**Задание**

С помощью Docker-compose развернуть среду из 4 контейнеров.

Контейнер 1. PostgreSQL

Контейнер 2. Spark – master

Контейнер 3. Spark – worker1

Контейнер 4. Spark – worker2

Источник данных – файл house\_prices.csv

Схема обработки данных и взаимодействия контейнеров:

1) «Контейнер 1. PostgreSQL» запускается и создает (подключается к существующей) БД.

2) В БД подгружаются данные из файла house\_prices.csv в соответствующую таблицу.

3) «Контейнер 2. Spark – master» имеет возможность подключения к БД.

4) Контейнеры 2, 3 и 4 настроены как кластер Spark.

5) Приложение PySpark в виде файла my\_spark.py загружено в «Контейнер 2. Spark – master». Его задача:

*Из данных* house\_prices *вывести информацию: Средние стоимости квартир и домов в зависимости от района и количества комнат.*

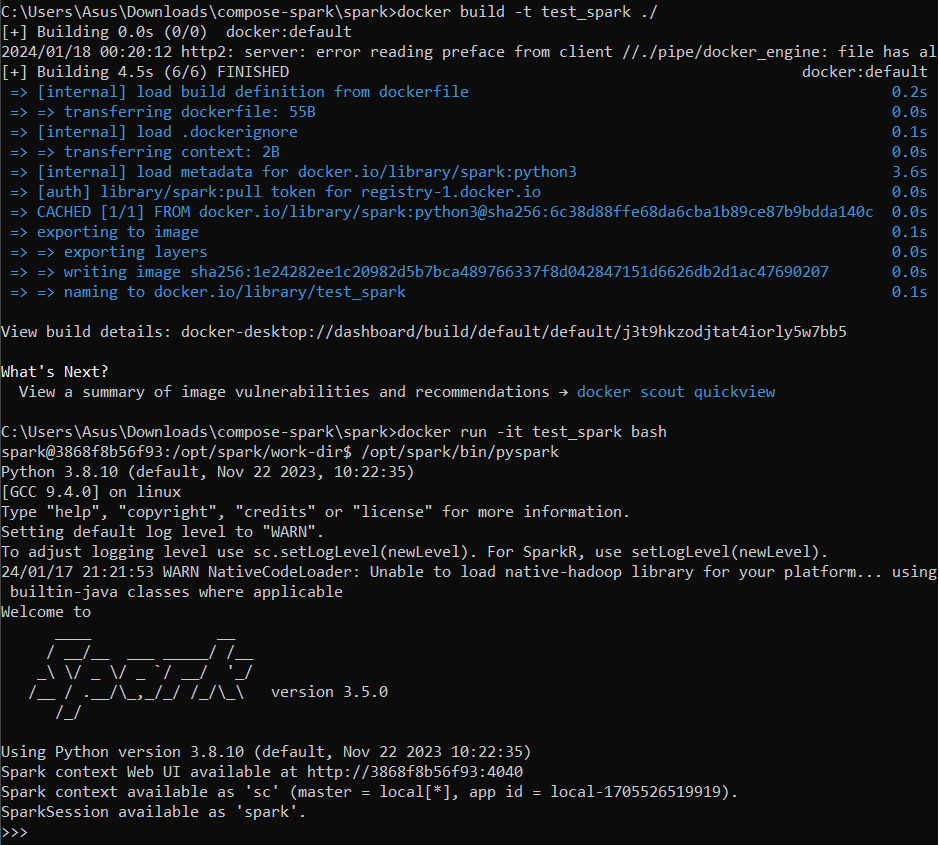
ОТЧЕТНОСТЬ:

На почту преподавателю необходимо отправить ссылку на Ваш проект в GitHub и скрин экрана с результатом работы приложения my\_spark.py.

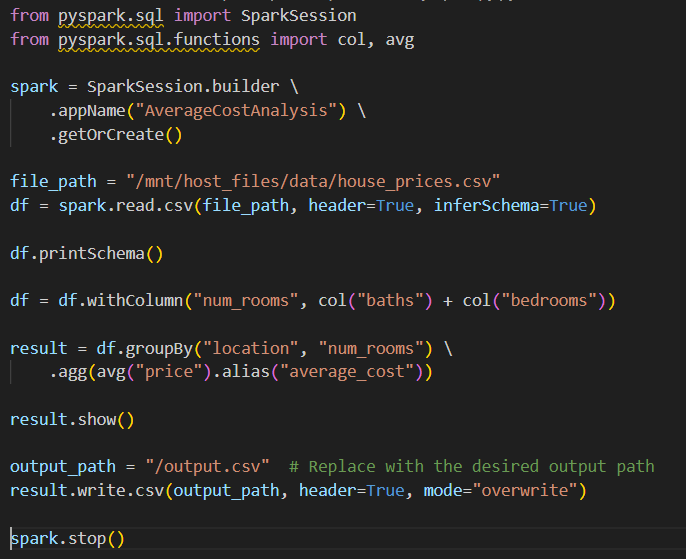
========================================

