# Đề tài : Mô phỏng dữ liệu Forex theo thời gian thực

## Nhóm 7:

1. Nguyễn Văn Huy 20170080
2. Nguyễn Ngọc Thịnh 20173386
3. Dương Hồng Sơn 20173347
4. Nguyễn Thanh Nhã 20170103

## Giới thiệu về đề tài

Hiện nay thị trường ngoại hối (forex) rât phát triển, kéo theo rất nhiều nhà đầu tư tham gia. Nhận thấy có một lượng lớn nhu cầu cần quan sát dữ liệu theo thời gian thực, nhóm chúng em đã đề xuất đề tài mô phỏng dữ liệu theo thời gian thực và cung cấp cho người dùng sự gợi ý tới từ model machine learning. Hệ thống được xây dựng với mục đích lưu trữ dữ liệu trên môi trường phân tán HDFS, sử dụng Spark để xử lý dữ liệu và visualize

## Mục tiêu của đề tài

1. Crawl dữ liệu theo từng giây và đẩy lên hệ thống
2. Xây dựng các cụm HDFS để lưu trữ dữ liệu phân tán.
3. Xây dựng được một model machine learning trên spark.
4. Visualize data realtime cùng với predict từ model.
5. Đánh giá khả năng chịu lỗi.

*Đã đạt*

1. Mô phỏng việc streaming data bằng cách đẩy lên dưới dạng streaming, chưa crawl được dữ liệu tại thời gian thực
2. Xây dựng được các cụm HDFS
3. Xây dựng model ở local, không đẩy lên spark được (spark chưa hỗ trợ deep learning)
4. Visualize data realtime.

## Dữ liệu

Nhóm mô phỏng việc streaming data bằng cách crawl dữ liệu về local, sau đó đẩy lên kafka dưới dạng streaming.

Dữ liệu được cung cấp bởi https://eaforexacademy.com/software/forex-historical-data/

Hệ thống sử dụng dữ liệu forex của USD và EUR, GBP và USD

## Kiến trúc hệ thống

Ta sẽ cài đặt hệ thống trên cụm docker container.

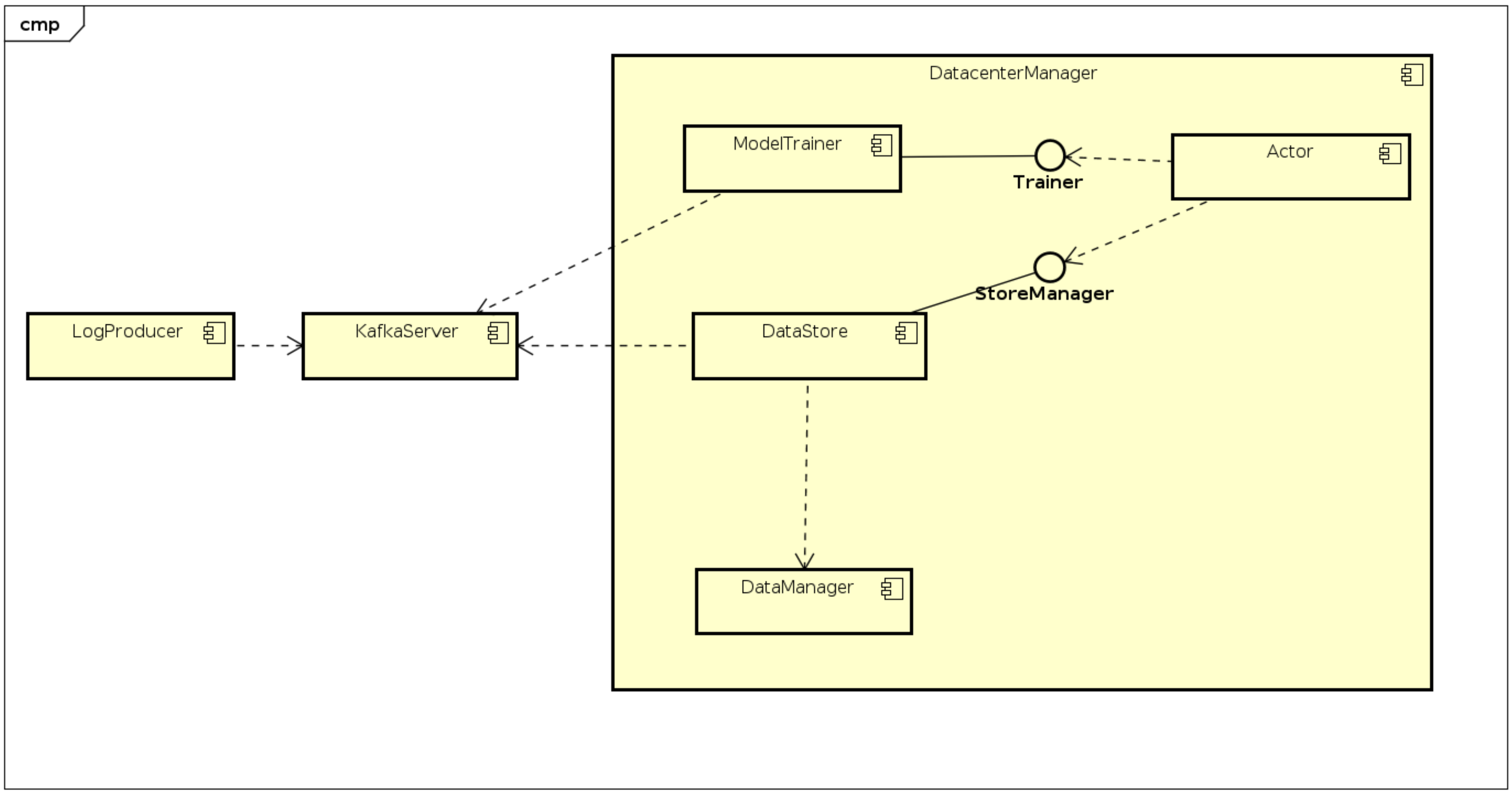
*Thông tin*
kafka bootstrap server: kafka:9092
hadoop default FS: hdfs://hadoop-namenode:8020
spark master: spark://spark-master:7077
spark master gui: localhost:8080

Hệ thông bao gồm các cụm HDFS với 1 namenode và 2 datanode, 2 kafka được quản lý bởi zookeeper, 2 spark worker.

