

[Презентация к уроку 8.5.6](#)

Текстовая расшифровка видео:

МОНИТОРИНГ ЗАДАЧ В AIRFLOW

План:

- Flower Monitoring.

Flower Monitoring

Flower Monitoring доступен по адресу: **localhost:5555/dashboard**. В целом, он доступен, когда executor находится на нескольких нодах, в данном случае – это CeleryExecutor.

Откроем yaml-файл с docker compose и найдем там Flower (он уже встроен):



▶ Запустить стенд



Дедлайн 04 августа, 23:59 Мск



```
File Edit View Search Tools Documents Help
docker-compose.yml x
<<: *airflow-common-env
  _AIRFLOW_DB_UPGRADE: 'true'
  _AIRFLOW_WWW_USER_CREATE: 'true'
  _AIRFLOW_WWW_USER_USERNAME: ${_AIRFLOW_WWW_USER_USERNAME:-airflow}
  _AIRFLOW_WWW_USER_PASSWORD: ${_AIRFLOW_WWW_USER_PASSWORD:-airflow}
  PIP_ADDITIONAL_REQUIREMENTS: ''
user: "0:0"
volumes:
  - ${AIRFLOW_PROJ_DIR:-.}/sources

airflow-cli:
  <<: *airflow-common
  profiles:
    - debug
  environment:
    <<: *airflow-common-env
    CONNECTION_CHECK_MAX_COUNT: "0"
  # Workaround for entrypoint issue. See: https://github.com/apache/airflow/issues/16252
  command:
    - bash
    - -c
    - airflow

# You can enable flower by adding "--profile flower" option e.g. docker-compose --profile flower up
# or by explicitly targeted on the command line e.g. docker-compose up flower.
# See: https://docs.docker.com/compose/profiles/
flower:
  <<: *airflow-common
  command: celery flower
  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:
```

Здесь есть команда для запуска Flower Monitoring (это старая версия команды, нам нужна новая):

```
# You can enable flower by adding "--profile flower" option e.g. docker-compose --profile flower up
# or by explicitly targeted on the command line e.g. docker-compose up flower.
# See: https://docs.docker.com/compose/profiles/
flower:
  <<: *airflow-common
  command: celery flower
```

Новая команда выглядит следующим образом:

```
asya@asya-Aspire-XC-886:~/airflow_edu$ docker compose --profile flower up
[+] Building 0.0s (0/0)
[+] Running 9/9
✔ Network airflow_edu default Created
✔ Container airflow_edu-redis-1 Created
✔ Container airflow_edu-postgres-1 Created
✔ Container airflow_edu-airflow-init-1 Created
✔ Container airflow_edu-airflow-triggerer-1 Created
✔ Container airflow_edu-airflow-webserver-1 Created
✔ Container airflow_edu-airflow-worker-1 Created
✔ Container airflow_edu-airflow-scheduler-1 Created
✔ Container airflow_edu-flower-1 Created
Attaching to airflow_edu-airflow-init-1, airflow_edu-airflow-scheduler-1, airflow_edu-airflow-triggerer-1, airflow_edu-airflow-webserver-1, airflow_edu-airflow-worker-1, airflow_edu-flower-1
airflow_edu-redis-1 | 1:C 28 Jun 2023 14:48:57.660 # 0000000000000000 Redis is starting o000o000o000o
airflow_edu-redis-1 | 1:C 28 Jun 2023 14:48:57.660 # Redis version=7.0.11, bits=64, commit=00000000, modified=0, pid=1, just started
airflow_edu-redis-1 | 1:C 28 Jun 2023 14:48:57.660 # Warning: no config file specified, using the default config. In order to specify a config file
airflow_edu-redis-1 | use redis-server /path/to/redis.conf
airflow_edu-redis-1 | 1:M 28 Jun 2023 14:48:57.660 * monotonic clock: POSIX clock_gettime
airflow_edu-redis-1 | 1:M 28 Jun 2023 14:48:57.661 * Running mode=standalone, port=6379.
airflow_edu-redis-1 | 1:M 28 Jun 2023 14:48:57.661 # Server initialized
airflow_edu-redis-1 | 1:M 28 Jun 2023 14:48:57.661 # WARNING Memory overcommit must be enabled! Without it, a background save or replication may fail under low memory condition. Being disabled, it can also cause failures without low memory condition, see https://github.com/jemalloc/jemalloc/issues/1328
airflow_edu-redis-1 | . To fix this issue add 'vm.overcommit_memory = 1' to /etc/sysctl.conf and then reboot or run the command 'sysctl vm.overcommit_memory=1' for this to take effect.
airflow_edu-redis-1 | 1:M 28 Jun 2023 14:48:57.662 * Ready to accept connections
airflow_edu-postgres-1 | PostgreSQL Database directory appears to contain a database; Skipping initialization
airflow_edu-postgres-1 | 2023-06-28 14:48:57.780 UTC [1] LOG: starting PostgreSQL 13.11 (Debian 13.11-1.pgdg120+1) on x86_64-pc-linux-gnu, compiled by gcc (Debian 12.2.0-14) 12.2.0, 64-bit
airflow_edu-postgres-1 | 2023-06-28 14:48:57.780 UTC [1] LOG: listening on IPv4 address "0.0.0.0", port 5432
airflow_edu-postgres-1 | 2023-06-28 14:48:57.780 UTC [1] LOG: listening on IPv6 address ":::", port 5432
airflow_edu-postgres-1 | 2023-06-28 14:48:57.790 UTC [1] LOG: listening on Unix socket "/var/run/postgresql/.s.PGSQL.5432"
airflow_edu-postgres-1 | 2023-06-28 14:48:57.801 UTC [27] LOG: database system was shut down at 2023-06-28 14:38:08 UTC
airflow_edu-postgres-1 | 2023-06-28 14:48:57.824 UTC [1] LOG: database system is ready to accept connections
```

