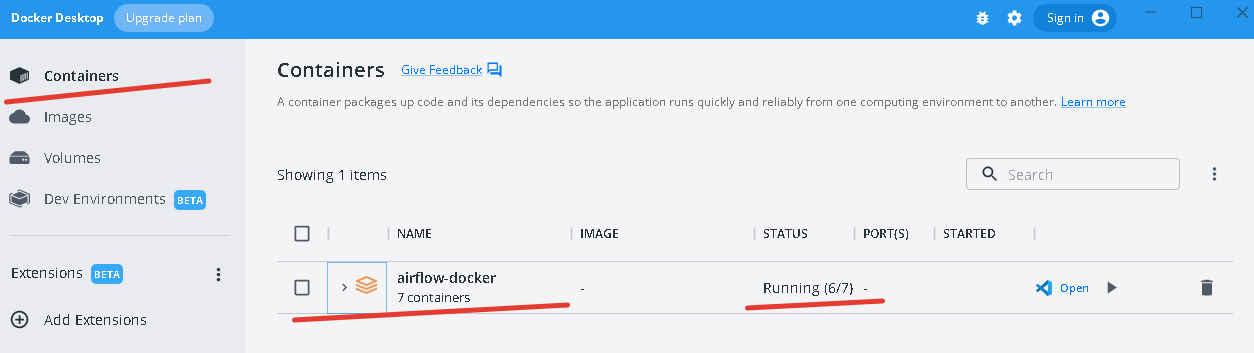
**Запуск Airflow из контейнера Docker.**

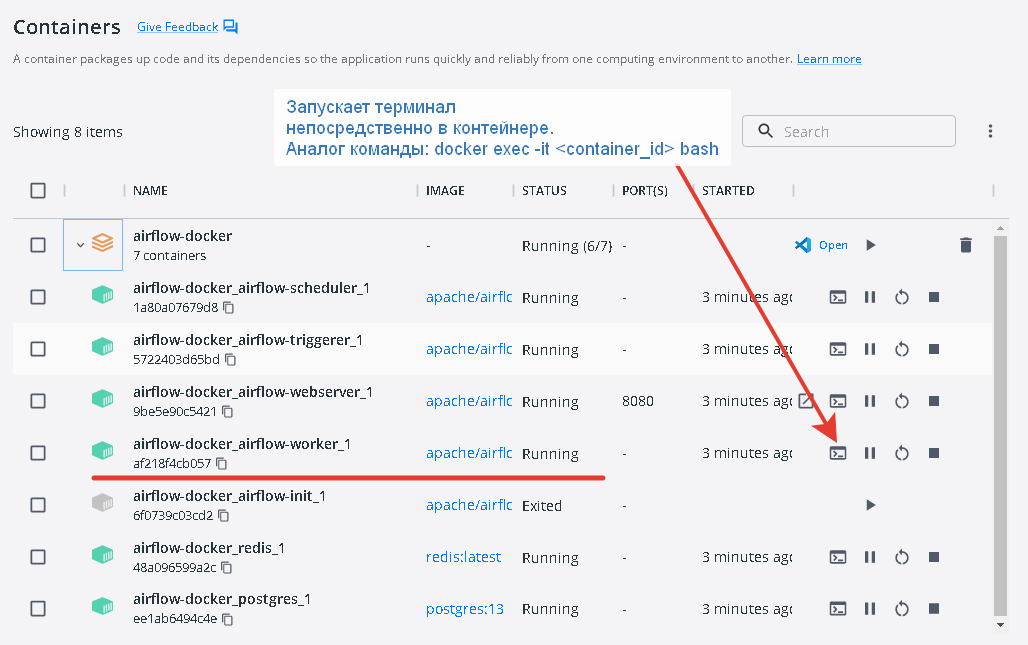
1. Необходимо наглядно проверять работоспособность сервера, запущенного внутри контейнера и контролировать наличие изменений, которые будут внесены в этот контейнер.

Наиболее доступный способ – воспользоваться терминалом Docker desktop.