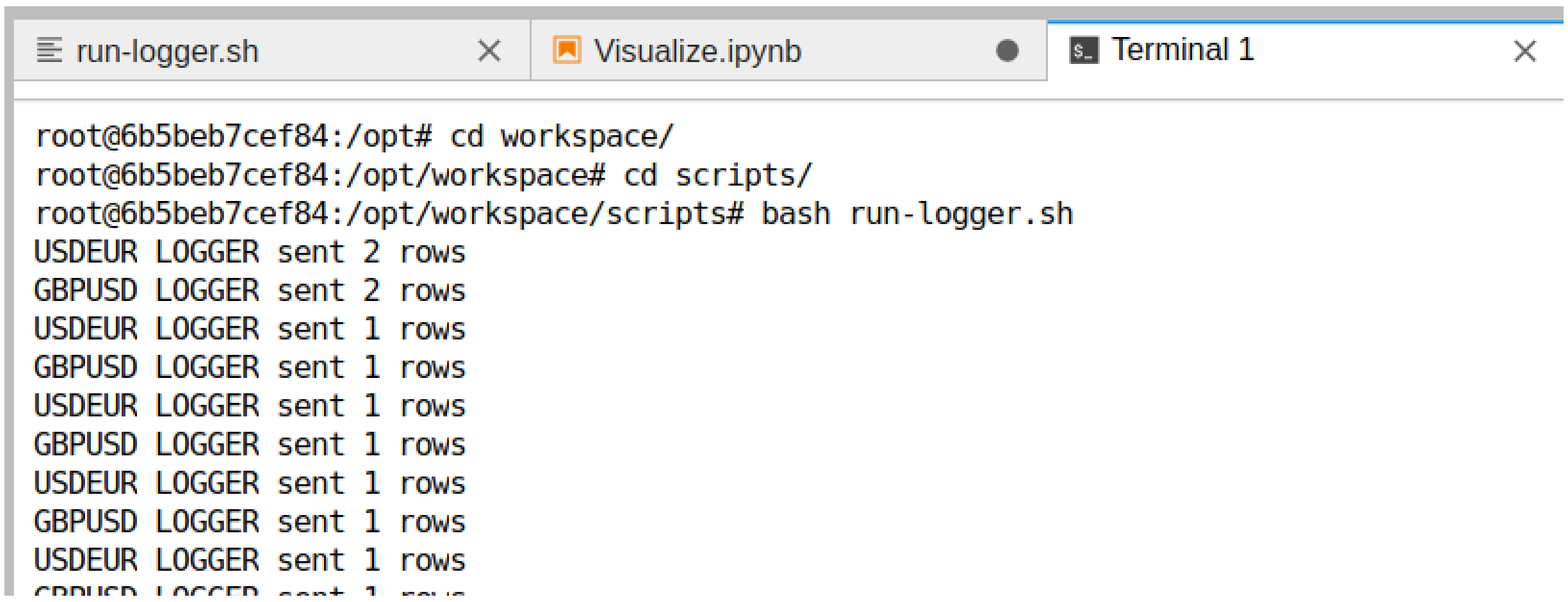
*Về tổng quan, luồng dữ liệu được di chuyển theo thứ tự:*

Dữ liệu được crawl về và gửi lên kafka server theo dạng streaming Bộ xử lý dữ liệu sẽ lấy dự liệu từ server kafka và xử lý theo kiến trúc lambda, từ là sẽ gồm 2 phần chính:

1. DataStoreHDFS: Nhận dữ liệu từ kafka server và lưu lại vào trong hệ thống hadoop
2. Spark: Xử lý real-time dữ liệu bằng notebook, visualize dữ liệu và xây dựng 1 mô hình lstm, predict realtime. Dưới đây là component diagram của hệ thống:

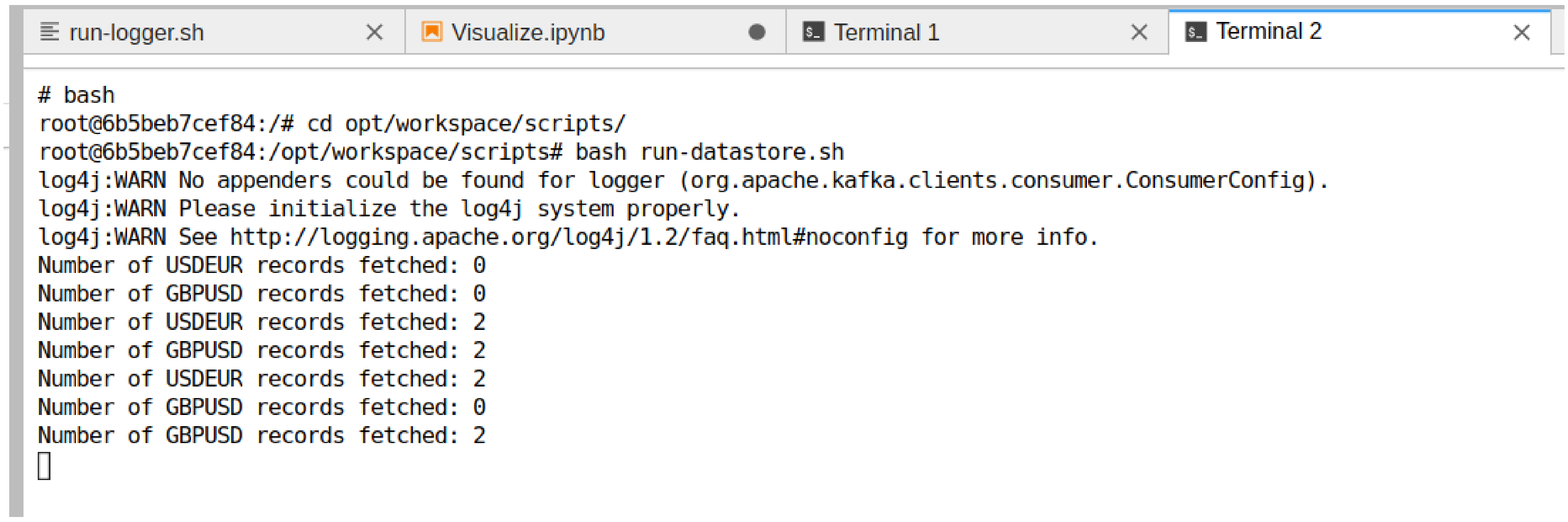


## Quá trình gửi xữ liệu từ local lên kafka



```
root@6b5beb7cef84:/opt# cd workspace/
root@6b5beb7cef84:/opt/workspace# cd scripts/
root@6b5beb7cef84:/opt/workspace/scripts# bash run-logger.sh
USDEUR LOGGER sent 2 rows
GBPUSD LOGGER sent 2 rows
USDEUR LOGGER sent 1 rows
GBPUSD LOGGER sent 1 rows
USDEUR LOGGER sent 1 rows
GBPUSD LOGGER sent 1 rows
USDEUR LOGGER sent 1 rows
GBPUSD LOGGER sent 1 rows
USDEUR LOGGER sent 1 rows
GBPUSD LOGGER sent 1 rows
```

## Quá trình lấy dữ liệu từ kafka lưu vào hdfs



```
# bash
root@6b5beb7cef84:/# cd opt/workspace/scripts/
root@6b5beb7cef84:/opt/workspace/scripts# bash run-datastore.sh
log4j:WARN No appenders could be found for logger (org.apache.kafka.clients.consumer.ConsumerConfig).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
Number of USDEUR records fetched: 0
Number of GBPUSD records fetched: 0
Number of USDEUR records fetched: 2
Number of GBPUSD records fetched: 2
Number of USDEUR records fetched: 2
Number of GBPUSD records fetched: 0
Number of GBPUSD records fetched: 2
▯
```