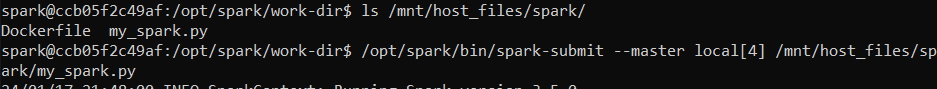
Работу можно разделить для удобства обучения на 4 этапа:

1. Поднимаем Spark контейнер



2. Пишем и тестируем my\_spark.py с загрузкой из файла



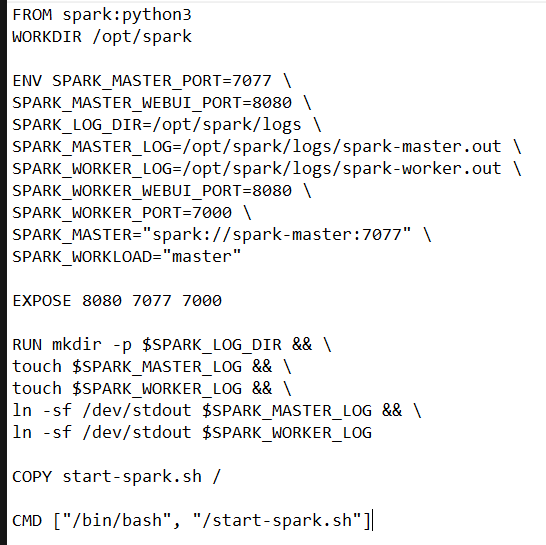


Изображение выглядит как текст, снимок экрана, меню, Шрифт

Автоматически созданное описание

3. Делаем чтобы спарк работал в кластере master+2workers

Dockerfile:



start-spark.sh

#!/bin/bash

. "/opt/spark/bin/load-spark-env.sh"

if [ "$SPARK\_WORKLOAD" == "master" ];

then

export SPARK\_MASTER\_HOST=`hostname`

cd /opt/spark/bin && ./spark-class org.apache.spark.deploy.master.Master --ip $SPARK\_MASTER\_HOST --port $SPARK\_MASTER\_PORT --webui-port $SPARK\_MASTER\_WEBUI\_PORT >> $SPARK\_MASTER\_LOG

elif [ "$SPARK\_WORKLOAD" == "worker" ];

then

cd /opt/spark/bin && ./spark-class org.apache.spark.deploy.worker.Worker --webui-port $SPARK\_WORKER\_WEBUI\_PORT $SPARK\_MASTER >> $SPARK\_WORKER\_LOG

elif [ "$SPARK\_WORKLOAD" == "submit" ];

then

    echo "SPARK SUBMIT"

else

    echo "Undefined Workload Type $SPARK\_WORKLOAD, must specify: master, worker, submit"

fi

docker-compose.yml

version: "3.9"

services:

  spark-master:

    build: ./

    container\_name: spark-cl-m

    ports:

      - "9090:8080"

      - "7077:7077"

    volumes:

       - ./apps:/opt/spark-apps

       - ./data:/opt/spark-data

    environment:

      - SPARK\_LOCAL\_IP=spark-master

      - SPARK\_WORKLOAD=master

  spark-worker-a:

    build: ./

    container\_name: spark-cl-w1

    ports:

      - "9091:8080"

      - "7000:7000"

    depends\_on:

      - spark-master

    environment:

      - SPARK\_MASTER=spark://spark-master:7077

      - SPARK\_WORKER\_CORES=1

      - SPARK\_WORKER\_MEMORY=1G

      - SPARK\_DRIVER\_MEMORY=1G

      - SPARK\_EXECUTOR\_MEMORY=1G

      - SPARK\_WORKLOAD=worker

      - SPARK\_LOCAL\_IP=spark-worker-a

    volumes:

       - ./apps:/opt/spark-apps

       - ./data:/opt/spark-data

  spark-worker-b:

    build: ./

    container\_name: spark-cl-w2

    ports:

      - "9092:8080"

      - "7001:7000"

    depends\_on:

      - spark-master

    environment:

      - SPARK\_MASTER=spark://spark-master:7077

      - SPARK\_WORKER\_CORES=1

      - SPARK\_WORKER\_MEMORY=1G

      - SPARK\_DRIVER\_MEMORY=1G

      - SPARK\_EXECUTOR\_MEMORY=1G

      - SPARK\_WORKLOAD=worker

      - SPARK\_LOCAL\_IP=spark-worker-b

    volumes:

        - ./apps:/opt/spark-apps

        - ./data:/opt/spark-data

  demo-database:

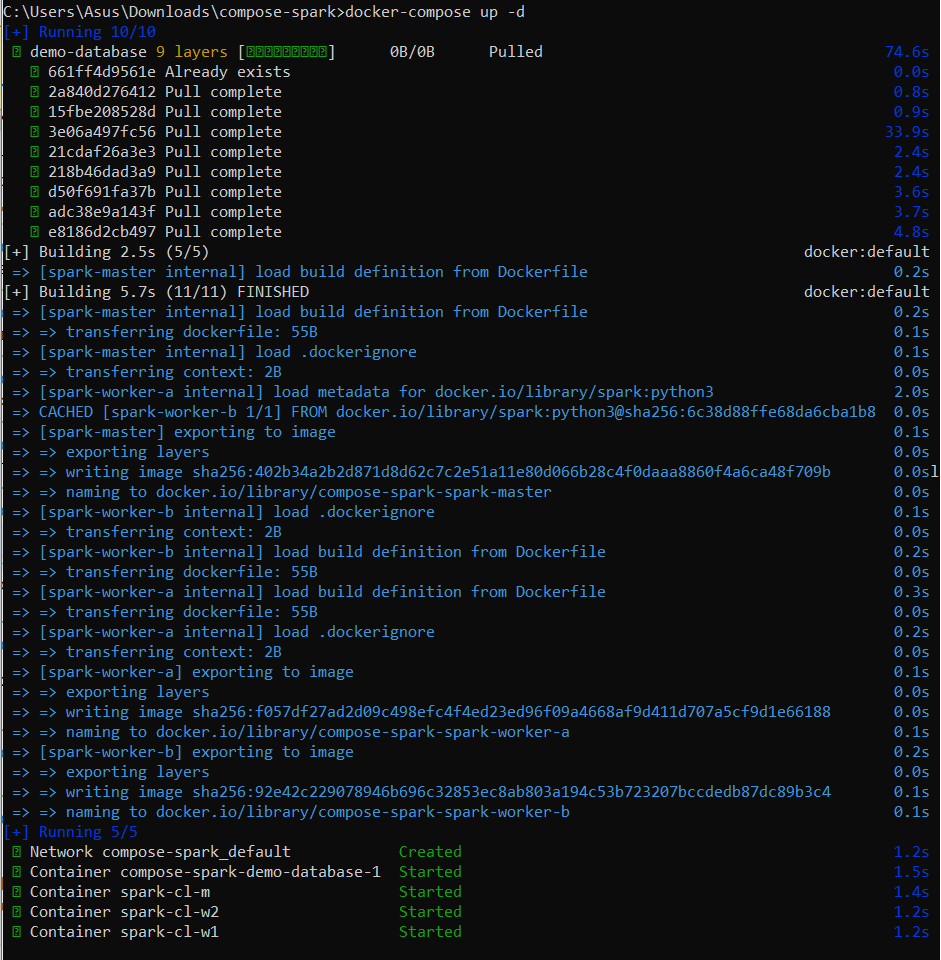
    image: postgres:alpine

    ports:

      - "5432:5432"

    environment:

      - POSTGRES\_PASSWORD=dfltpwd



Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Автоматически созданное описание

4. Заменяем подгрузку с файла на подгрузку из бд PostgreSQL

CREATE TABLE real\_estate (

id SERIAL PRIMARY KEY,

property\_type VARCHAR(255),

price INT,

location VARCHAR(255),

city VARCHAR(255),

baths INT,

purpose VARCHAR(255),

bedrooms INT,

area\_in\_marla FLOAT

);

COPY real\_estate(id,property\_type, price, location, city, baths, purpose, bedrooms, area\_in\_marla)

FROM '/data/house\_prices.csv' CSV HEADER DELIMITER ',';

++ в docker-compose.yml

    environment:

      - POSTGRES\_USER=postgres

      - POSTGRES\_PASSWORD=dfltpwd

      - POSTGRES\_DB=spark\_house

    volumes:

      - ./data:/data

      - ./db/init.sql:/docker-entrypoint-initdb.d/init.sql