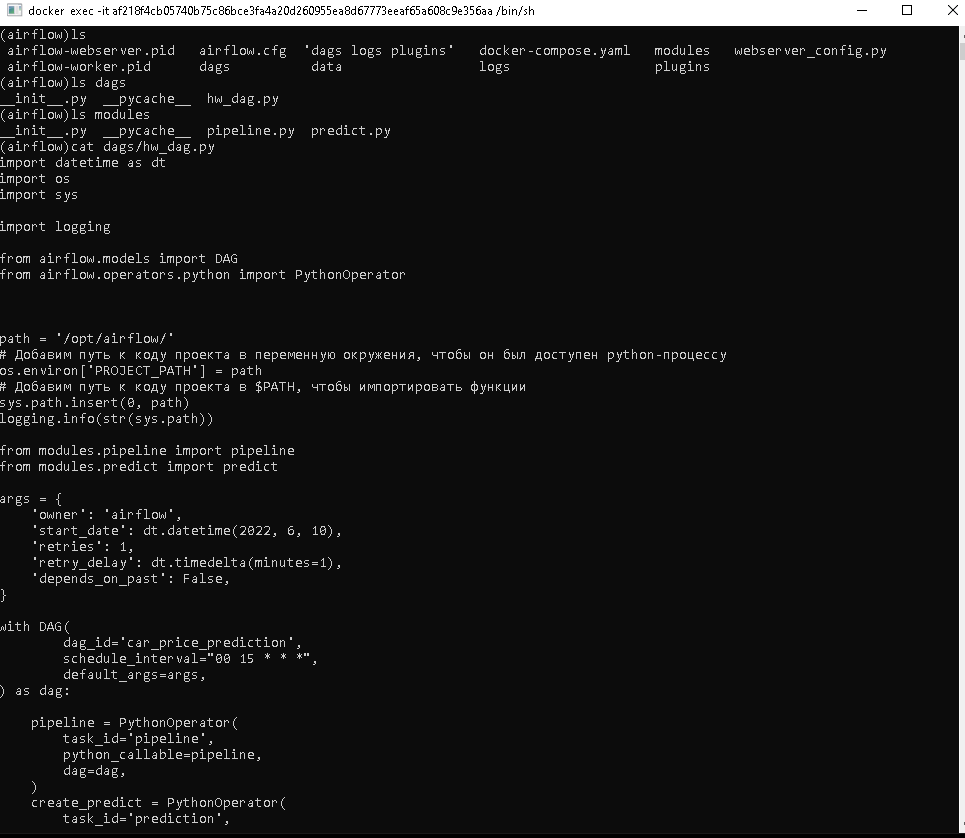
Далее предполагаем, что в предыдущей инструкции (2. Установка Airflow), был запущен Airflow, командой docker-compose up создан контейнер с сервисами.

Проверить это мы можем в приложении Docker desktop.



Внутри контейнера ожидаем увидеть следующее: 

Необходимо привести в соответствие структуру файлов в контейнере. Для этого использовать команды ls. Проверить содержание файла можно с помощью команды cat.



Также необходимо уделить внимание настройке доступного нам файла docker-compose. yawl. В данном репозитории необходимые настройки выполнены. Остаётся лишь проверить их соответствие. Итак, еще **ДО**  запуска контейнеров мы можем указать какие папки будут скопированы в контейнер. Иными словами, определить путь откуда взять папки и куда положить.

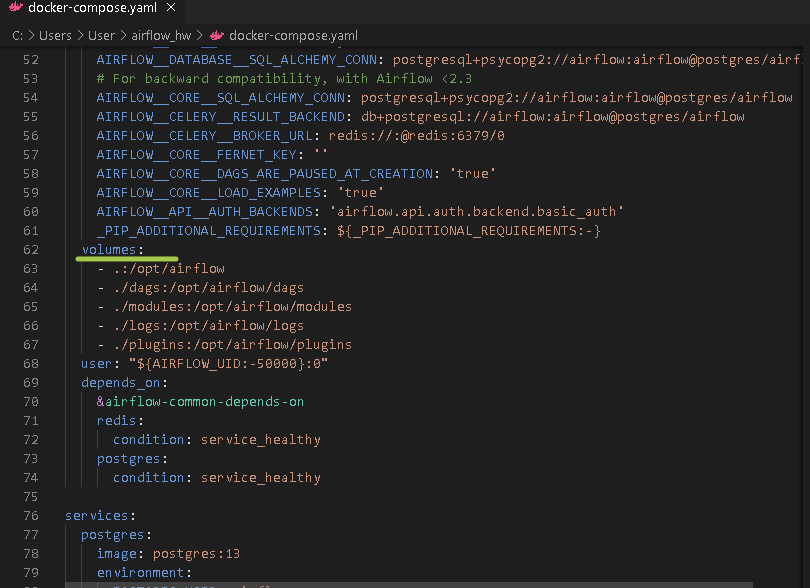
Интересующая нас область в файле - volumes:

Синтаксис в определении пути следующий:

- <путь1>:<путь2>

путь1 это путь на нашем компьютере

путь2 это путь внутри контейнера



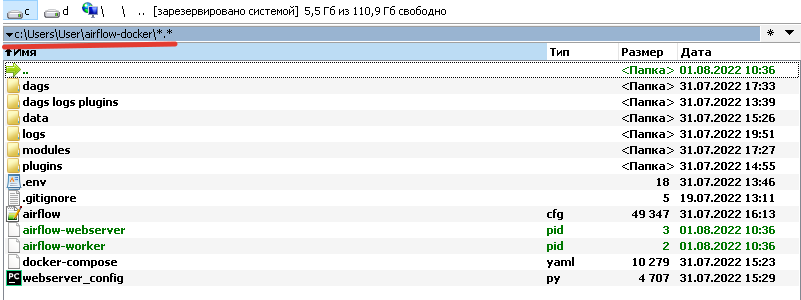
.:/opt/airflow -- путь1 тут это "." - текущая директория (из которой выполняется

запуск docker compose up

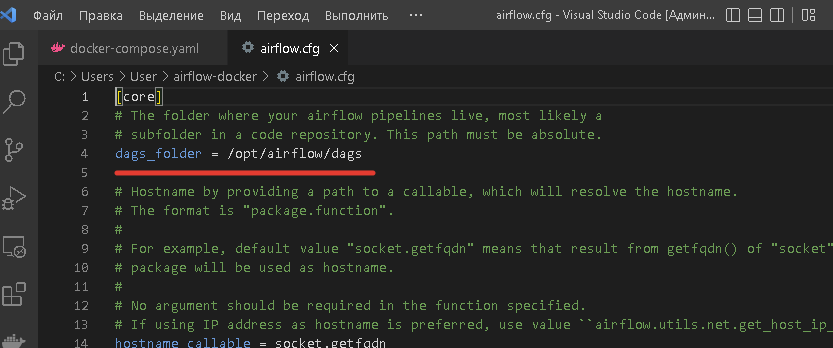
путь2 это /opt/airflow

Так выглядит текущая директория с docker-compose.yaml

Пример расположения:



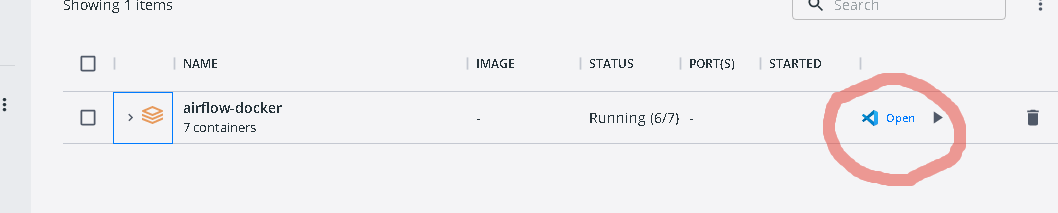
Далее необходимо в самом Airflow указать пути, с которыми будет работать сервис. Для этого требуется проверить изменения в файле airflow.cfg



Необходимо проверить изменения пути во всём файле airflow.cfg.

Проверив соответствие файлов, необходимо запустить/перезапустить контейнер.