Docker compose - -profile flower up

Ждем запуск. Если до этого у вас был запущен docker compose, сделайте Docker compose down, а затем команду выше.

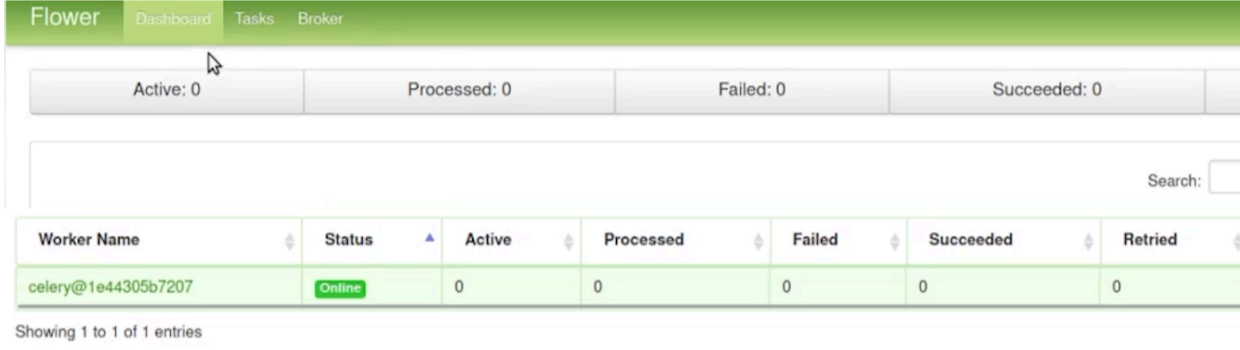
Здесь прописан Celery Executor:

```

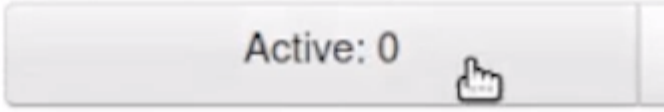
File Edit View Search Tools Documents Help
docker-compose.yml x
# AIRFLOW_IMAGE_NAME - Docker image name used to run Airflow.
# Default: apache/airflow:2.6.2
# AIRFLOW_UID - User ID in Airflow containers
# Default: 50000
# AIRFLOW_PROJ_DIR - Base path to which all the files will be volumed.
# Default: .
# Those configurations are useful mostly in case of standalone testing/running Airflow in test/try-out mode
#
# _AIRFLOW_WWW_USER_USERNAME - Username for the administrator account (if requested).
# Default: airflow
# _AIRFLOW_WWW_USER_PASSWORD - Password for the administrator account (if requested).
# Default: airflow
# PIP_ADDITIONAL_REQUIREMENTS - Additional PIP requirements to add when starting all containers.
# Use this option ONLY for quick checks. Installing requirements at container
# startup is done EVERY TIME the service is started.
# A better way is to build a custom image or extend the official image
# as described in https://airflow.apache.org/docs/docker-stack/build.html.
# Default: ''
#
# Feel free to modify this file to suit your needs.
---
version: '3.8'
x-airflow-common:
  &airflow-common
  # In order to add custom dependencies or upgrade provider packages you can use your extended image.
  # Comment the image line, place your Dockerfile in the directory where you placed the docker-compose.yml
  # and uncomment the "build" line below. Then run 'docker-compose build' to build the images.
  image: ${AIRFLOW_IMAGE_NAME:-apache/airflow:2.6.2}
  # build: .
  environment:
    &airflow-common-env
    AIRFLOW_CORE_EXECUTOR: CeleryExecutor
    AIRFLOW_DATABASE__SQL_ALCHEMY_CONN: postgresql+psycopg2://airflow:airflow@postgres/airflow
    # For backward compatibility, with Airflow <2.3
    AIRFLOW_CORE__SQL_ALCHEMY_CONN: postgresql+psycopg2://airflow:airflow@postgres/airflow
    AIRFLOW_CELERY__RESULT_BACKEND: db+postgresql://airflow:airflow@postgres/airflow
    AIRFLOW_CELERY__BROKER_URL: redis://:@redis:6379/0
    AIRFLOW_CORE__FERNET_KEY: ''
    AIRFLOW_CORE__DAGS_ARE_PAUSED_AT_CREATION: 'true'
    AIRFLOW_CORE__LOAD_EXAMPLES: 'true'
    AIRFLOW_API__AUTH_BACKENDS: 'airflow.api.auth.backend.basic_auth,airflow.api.auth.backend.session'
    # yamllint disable rule:line-length
    # Use simple http server on scheduler for health checks
    # See https://airflow.apache.org/docs/apache-airflow/stable/administration-and-deployment/logging-monitoring/check-health.html#scheduler-health-check-server
    # yamllint enable rule:line-length
    AIRFLOW_SCHEDULER__ENABLE_HEALTH_CHECK: 'true'