## Xử lý dữ liệu với Spark, visualize và build model

```
!pip install matplotlib
!pip install tensorflow==1.14
!pip install sklearn
!pip install joblib
!pip install --upgrade scikit-learn==0.20.3
!pip install keras==2.4.3
```

```
Requirement already satisfied: matplotlib in /usr/local/lib/python3.7/site-packages (3.3.3)
Requirement already satisfied: python-dateutil>=2.1 in /usr/local/lib/python3.7/site-packages (from matplotlib) (2.8
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.7/site-packages (from matplotlib) (0.10.0)
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.7/site-packages (from matplotlib) (8.0.1)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.3 in /usr/local/lib/python3.7/site-packages (f
Requirement already satisfied: numpy>=1.15 in /usr/local/lib/python3.7/site-packages (from matplotlib) (1.19.4)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.7/site-packages (from matplotlib) (1.3.1)
Requirement already satisfied: six in /usr/local/lib/python3.7/site-packages (from cycler>=0.10->matplotlib) (1.15.0
Requirement already satisfied: tensorflow==1.14 in /usr/local/lib/python3.7/site-packages (1.14.0)
Requirement already satisfied: termcolor>=1.1.0 in /usr/local/lib/python3.7/site-packages (from tensorflow==1.14) (1
Requirement already satisfied: keras-applications>=1.0.6 in /usr/local/lib/python3.7/site-packages (from tensorflow=
Requirement already satisfied: six>=1.10.0 in /usr/local/lib/python3.7/site-packages (from tensorflow==1.14) (1.15.0
Requirement already satisfied: grpcio>=1.8.6 in /usr/local/lib/python3.7/site-packages (from tensorflow==1.14) (1.34
Requirement already satisfied: numpy<2.0,>=1.14.5 in /usr/local/lib/python3.7/site-packages (from tensorflow==1.14)
Requirement already satisfied: wrapt>=1.11.1 in /usr/local/lib/python3.7/site-packages (from tensorflow==1.14) (1.12
Requirement already satisfied: tensorboard<1.15.0,>=1.14.0 in /usr/local/lib/python3.7/site-packages (from tensorflo
Requirement already satisfied: wheel>=0.26 in /usr/local/lib/python3.7/site-packages (from tensorflow==1.14) (0.36.2
Requirement already satisfied: google-pasta>=0.1.6 in /usr/local/lib/python3.7/site-packages (from tensorflow==1.14)
Requirement already satisfied: astor>=0.6.0 in /usr/local/lib/python3.7/site-packages (from tensorflow==1.14) (0.8.1
Requirement already satisfied: absl-py>=0.7.0 in /usr/local/lib/python3.7/site-packages (from tensorflow==1.14) (0.1
Requirement already satisfied: protobuf>=3.6.1 in /usr/local/lib/python3.7/site-packages (from tensorflow==1.14) (3.
Requirement already satisfied: tensorflow-estimator<1.15.0rc0,>=1.14.0rc0 in /usr/local/lib/python3.7/site-packages
Requirement already satisfied: gast>=0.2.0 in /usr/local/lib/python3.7/site-packages (from tensorflow==1.14) (0.4.0)
Requirement already satisfied: keras-preprocessing>=1.0.5 in /usr/local/lib/python3.7/site-packages (from tensorflow
Requirement already satisfied: h5py in /usr/local/lib/python3.7/site-packages (from keras-applications>=1.0.6->tenso
Requirement already satisfied: werkzeug>=0.11.15 in /usr/local/lib/python3.7/site-packages (from tensorboard<1.15.0,
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.7/site-packages (from tensorboard<1.15.0,>=
Requirement already satisfied: setuptools>=41.0.0 in /usr/local/lib/python3.7/site-packages (from tensorboard<1.15.0
Requirement already satisfied: importlib-metadata in /usr/local/lib/python3.7/site-packages (from markdown>=2.6.8->t
Requirement already satisfied: cached-property in /usr/local/lib/python3.7/site-packages (from h5py->keras-applicati
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.7/site-packages (from importlib-metadata->markdow
Requirement already satisfied: typing-extensions>=3.6.4 in /usr/local/lib/python3.7/site-packages (from importlib-me
Requirement already satisfied: sklearn in /usr/local/lib/python3.7/site-packages (0.0)
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.7/site-packages (from sklearn) (0.20.3)
Requirement already satisfied: scipy>=0.13.3 in /usr/local/lib/python3.7/site-packages (from scikit-learn->sklearn)
Requirement already satisfied: numpy>=1.8.2 in /usr/local/lib/python3.7/site-packages (from scikit-learn->sklearn) (
Requirement already satisfied: joblib in /usr/local/lib/python3.7/site-packages (1.0.0)
Requirement already satisfied: scikit-learn==0.20.3 in /usr/local/lib/python3.7/site-packages (0.20.3)
Requirement already satisfied: scipy>=0.13.3 in /usr/local/lib/python3.7/site-packages (from scikit-learn==0.20.3) (
Requirement already satisfied: numpy>=1.8.2 in /usr/local/lib/python3.7/site-packages (from scikit-learn==0.20.3) (1
Requirement already satisfied: keras==2.4.3 in /usr/local/lib/python3.7/site-packages (2.4.3)
Requirement already satisfied: scipy>=0.14 in /usr/local/lib/python3.7/site-packages (from keras==2.4.3) (1.5.4)
Requirement already satisfied: pyyaml in /usr/local/lib/python3.7/site-packages (from keras==2.4.3) (5.3.1)
Requirement already satisfied: h5py in /usr/local/lib/python3.7/site-packages (from keras==2.4.3) (3.1.0)
Requirement already satisfied: numpy>=1.9.1 in /usr/local/lib/python3.7/site-packages (from keras==2.4.3) (1.19.4)
Requirement already satisfied: cached-property in /usr/local/lib/python3.7/site-packages (from h5py->keras==2.4.3) (
```

```python
from pyspark.sql import SparkSession
```

Chúng ta cần khai báo địa chỉ spark trong môi trường docker: "spark://spark-master:7077" *Sau khi kết nối được với spark thông qua spark context, mở trang localhost:8080 sẽ thấy một job đang chạy*

```python
packages = ["org.apache.spark:spark-streaming-kafka-0-10_2.12:3.0.1",
            "org.apache.spark:spark-sql-kafka-0-10_2.12:3.0.1"]
packages = ",".join(packages)

master = "spark://spark-master:7077"
appName = "Visualize"

spark = SparkSession.builder \
                    .master(master) \
                    .appName(appName) \
                    .config("spark.jars.packages", packages) \
                    .config("spark.executor.memory", "1024m") \
                    .config("spark.executor.core", "1") \
                    .getOrCreate()

spark
```

**SparkSession - in-memory**

**SparkContext**

Spark UI

Version
    v3.0.1
Master
    spark://spark-master:7077
AppName
    Visualize

Chạy chương trình logger trong scripts/run-logger.sh để thực hiện streaming dữ liệu. Dùng spark để nhận dữ liệu được streaming từ kafka tại df

```python
df = spark \
  .readStream \
  .format("kafka") \
  .option("kafka.bootstrap.servers", "kafka:9093, kafka1:9093") \
  .option("subscribe", "USDEUR") \
  .option("startingOffsets", "earliest") \
  .load()
```

```
from pyspark.sql.functions import col, udf, split, mean
from pyspark.sql import Row, DataFrame
import numpy as np
```

Trích xuất các thông tin cần thiết từ dữ liệu và ghi vào bộ nhớ ram ở dạng temporary table

```
value = df.select(col("value").cast("string").alias("Value"))
# value.writeStream.format("memory").queryName("value").outputMode("update").start()

split_col = split(col("Value"), ",")
extract_task = value.select(split_col.getItem(0).cast("int").alias('id'),
                            split_col.getItem(1).cast("string").alias("Time"),
                            split_col.getItem(2).cast("float").alias("Open"),
                            split_col.getItem(3).cast("float").alias("High"),
                            split_col.getItem(4).cast("float").alias("Low"),
                            split_col.getItem(5).cast("float").alias("Close"),
                            split_col.getItem(6).cast("float").alias("Volume"))

query = extract_task.writeStream \
                     .format("memory") \
                     .queryName("total") \
                     .outputMode("update") \
                     .start()
```

