Оркестрация скриптов работы с данными

Цель: освоить работу с Apache Airflow для автоматизации задач

Описание задания:

Ранее была сделана выгрузка из API customs_data.csv и разово обогащена данными налоговой службы https://www.nalog.gov.ru/rn77/program/5961290/ классификатором ТН ВЭД.

- Теперь этот архив с сайта в ZIP-формате необходимо сохранить в зависимости от того как запущен инстанс Airflow - если без Docker, то на диске, если через Docker, то в контейнер с worker.
- Необходимо создать DAG в Apache Airflow под названием customs_enriched , который с помощью оператора-сенсора будет проверять появление файла в директории, где лежит архив.

В ходе выполнения DAG также должен меняться параметр `relevant_date` в Variables Airflow (в коде), где хранится дата на которую данные актуальны (“Дата актуальности” на сайте).

При автоматизации скачивания необходимо также считать эту строчку с сайта и поместить в переменную `relevant_date`

Если дата актуальности не обновилась, дальнейшие действия не требуются. Если дата актуальности обновилась, необходимо повторять процедуру обогащения данных, используя уже скачанный CSV customs_data.csv, выполнять MapReduce-пайплайн из 3 модуля в DAG, в итоге произведя вставку обогащенных данных в БД Postgres или Clickhouse на выбор. Важно, что эта БД отличается от metastore базы Airflow-инстанса, то есть вам необходимо указать в Airflow Connection данные внешней БД.

```
Подсказка если БД развернута в Docker контейнере, не находящемся в одной подсети с Airflow (в docker-compose нет этой БД), то IP контейнера можно узнать при помощи команды:

docker inspect имя_контейнера | grep IPAddress
```

Ответом на задание является ссылка на Git-репозиторий, где лежит код DAG `customs_enriched`.

Инструменты, которые пригодятся для выполнения: SQL, Apache Airflow, Python, Docker, Git.

Дополнительные задания (опционально):

Задача со звездочкой №1: скачивать данные не вручную, а автоматизировать скачивание, используя к примеру библиотеку Selenuim https://selenium-python.readthedocs.io/ и сделать скачивание таском Airflow.

Задача со звездочкой №2: В следующей таске в том же DAG выполнять агрегацию исходных данных как в 3 модуле в DAG любым инструментом по желанию (запросами в БД, MapReduce-пайплайном через BashOperator), в итоге произвести вставку обогащенных данных в ту же БД где исходные данные, в другую таблицу — `econ.customs_enriched`

Критерии, по которым будет оцениваться задание:

1. DAG `customs_enriched` корректно проверяет нахождение файла в директории в контейнере и заполняет параметр `relevant_date` актуальным значением с сайта.
2. В БД поступают данные
3. Автоматизация процеолнения таблицы `econ.customs_enriched` будут давать преимущество при оценк