# 1. Thu thập dữ liệu

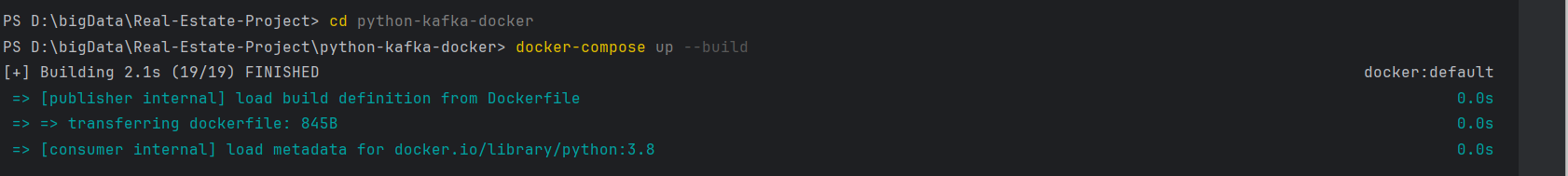
A. Thiết lập môi trường Docker

Chuyển đến thư mục dự án Docker

PS D:\bigData\Real-Estate-Project> cd python-kafka-docker

Khởi động Docker Compose với tùy chọn build

PS D:\bigData\Real-Estate-Project\python-kafka-docker> docker-compose up --build



Kiểm tra lại môi trường docker

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

Thông tin

 **Kafdrop**

* **Image**: obsidiandynamics/kafdrop:3.27.0
* **Chức năng**: Kafdrop là một giao diện web để quản lý và giám sát Kafka. Nó cung cấp các tính năng như duyệt topic, xem chi tiết message, và giám sát consumer group.
* **Port**: 19000:9000 - Truy cập giao diện web Kafdrop qua port 19000 trên host.

 **Publisher**

* **Image**: python-kafka-docker-publisher
* **Chức năng**: Đây là một ứng dụng Python đóng vai trò là publisher, gửi message đến Kafka topic.
* **Port**: 8000:8000 - Ứng dụng này có thể cung cấp một API hoặc giao diện khác trên port 8000.

 **Zookeeper**

* **Image**: confluentinc/cp-zookeeper:6.2.0
* **Chức năng**: Zookeeper cung cấp dịch vụ quản lý cấu hình và đồng bộ hóa cho Kafka. Nó theo dõi trạng thái của Kafka cluster nodes và duy trì metadata cho Kafka topics và partitions.
* **Port**: 2181:2181 - Zookeeper chạy trên port 2181.

 **Consumer**

* **Image**: python-kafka-docker-consumer
* **Chức năng**: Đây là một ứng dụng Python đóng vai trò là consumer, nhận và xử lý message từ Kafka topic.
* **Port**: 8001:8001 - Ứng dụng này có thể cung cấp một API hoặc giao diện khác trên port 8001.

 **MongoDB**

* **Image**: mongo
* **Chức năng**: MongoDB là cơ sở dữ liệu NoSQL được sử dụng để lưu trữ dữ liệu. Ứng dụng này có thể sử dụng MongoDB để lưu trữ các message hoặc dữ liệu xử lý từ Kafka.
* **Port**: 27017:27017 - MongoDB chạy trên port 27017.

 **Kafka**

* **Image**: confluentinc/cp-kafka:6.2.0
* **Chức năng**: Kafka là hệ thống messaging phân tán, được sử dụng để truyền tải message giữa các dịch vụ (như giữa publisher và consumer). Kafka lưu trữ và phát tán message thông qua các topic.
* **Port**: 9092:9092 - Kafka chạy trên port 9092.

B.Tiến hành thu thập dữ liệu

Chuyển đến thư mục chứa công cụ crawler

(.venv) PS D:\bigData\Real-Estate-Project> cd crawlerData

Chạy chương trình crawler với các tham số đầu vào

(.venv) PS D:\bigData\Real-Estate-Project\crawlerData> python crawler.py 1 10

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động

Kiểm tra

python test\_crawler.py

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phần mềm đa phương tiện

Mô tả được tạo tự động

# 2. Hướng dẫn sử dụng Docker Spark với dự án "Spark Projects with Docker"

**Các phiên bản hỗ trợ:**

* **Spark 3.0.0** cho Hadoop 3.2 với OpenJDK 8 và Scala 2.12
* **Spark 2.4.5** cho Hadoop 2.7+ với OpenJDK 8

#### Cách bắt đầu:

1. **Khởi động Docker Compose**

Chạy lệnh sau trong thư mục chứa tệp docker-compose.yml

docker-compose up

Ảnh có chứa văn bản, Phông chữ, phần mềm, ảnh chụp màn hình

Mô tả được tạo tự động

Ảnh có chứa văn bản, ảnh chụp màn hình, số, phần mềm

Mô tả được tạo tự động

**Truy cập giao diện web của Spark Master và các Worker**:

* Spark Master: <http://localhost:8080> Ảnh có chứa văn bản, ảnh chụp màn hình, hàng, số

  Mô tả được tạo tự động
* Worker 1: <http://localhost:8081>

Ảnh có chứa văn bản, ảnh chụp màn hình, hàng

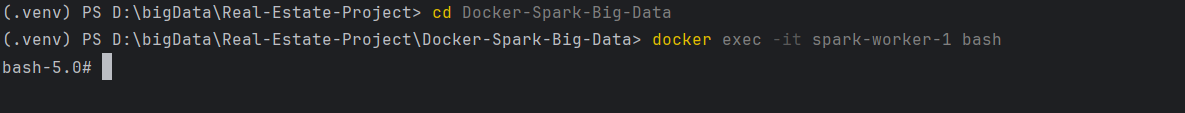
Mô tả được tạo tự động

#### Thực thi container với worker 1:

* Để truy cập vào container của worker 1, chạy lệnh sau:

Để truy cập vào container của worker 1, chạy lệnh sau:

docker exec -it spark-worker-1 bash



#### Các ví dụ Python:

* **Chạy giao diện dòng lệnh của PySpark**:

./spark/bin/pyspark

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động

* **Thực thi một tệp Python**:

cd /home/python/example

./../../../spark/bin/spark-submit example.py data.csv

Ảnh có chứa ảnh chụp màn hình, văn bản, Phông chữ

Mô tả được tạo tự động

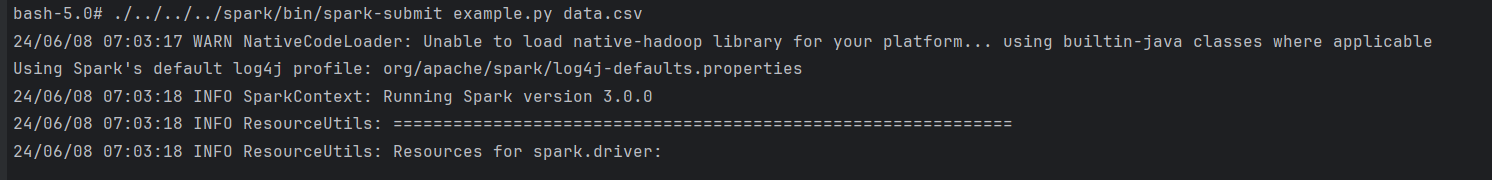
**Các ví dụ Python:**

**docker exec -it spark-worker-1 bash**

**bash-5.0# cd /home/python/example1**

**./../../../spark/bin/spark-submit example.py data.csv**

****

****

**Ảnh có chứa văn bản, Phông chữ, ảnh chụp màn hình, thiết kế

Mô tả được tạo tự động**

#### Theo dõi Spark:

* Truy cập các liên kết sau để theo dõi Spark:
  + <http://localhost:4040>
  + <http://localhost:4041>

#### Cài đặt thêm các công cụ cần thiết:

* Cài đặt gcc và notebook:

apk add gcc

pip3 install notebook

# 3,Dùng Spark lấy dữ liệu từ Topic Kafka gửi lên máy chủ mongoDB

Ảnh có chứa văn bản, Phông chữ, ảnh chụp màn hình

Mô tả được tạo tự động

**docker exec -it spark-worker-1 bash**

cd /home/python/Project

./../../../spark/bin/spark-submit \

> --packages org.apache.spark:spark-sql-kafka-0-10\_2.12:3.0.0 \

> batDongSan.py