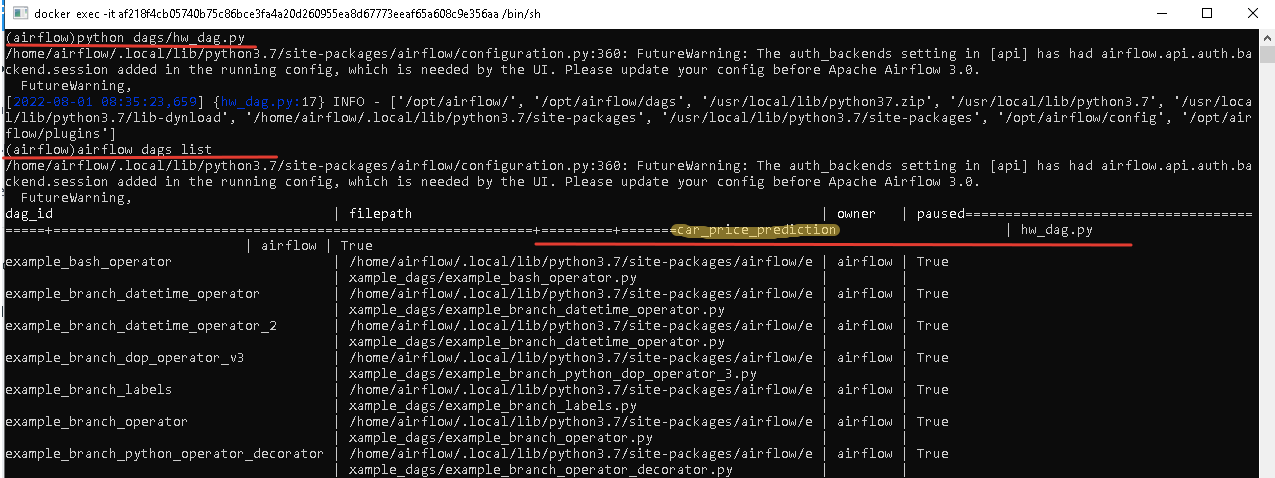
Самый доступный способ сделать это, воспользоваться приложением Docker-desktop.



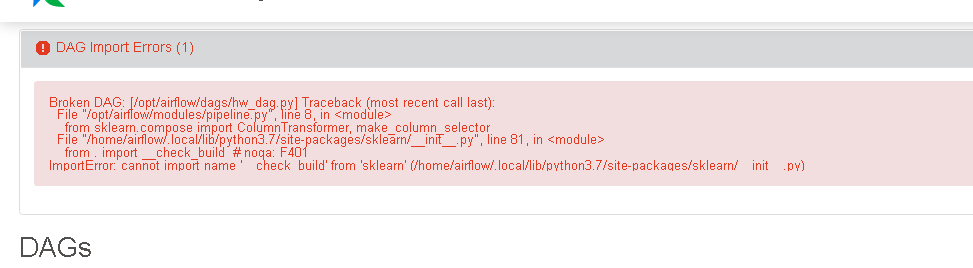
Нажимаем на треугольник и ждём некоторое время запуска сервисов.

Дале переходим в терминал worker и выполняем две проверки:

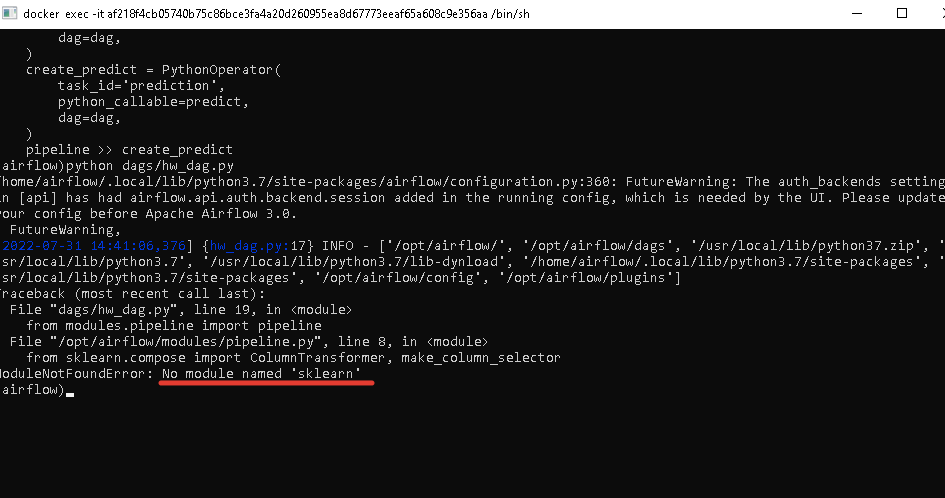
1. Запускаем даг файл напрямую, чтобы убедиться в отсутствии ошибок.
2. Выводим список даг-файлов, в котором должен быть hw\_dag.py с названием car\_price\_prediction.



