Vincenzo Cutrona
vincenzo.cutrona@unimib.it

Kafka

INSIDE Lab
Interaction and Semantics for Interoperable Data and sErvices

Department of Informatics, Systems and Communication
University of Milano - Bicocca

Outline

Play with Kafka

- Topics
- Producer/Consumer

Kafka + Python

Kafka Summary

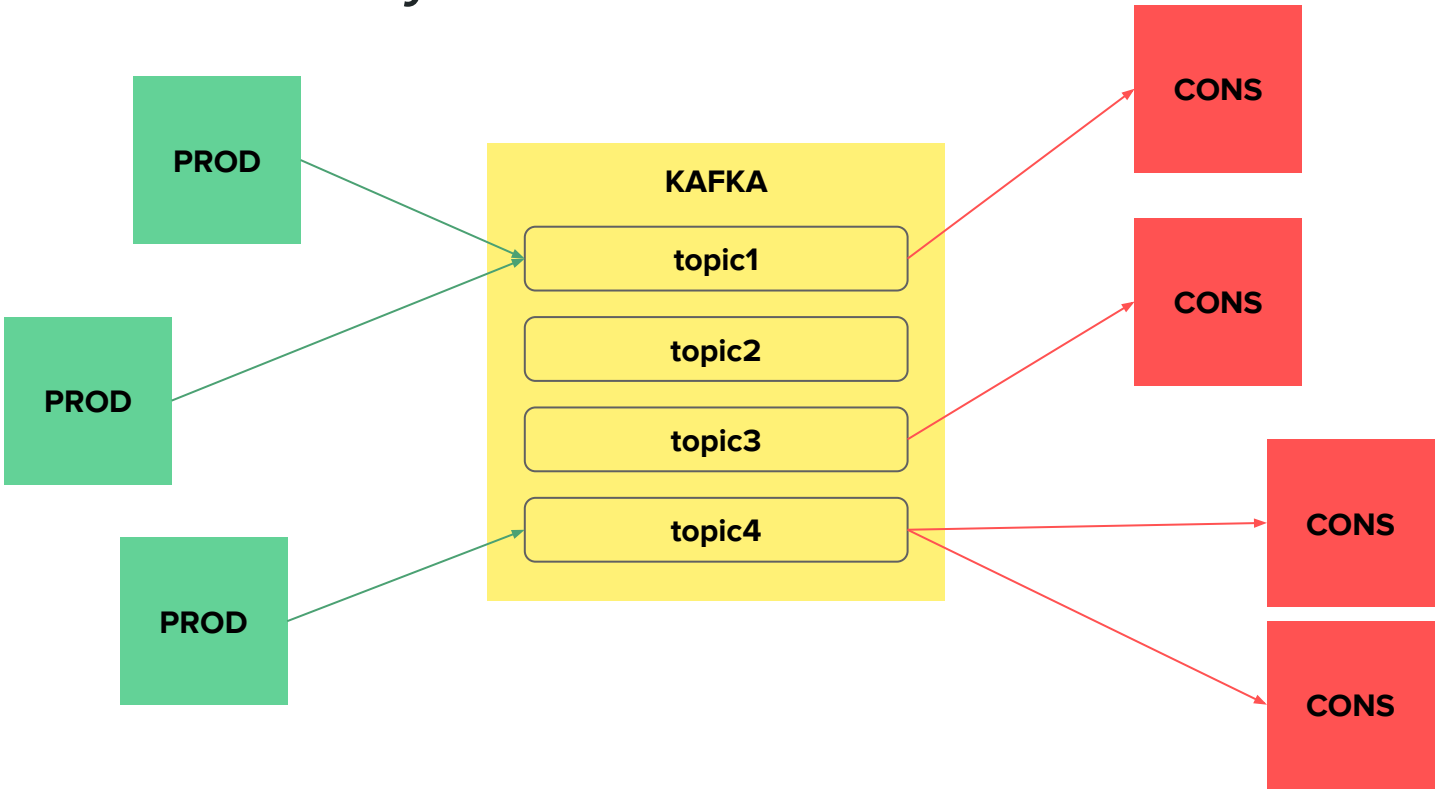
A **messaging systems** that uses a publish/subscribe structure

- **Producers** are application/components that **write messages** to Kafka
- **Consumers** are application/components that **receive messages** from Kafka