Danh sách các tables trong memory

```
spark.sql("show tables").show()
```

```
+--------+---------+-----------+
|database|tableName|isTemporary|
+--------+---------+-----------+
|        |    total|       true|
+--------+---------+-----------+
```

Query quan sát dữ liệu với 20 rows đầu tiên

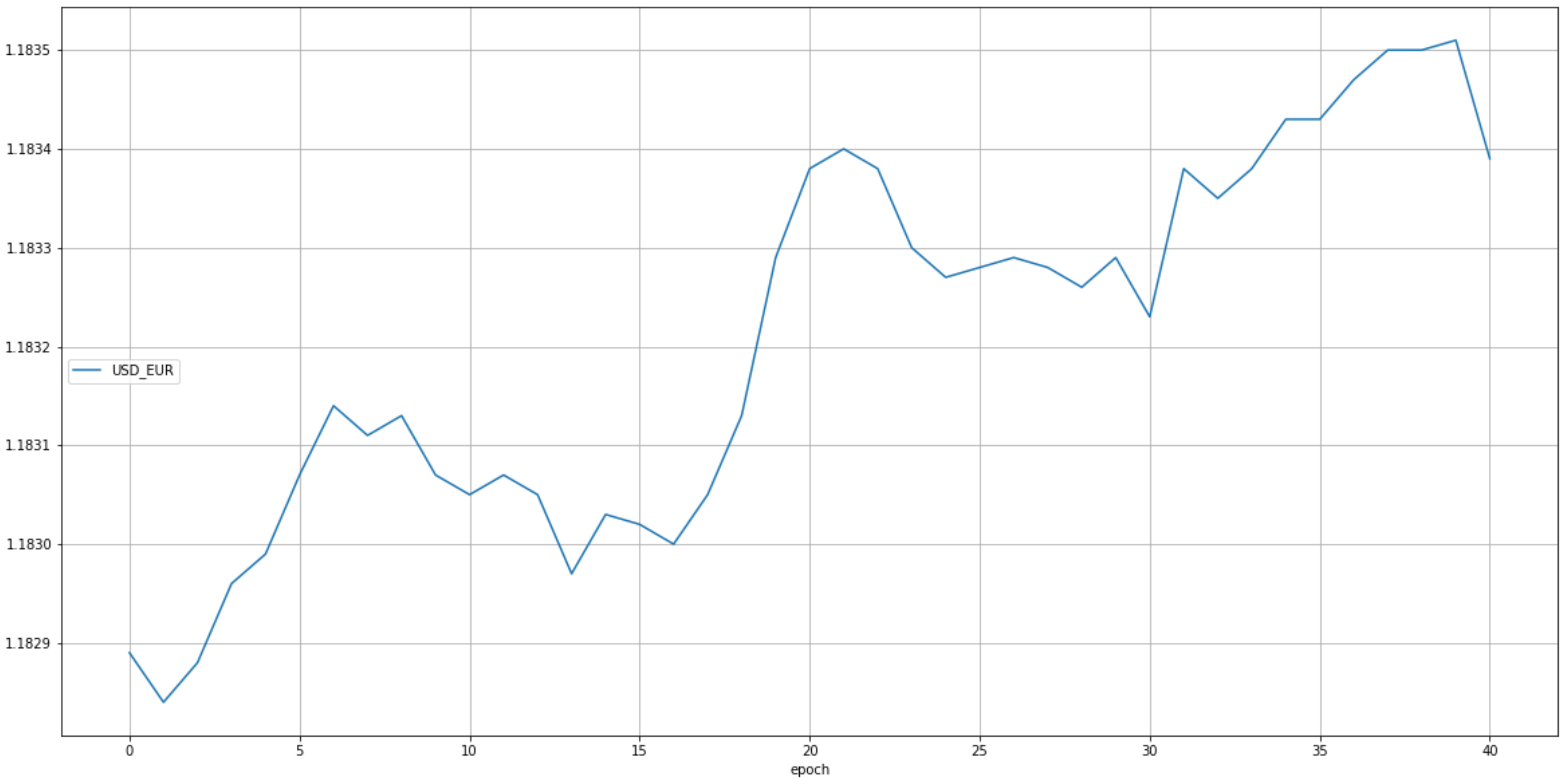
```
data = spark.sql("select * from total")
print(len(np.array(data.collect())))
data.show()
```

```
2261
+---+----------------+-------+-------+-------+-------+------+
| id|            Time|   Open|   High|    Low|  Close|Volume|
+---+----------------+-------+-------+-------+-------+------+
|  0|2020-10-27 04:03| 1.1821|1.18215|1.18208|1.18209| 419.0|
|  1|2020-10-27 04:04|1.18208|1.18214|1.18208|1.18214| 145.0|
|  2|2020-10-27 04:05|1.18214|1.18216|1.18212|1.18214| 176.0|
|  3|2020-10-27 04:07|1.18214|1.18225|1.18214|1.18224| 119.0|
|  4|2020-10-27 04:08|1.18223|1.18225|1.18221|1.18222| 255.0|
|  5|2020-10-27 04:09|1.18223|1.18226|1.18221|1.18224| 219.0|
|  6|2020-10-27 04:10|1.18225|1.18226|1.18224|1.18225| 156.0|
|  7|2020-10-27 04:11|1.18224|1.18225|1.18224|1.18224| 102.0|
|  8|2020-10-27 04:12|1.18225|1.18235|1.18221|1.18234| 641.0|
|  9|2020-10-27 04:13|1.18234|1.18235|1.18226|1.18229| 450.0|
| 10|2020-10-27 04:14|1.18228|1.18231|1.18227|1.18229| 582.0|
| 11|2020-10-27 04:15|1.18231|1.18231|1.18228|1.18229| 306.0|
| 12|2020-10-27 04:16| 1.1823|1.18233|1.18228|1.18233| 379.0|
| 13|2020-10-27 04:17|1.18235| 1.1824|1.18233|1.18239| 275.0|
| 14|2020-10-27 04:18|1.18241|1.18243|1.18239|1.18241| 421.0|
| 15|2020-10-27 04:19|1.18241|1.18244|1.18239|1.18243| 377.0|
| 16|2020-10-27 04:20|1.18244|1.18244|1.18236|1.18236| 328.0|
| 17|2020-10-27 04:21|1.18236|1.18244|1.18233|1.18243| 633.0|
| 18|2020-10-27 04:22|1.18243|1.18243|1.18233|1.18235| 254.0|
| 19|2020-10-27 04:23|1.18236|1.18239|1.18236|1.18236| 321.0|
+---+----------------+-------+-------+-------+-------+------+
only showing top 20 rows
```