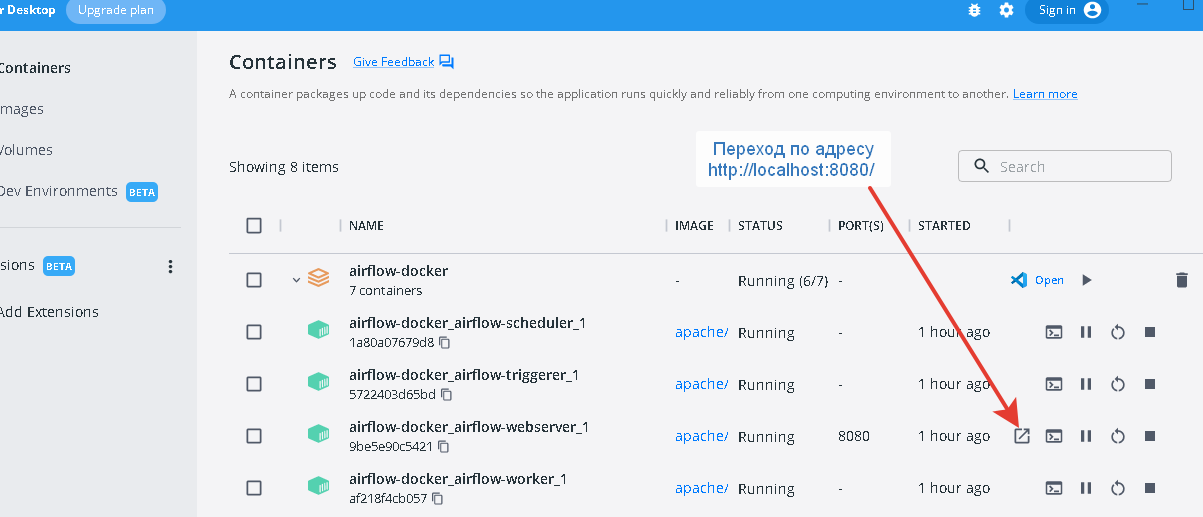
Типовая ошибка, на данном этапе – уведомление об ошибке, отсутствие библиотеки.

Уведомление в веб интерфейсе Airflow, например: 

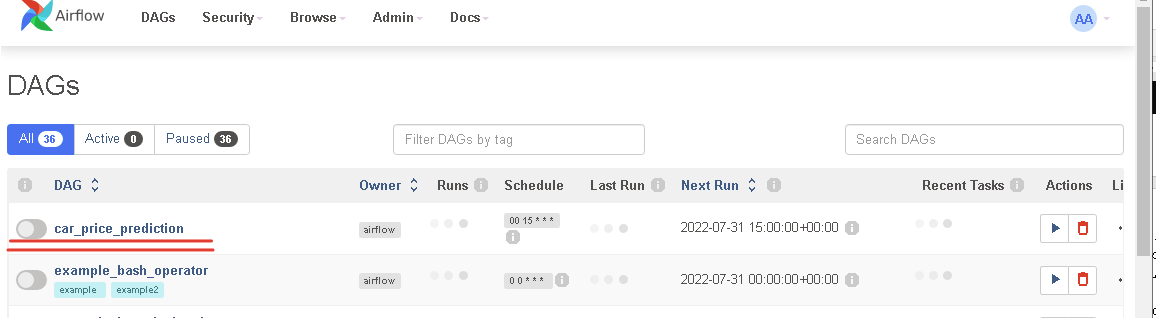
Тоже уведомление в терминале, контейнера worker airflow:



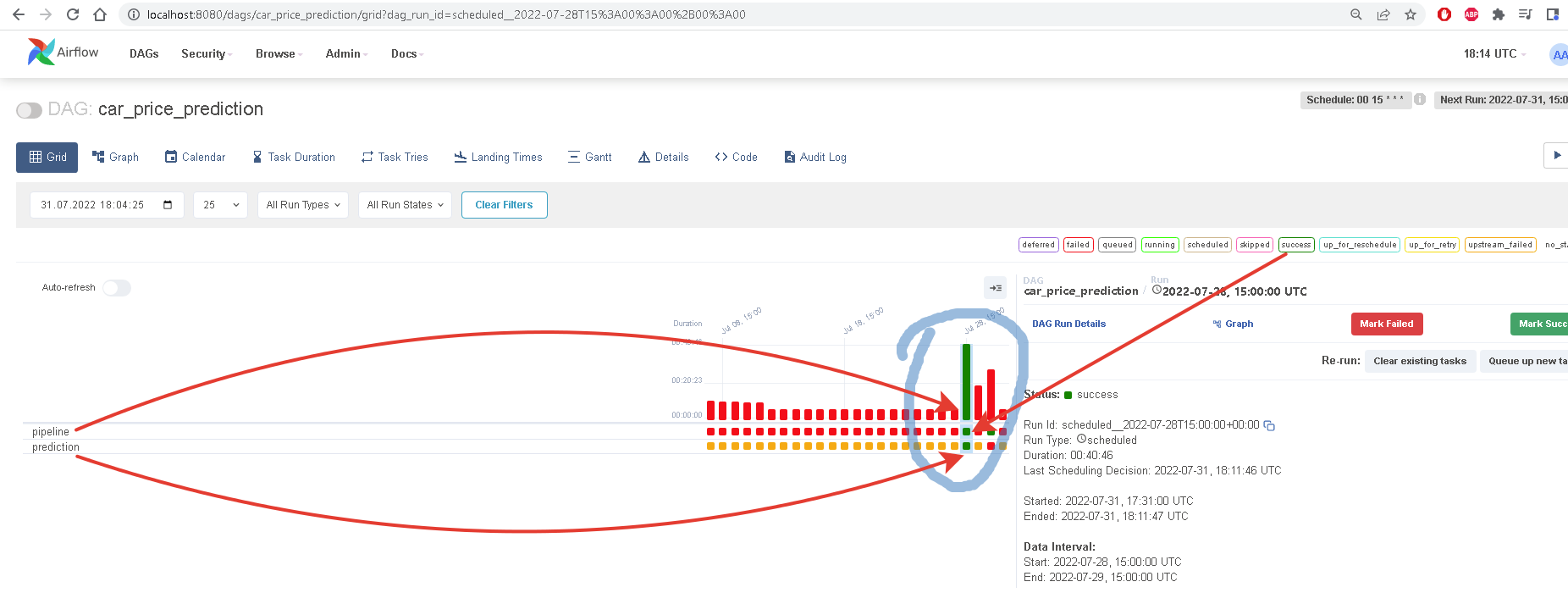
Решить задачу отсутствия требуемых библиотек необходимо путём установки этих библиотек **в два контейнера worker, а также scheduler.** В данном примере команда pip install sklearn. Важно выполнять установку непосредственно в контейнеры Airflow.   
  
Финальный шаг – переход запуск и переход в веб интерфейс Airflow. Самый доступный способ – воспользоваться графический интерфейсом Docker-desktop.



Требуемые настройки совершены и в веб-интерфейсе Airflow обнаружим готовый к исполнению даг-файл car\_price\_prediction.



Запускаем наш даг-файл и даём некоторое время Airflow для работы.



Возможно, потребуется несколько попыток для успешного выполнения нашей задачи, соответственно будет затрачено время. Достаточно получить одну успешную итерацию для достижения цели. Результат - csv-файл в контейнере worker проверить наличие которого, возможно следующим способом:

