserver-1    apache/airflow:2.6.2              "/usr/bin/dumb-init ..."         airflow-webserver    About a minute ago    Up
42 seconds (healthy)              0.0.0.0:8080->8080/tcp, :::8080->8080/tcp
airflow_edu-airflow-worker-1       apache/airflow:2.6.2              "/usr/bin/dumb-init ..."         airflow-worker       About a minute ago    Up
42 seconds (healthy)              8080/tcp
airflow_edu-postgres-1            postgres:13                        "docker-entrypoint.s..."         postgres            About a minute ago    Up
About a minute (healthy)          5432/tcp
airflow_edu-redis-1              redis:latest                       "docker-entrypoint.s..."         redis               About a minute ago    Up
About a minute (healthy)          6379/tcp
asya@asya-Aspire-XC-886:~/airflow_edu$

```

Перейдем в Web-UI в localhost: 8080, где username и пароль – airflow.

Мы видим все Dags:



Это примеры, которые можно отключить в файле конфигурации (позже мы расскажем об этом подробнее).

Открыв Web-UI, пробуем траблшутинг:

```

asya@asya-Aspire-XC-886:~/airflow_edu$ docker compose ps
NAME                                IMAGE                                COMMAND                                SERVICE    CREATED         STA
TUS
airflow_edu-airflow-scheduler-1    apache/airflow:2.6.2              "/usr/bin/dumb-init ..."         airflow-scheduler    About a minute ago    Up
43 seconds (healthy)              8080/tcp
airflow_edu-airflow-triggerer-1    apache/airflow:2.6.2              "/usr/bin/dumb-init ..."         airflow-triggerer    About a minute ago    Up
42 seconds (healthy)              8080/tcp
airflow_edu-airflow-webserver-1    apache/airflow:2.6.2              "/usr/bin/dumb-init ..."         airflow-webserver    About a minute ago    Up
42 seconds (healthy)              0.0.0.0:8080->8080/tcp, :::8080->8080/tcp
airflow_edu-airflow-worker-1       apache/airflow:2.6.2              "/usr/bin/dumb-init ..."         airflow-worker       About a minute ago    Up
42 seconds (healthy)              8080/tcp
airflow_edu-postgres-1            postgres:13                        "docker-entrypoint.s..."         postgres            About a minute ago    Up
About a minute (healthy)          5432/tcp
airflow_edu-redis-1              redis:latest                       "docker-entrypoint.s..."         redis               About a minute ago    Up
About a minute (healthy)          6379/tcp
asya@asya-Aspire-XC-886:~/airflow_edu$ docker logs

```

Посмотрим логи контейнера «Scheduler»:

```

File Edit View Search Terminal Help
2 seconds (healthy)              5432/tcp
airflow_edu-redis-1            redis:latest                       "docker-entrypoint.s..."         redis
2 seconds (healthy)              6379/tcp
asya@asya-Aspire-XC-886:~/airflow_edu$ docker compose ps
NAME                                IMAGE                                COMMAND                                SERVICE
TUS
airflow_edu-airflow-scheduler-1    apache/airflow:2.6.2              "/usr/bin/dumb-init ..."         airflow-scheduler
43 seconds (healthy)              8080/tcp
airflow_edu-airflow-triggerer-1    apache/airflow:2.6.2              "/usr/bin/dumb-init ..."         airflow-triggerer
42 seconds (healthy)              8080/tcp
airflow_edu-airflow-webserver-1    apache/airflow:2.6.2              "/usr/bin/dumb-init ..."         airflow-webserver
42 seconds (healthy)              0.0.0.0:8080->8080/tcp, :::8080->8080/tcp
airflow_edu-airflow-worker-1       apache/airflow:2.6.2              "/usr/bin/dumb-init ..."         airflow-worker
42 seconds (healthy)              8080/tcp
airflow_edu-postgres-1            postgres:13                        "docker-entrypoint.s..."         postgres
About a minute (healthy)          5432/tcp
airflow_edu-redis-1              redis:latest                       "docker-entrypoint.s..."         redis
About a minute (healthy)          6379/tcp
asya@asya-Aspire-XC-886:~/airflow_edu$ docker logs airflow_edu-airflow-scheduler-1

BACKEND=redis
DB_HOST=redis
DB_PORT=6379

[2023-06-25T15:35:18.898+0000] {executor_loader.py:114} INFO - Loaded executor: CeleryExecutor
[2023-06-25T15:35:18.964+0000] {scheduler_job_runner.py:788} INFO - Starting the scheduler
[2023-06-25T15:35:18.965+0000] {scheduler_job_runner.py:795} INFO - Processing each file at most -1 times
[2023-06-25T15:35:18.969+0000] {manager.py:165} INFO - Launched DagFileProcessorManager with pid: 30
[2023-06-25T15:35:18.979+0000] {scheduler_job_runner.py:1553} INFO - Resetting orphaned tasks for active dag runs
[2023-06-25T15:35:18.979+0000] {settings.py:60} INFO - Configured default timezone Timezone('UTC')

```

Это логи его запуска.

Также мы можем зайти внутрь контейнера с помощью команды «`docker exec -it название_контейнера команда_для_входа`». В нашем случае команда для входа – «`/bin/bash`»:

```
asya@asya-Aspire-XC-886:~/airflow_edu$ docker exec -it airflow_edu-airflow-scheduler-1 /bin/bash
airflow@269fef7d8402:/opt/airflow$ ls
airflow.cfg  config  dags  logs  plugins  webserver_config.py
airflow@269fef7d8402:/opt/airflow$ exit
exit
asya@asya-Aspire-XC-886:~/airflow_edu$
```

Выключаем данный инстанс и пишем «`docker compose down`»:

```
asya@asya-Aspire-XC-886:~/airflow_edu$ docker compose down
[+] Running 8/8
✔ Container airflow_edu-airflow-webserver-1  Removed           3.9s
✔ Container airflow_edu-airflow-triggerer-1  Removed           1.6s
✔ Container airflow_edu-airflow-worker-1     Removed           2.7s
✔ Container airflow_edu-airflow-scheduler-1  Removed           2.4s
✔ Container airflow_edu-airflow-init-1       Removed           0.1s
✔ Container airflow_edu-postgres-1          Removed           0.4s
✔ Container airflow_edu-redis-1             Removed           0.3s
✔ Network airflow_edu_default               Removed           0.1s
asya@asya-Aspire-XC-886:~/airflow_edu$
```

Мы корректно завершили работу.

Как вам урок?



Изучил, далее >

Слёрм ©

[+7 \(495\) 248-05-80](tel:+74952480580)

[Лицензия №ДЛ-1368 от 22.08.2019](#)

[Политика конфиденциальности](#)

[Публичная оферта](#)