Sử dụng matplotlib để visualize dữ liệu dạng streaming, những dữ liệu mới sẽ được tự động visualize lên.

```
from IPython.display import clear_output
from matplotlib import pyplot as plt
import collections
```

```
%matplotlib inline

def live_plot(data_dict, figsize=(20,10), title=''):
    clear_output(wait=True)
    plt.figure(figsize=figsize)
    for label,data in data_dict.items():
        plt.plot(data, label=label)
    plt.title(title)
    plt.grid(True)
    plt.xlabel('epoch')
    plt.legend(loc='center left') # the plot evolves to the right
    plt.show();

data_visua = collections.defaultdict(list)
data_visua['USD_EUR'] = list(np.array(data.select("Close").collect())[2220:])
# data1['fa'] = list(np.array(data.select("Close").collect())[:,0] + 0.01)
while(1):
    live_plot(data_visua)
    if(data_visua['USD_EUR'][-1] != np.array(data.select("Close").collect())[-1][0]) :
        data_visua['USD_EUR'].append(np.array(data.select("Close").collect())[-1][0])
        live_plot(data_visua)
```



## Build model LSTM

Xây dựng một model đơn giản sử dụng LSTM, predict streaming data

```
from sklearn.preprocessing import MinMaxScaler
import joblib
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense,LSTM
```

```
/usr/local/lib/python3.7/site-packages/tensorflow/python/framework/dtypes.py:516: FutureWarning: Passing (type, 1) o
  _np_qint8 = np.dtype([("qint8", np.int8, 1)])
/usr/local/lib/python3.7/site-packages/tensorflow/python/framework/dtypes.py:517: FutureWarning: Passing (type, 1) o
  _np_quint8 = np.dtype([("quint8", np.uint8, 1)])
/usr/local/lib/python3.7/site-packages/tensorflow/python/framework/dtypes.py:518: FutureWarning: Passing (type, 1) o
  _np_qint16 = np.dtype([("qint16", np.int16, 1)])
/usr/local/lib/python3.7/site-packages/tensorflow/python/framework/dtypes.py:519: FutureWarning: Passing (type, 1) o
  _np_quint16 = np.dtype([("quint16", np.uint16, 1)])
/usr/local/lib/python3.7/site-packages/tensorflow/python/framework/dtypes.py:520: FutureWarning: Passing (type, 1) o
  _np_qint32 = np.dtype([("qint32", np.int32, 1)])
/usr/local/lib/python3.7/site-packages/tensorflow/python/framework/dtypes.py:525: FutureWarning: Passing (type, 1) o
  np_resource = np.dtype([("resource", np.ubyte, 1)])
/usr/local/lib/python3.7/site-packages/tensorboard/compat/tensorflow_stub/dtypes.py:541: FutureWarning: Passing (typ
  _np_qint8 = np.dtype([("qint8", np.int8, 1)])
/usr/local/lib/python3.7/site-packages/tensorboard/compat/tensorflow_stub/dtypes.py:542: FutureWarning: Passing (typ
  _np_quint8 = np.dtype([("quint8", np.uint8, 1)])
/usr/local/lib/python3.7/site-packages/tensorboard/compat/tensorflow_stub/dtypes.py:543: FutureWarning: Passing (typ
  _np_qint16 = np.dtype([("qint16", np.int16, 1)])
/usr/local/lib/python3.7/site-packages/tensorboard/compat/tensorflow_stub/dtypes.py:544: FutureWarning: Passing (typ
  _np_quint16 = np.dtype([("quint16", np.uint16, 1)])
```

```
/usr/local/lib/python3.7/site-packages/tensorboard/compat/tensorflow_stub/dtypes.py:545: FutureWarning: Passing (typ
  _np_qint32 = np.dtype([("qint32", np.int32, 1)])
/usr/local/lib/python3.7/site-packages/tensorboard/compat/tensorflow_stub/dtypes.py:550: FutureWarning: Passing (typ
  np_resource = np.dtype([("resource", np.ubyte, 1)])
```

Xử lý dữ liệu trước khi đưa vào model

```
# data = spark.sql("select * from total")
dataset = np.array(data.select("Close").collect())

scaler = MinMaxScaler()
dataset = scaler.fit_transform(dataset)
#Training and test sets
train = dataset[:1500]
test = dataset[1500:]

def get_data(data, look_back):
    datax, datay = [],[]
    for i in range(len(data)-look_back-1):
        datax.append(data[i:(i+look_back),0])
        datay.append(data[i+look_back,0])
    return np.array(datax) , np.array(datay)
look_back = 1

x_train , y_train = get_data(train, look_back)
x_test , y_test = get_data(test,look_back)


#Processing train and test sets for LSTM model
x_train = x_train.reshape(x_train.shape[0],x_train.shape[1], 1)
x_test = x_test.reshape(x_test.shape[0],x_test.shape[1], 1)
print(x_train.shape)
print(y_train.shape)
print(x_test.shape)
print(y_test.shape)
```

```
(1498, 1, 1)
(1498,)
(759, 1, 1)
(759,)
```

Xây dựng model và training

```
#Defining the LSTM model

n_features=x_train.shape[1]
model=Sequential()
model.add(LSTM(100,activation='relu',input_shape=(1,1)))
model.add(Dense(n_features))

model.summary()
model.compile(optimizer='adam', loss = 'mse')
#Compiling
model.compile(optimizer='adam', loss = 'mse')

#Training
model.fit(x_train,y_train, epochs = 5, batch_size=1)
model.save("model.h5")
scaler_filename = "scaler.save"
joblib.dump(scaler, scaler_filename)
```

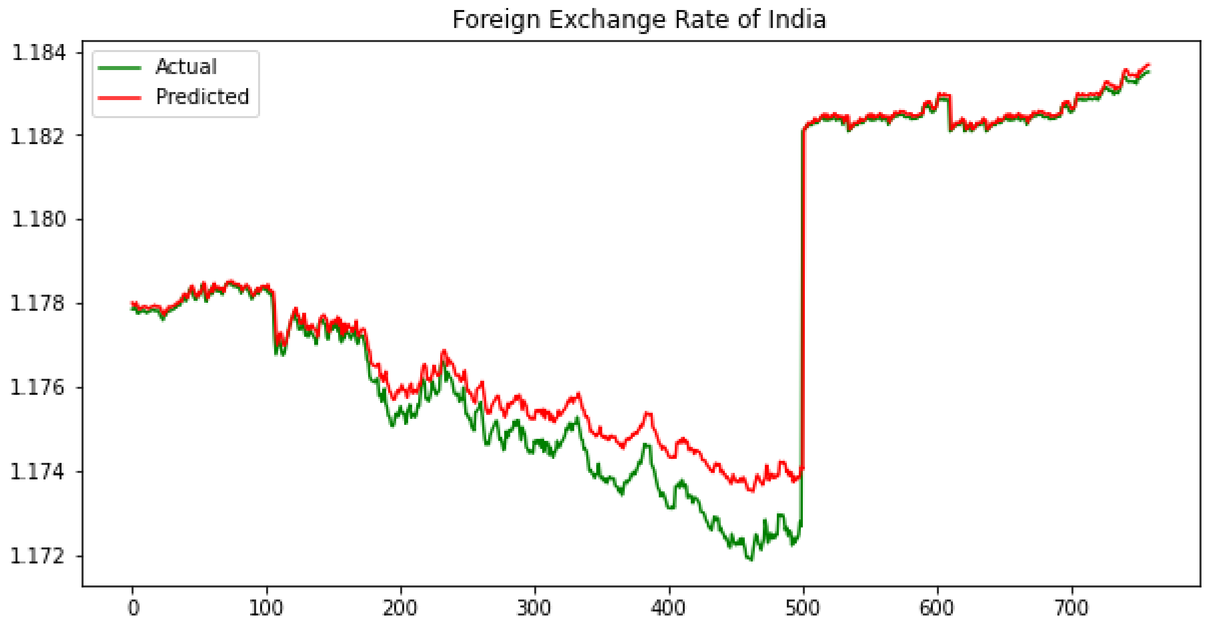
```
WARNING:tensorflow:From /usr/local/lib/python3.7/site-packages/tensorflow/python/ops/init_ops.py:1251: calling Varia
Instructions for updating:
Call initializer instance with the dtype argument instead of passing it to the constructor
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
lstm (LSTM)                  (None, 100)               40800     
_________________________________________________________________
dense (Dense)                (None, 1)                 101       
=================================================================
Total params: 40,901
Trainable params: 40,901
Non-trainable params: 0
_________________________________________________________________
WARNING:tensorflow:From /usr/local/lib/python3.7/site-packages/tensorflow/python/ops/math_grad.py:1250: add_dispatch
Instructions for updating:
Use tf.where in 2.0, which has the same broadcast rule as np.where
Epoch 1/5
1498/1498 [==============================] - 6s 4ms/sample - loss: 0.0307
Epoch 2/5
```

```
1498/1498 [==============================] - 5s 3ms/sample - loss: 2.1172e-04
Epoch 3/5
1498/1498 [==============================] - 4s 3ms/sample - loss: 1.2336e-04
Epoch 4/5
1498/1498 [==============================] - 4s 3ms/sample - loss: 1.2870e-04
Epoch 5/5
1498/1498 [==============================] - 4s 3ms/sample - loss: 1.4191e-04
['scaler.save']
```

Visualize kết quả test của model

```
y_pred = model.predict(x_test)
y_pred = scaler.inverse_transform(y_pred)
y_test = np.array(y_test).reshape(-1,1)
y_test = scaler.inverse_transform(y_test)

#Visualizing the results
plt.figure(figsize=(10,5))
plt.title('Foreign Exchange Rate of India')
plt.plot(y_test , label = 'Actual', color = 'g')
plt.plot(y_pred , label = 'Predicted', color = 'r')
plt.legend()
```

```
<matplotlib.legend.Legend at 0x7ff4903cf2d0>
```