```

Эти переменные считаются более первичными по отношению к тем, которые находятся в файле конфига. По адресу открылся localhost:5555/dashboard, где мы видим Celery Executor. Здесь находятся информация о воркерах и его загрузки:



Это количество активных задач:



Вверху информация о задачах и их статусах:



Это воркер (он один, так как одна нода):



Также здесь информация о его статусе, количестве задач в разном статусе.

Последнее поле – это загрузка за последнюю минуту (5 минут, 15 в процентах для данного воркера):



Worker:
celery@1e44305b7207

Pool | **Broker** | Queues | Tasks | Limits | Config | System | Other

Worker pool options

Max concurrency	16
Processes	34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49
Max tasks per child	N/A
Put guarded by semaphore	False
Timeouts	0, 0
Writes	{'total': 0, 'avg': '0.00', 'all': '', 'raw': '', 'strategy': 'fair', 'inqueues': {'total': 16, 'active': 0}}
Worker PID	6
Prefetch Count	16

Pool size control

Pool size: 1 | Grow | Shrink

Min/Max autoscale: | | Apply

- Максимальная конкуренция;
- Какое количество задач может выполняться одновременно в воркере;
- Какие процессы соответствуют и т.д.

Это раздел, где размещена информация о брокере:

Worker:
celery@1e44305b7207

Pool | **Broker** | Queues | Tasks | Limits | Config | System | Other

Broker options

Hostname	redis
Userid	None
Virtual host	0
Port	6379
Insist	False
SSL	False
Transport	redis
Connect timeout	4
Transport options	{'visibility_timeout': 21600}
Login method	None
URI prefix	None
Heartbeat	120.0
Failover strategy	round-robin

Здесь находится информация о задачах:

Flower | Dashboard | **Tasks** | Broker

Show 10 entries | Search:

Name	UUID	State	args	kwargs	Result	Received	Started	Runtime
No data available in table								

Showing 0 to 0 of 0 entries

Запустим какой-нибудь DAG и посмотрим, как будут пробегать задачи через наш мониторинг. Запустим **xcom_edu_dag**. В Flower Monitoring происходят изменения:

celery@1e44305b7207	Online	1	2	0	1	0	2.96, 2.5, 2.51
---------------------	--------	---	---	---	---	---	-----------------

Showing 1 to 1 of 1 entries

Виден Processed. В самом воркере видно, как меняется количество задач. Всего их два, они проходят через Processed и доходят до статуса «успешно выполнены».

Это наш брокер:

Broker: **redis://redis:6379/0**

Queues

Name	Messages	Unacked	Ready	Consumers	Idle
------	----------	---------	-------	-----------	------

Смотрим по задачам:

celery@1e44305b7207	Online	0	2	0	2	0	3.6, 2.65, 2.56
---------------------	--------	---	---	---	---	---	-----------------

Showing 1 to 1 of 1 entries

Смотрим внутри:

Worker:
celery@1e44305b7207

Pool | Broker | Queues | Tasks | Limits | Config | System | Other

Worker pool options

Max concurrency	16
Processes	34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49
Max tasks per child	N/A
Put guarded by semaphore	False
Timeouts	0, 0
Writes	{'total': 0, 'avg': '0.00', 'all': '', 'raw': '', 'strategy': 'fair', 'inqueues': {'total': 16, 'active': 0}}
Worker PID	6
Prefetch Count	16

Pool size control

Pool size: 1 | Grow | Shrink

Min/Max autoscale: | | Apply

В воркерах:

Worker:
celery@1e44305b7207

Pool | Broker | Queues | Tasks | Limits | Config | System | Other

Broker options

Hostname	redis
Userid	None
Virtual host	0
Port	6379
Insist	False
SSL	False
Transport	redis
Connect timeout	4
Transport options	{'visibility_timeout': 21600}
Login method	None
URI prefix	None

Попробуем увидеть задачи, когда они проходят. После того, как был запущен дважды, появляются четыре экземпляра задач, то есть на каждый на каждый запуск DAG по одному:

Name	UUID	State	args	kwargs	Res
airflow.executors.celery_executor.execute_command	13390a0d-82c7-4a42-bdbf-e15610778213	SUCCESS	[[airflow, tasks, run, xcom_edu_dag, fetching_data, manual_2023-06-28T14:50:04.631981+00:00, --local, --subdir, DAGS_FOLDER/xcom_edu.py]]	{}	None
airflow.executors.celery_executor.execute_command	863938e0-c916-4453-9328-193da4adff32	SUCCESS	[[airflow, tasks, run, xcom_edu_dag, downloading_data, manual_2023-06-28T14:50:04.631981+00:00, --local, --subdir, DAGS_FOLDER/xcom_edu.py]]	{}	None
airflow.executors.celery_executor.execute_command	7946228a-bd78-4581-87c3-111e30da8eb7	SUCCESS	[[airflow, tasks, run, xcom_edu_dag, fetching_data, manual_2023-06-28T14:49:07.956028+00:00, --local, --subdir, DAGS_FOLDER/xcom_edu.py]]	{}	None
airflow.executors.celery_executor.execute_command	301e3caf-c897-4b12-8917-bb48e0f01c5	SUCCESS	[[airflow, tasks, run, xcom_edu_dag, downloading_data, manual_2023-06-28T14:49:07.956028+00:00, --local, --subdir, DAGS_FOLDER/xcom_edu.py]]	{}	None

Showing 1 to 4 of 4 entries

Запустим более длительно выполняющийся DAG. Здесь отображаются все задачи со своими статусами:

airflow.executors.celery_executor.execute_command	13390a0d-82c7-4a42-bdbf-e15610778213	SUCCESS	[[airflow, 'tasks', 'run', 'xcom_edu_dag', 'fetching_data', 'manual__2023-06-28T14:50:04.631981+00:00', '--local', '--subdir', 'DAGS_FOLDER/xcom_edu.py]]	{}	None	2023-06-28 14:50:08.212	2023-06-14:50:08
airflow.executors.celery_executor.execute_command	863938e0-c916-4453-9328-193da4adff32	SUCCESS	[[airflow, 'tasks', 'run', 'xcom_edu_dag', 'downloading_data', 'manual__2023-06-28T14:50:04.631981+00:00', '--local', '--subdir', 'DAGS_FOLDER/xcom_edu.py]]	{}	None	2023-06-28 14:50:05.568	2023-06-14:50:05
airflow.executors.celery_executor.execute_command	7946228a-bd78-4581-87c3-111e30da8eb7	SUCCESS	[[airflow, 'tasks', 'run', 'xcom_edu_dag', 'fetching_data', 'manual__2023-06-28T14:49:07.956028+00:00', '--local', '--subdir', 'DAGS_FOLDER/xcom_edu.py]]	{}	None	2023-06-28 14:49:13.222	2023-06-14:49:13
airflow.executors.celery_executor.execute_command	301e3caf-c897-4b12-8917-bb48e0f01c5	SUCCESS	[[airflow, 'tasks', 'run', 'xcom_edu_dag', 'downloading_data', 'manual__2023-06-28T14:49:07.956028+00:00', '--local', '--subdir', 'DAGS_FOLDER/xcom_edu.py]]	{}	None	2023-06-28 14:49:09.176	2023-06-14:49:09

Showing 1 to 6 of 6 entries

Previous 1 Next

Здесь также находится информация о DAG Run, о воркере. Более наглядно эту информацию можно увидеть, когда у нас несколько воркеров. Однако даже на примере одного воркера мы увидели, как выглядят задачи, когда проходят через очередь и воркер.

Мы поговорили про мониторинг задач, про ветвление, про файл-конфигурации. На этом изучение Apache Airflow окончено!

Желаем вам успешного прохождения заданий, удачи!

Как вам урок?



Изучил, далее >>

Слёрм ©

+7 (495) 248-05-80

[Лицензия №ДЛ-1368 от 22.08.2019](#)

[Политика конфиденциальности](#)

[Публичная оферта](#)