## Visualize data and model predict

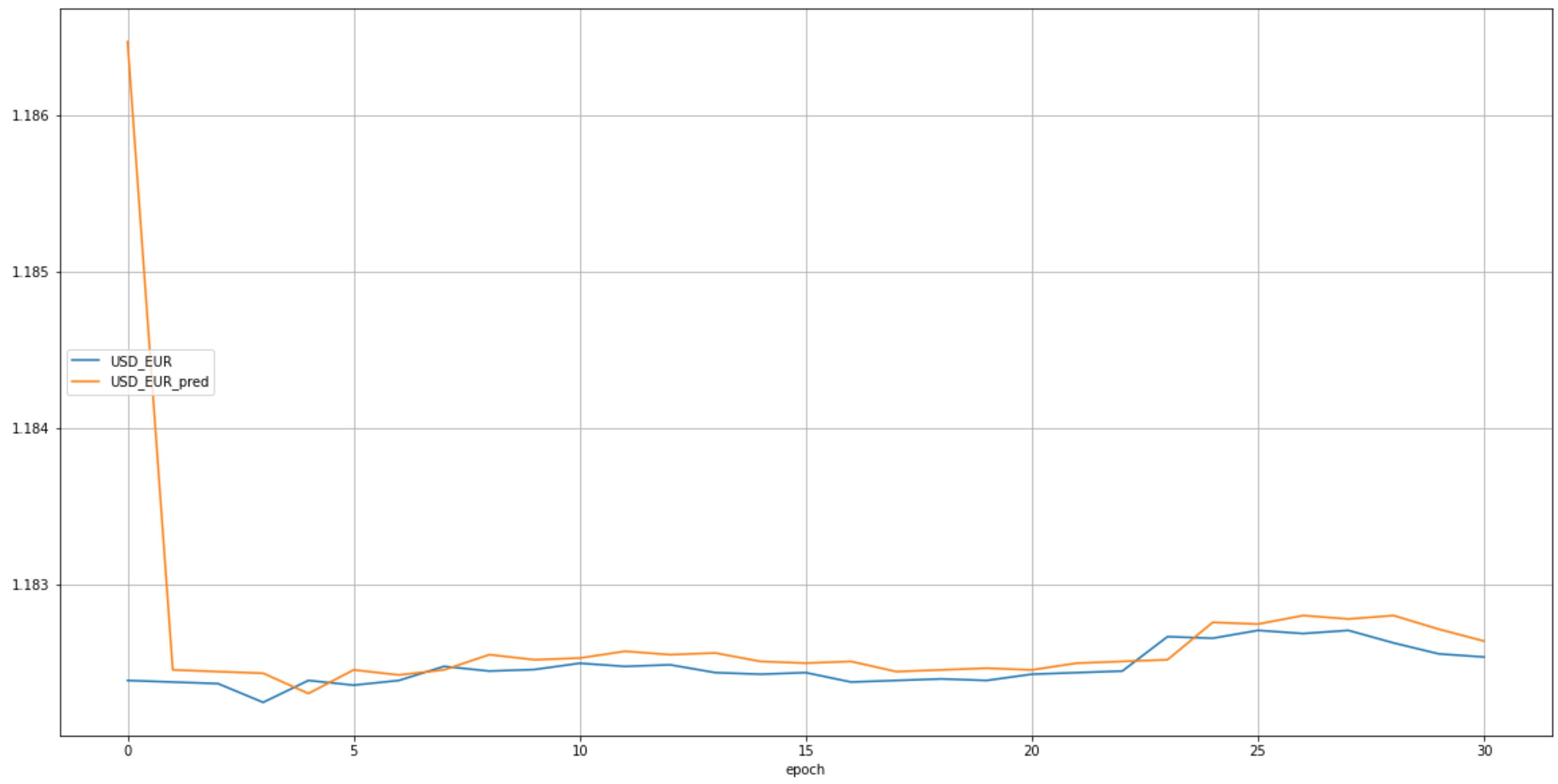
```
data = spark.sql("select * from total")
print(len(np.array(data.collect())))
```

```
2287
```

Data sẽ được đọc theo dạng streaming, mỗi khi data tới, biểu đồ sẽ tự động vẽ ra data và pred tương ứng với nó bằng matplotlib

```
num_start = 2286
data_visua = collections.defaultdict(list)
rt_x_test = np.array(data.select("Close").collect())[num_start:]
scaler.transform(rt_x_test)
rt_x_test = rt_x_test.reshape(rt_x_test.shape[0], rt_x_test.shape[1], 1)
rt_y_test = scaler.inverse_transform(model.predict(rt_x_test))

data_visua['USD_EUR'] = list(np.array(data.select("Close").collect())[num_start+1:, 0])
data_visua['USD_EUR_pred'] = list(rt_y_test[:-1, 0])

# v = 0
while(1):
    live_plot(data_visua)
    if(data_visua['USD_EUR'][-1] != np.array(data.select("Close").collect())[-1, 0]) :
        x = data_visua['USD_EUR'][-1].reshape(1, 1)
        data_visua['USD_EUR'].append(np.array(data.select("Close").collect())[-1, 0])

        x = scaler.transform(x)
        x = x.reshape(1, 1, 1)
        pred = scaler.inverse_transform(model.predict(x))
        data_visua['USD_EUR_pred'].append(pred)
        live_plot(data_visua)
#         v += 1
#         if v == 30:
#             break

```

```
/usr/local/lib/python3.7/site-packages/numpy/core/_asarray.py:136: VisibleDeprecationWarning: Creating an ndarray fr
  return array(a, dtype, copy=False, order=order, subok=True)
```


USD_EUR
USD_EUR_pred
epoch