Kafka uses **topics:**

- Messages are sent to given **topics** in Kafka
- Consumers **subscribe topics** and they receive messages only from the topics they are subscribed to

Kafka Summary



Play with Kafka

We will make two consoles talk together using Kafka in the middle

We will write messages into the first console and the second one will receive those messages

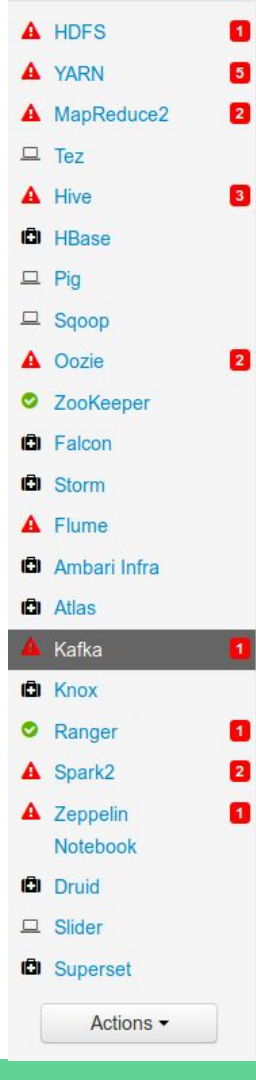
Start Kafka from Ambari

1. Check if you HDP sandbox is running:
   a. Log in to your VM as studente (**ssh studente@*<your_VM_IP>***)
   b. **sudo docker ps**
   c. If the result is the following, go to step 2, otherwise go to step 3

```
studente@hdptest2:~$ sudo docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
studente@hdptest2:~$
```

2. **Start the HDP sandbox**
   a. Run **sudo docker-compose -f /home/studente/HDP/docker-compose.yml up -d**
3. Log in to Ambari (http://*<your_VM_IP>*:1080)
   a. User: maria_dev
   b. Pwd: maria_dev
4. Check if Kafka is already running, otherwise start it manually

Note: Ambari booting will take up to 10 minutes



Create a Kafka Topic

1. Log in to your Sandbox
   a. From Terminal: **ssh maria_dev@<*your_VM_IP*> -p 2222** , or
   b. From Shell in a Box: http://<*your_VM_IP*>:4200
2. Head to **/usr/hdp/current/kafka-broker/bin**
3. Create the topic
   a. **./kafka-topics.sh --create --zookeeper sandbox-hdp.hortonworks.com:2181 --replication-factor 1 --partitions 1 --topic pubsub**
4. Check the list of all topics
   a. **./kafka-topics.sh --list --zookeeper sandbox-hdp.hortonworks.com:2181**

Create the Producer

The Producer accepts messages sent from the console:

1. Open a new console, log in to you Sandbox as maria_dev, head to /usr/hdp/current/kafka-broker/bin
2. Run **./kafka-console-producer.sh --broker-list sandbox-hdp.hortonworks.com:6667 --topic pubsub**
3. Start writing whatever you want

Create the Consumer

The Consumer reads all the console messages:

1. Open a new console, log in to you Sandbox as maria_dev, head to /usr/hdp/current/kafka-broker/bin
2. Run **./kafka-console-consumer.sh --bootstrap-server sandbox-hdp.hortonworks.com:6667 --topic pubsub**
3. Go back to the other console, and write something else

If you get an error about Zookeeper, try this command:

./kafka-console-consumer.sh --bootstrap-server sandbox-hdp.hortonworks.com:6667 --topic pubsub --zookeeper localhost:2181

Demo

Create the Consumer (cont.)

The Consumer reads all the console messages:

4. “Ehy! What about the messages I wrote before starting the consumer?”
    a. Stop the consumer process (CTRL+D)
    b. Run **./kafka-console-consumer.sh --bootstrap-server sandbox-hdp.hortonworks.com:6667 --topic pubsub --from-beginning**
    c. Now the consumer can access the full messages history. Kafka made the job for you.

If you get an error about Zookeeper, try this command:
./kafka-console-consumer.sh --bootstrap-server sandbox-hdp.hortonworks.com:6667 --topic pubsub --from-beginning --zookeeper localhost:2181

Kafka + Python

https://github.com/dpkp/kafka-python

Kafka and Python

Use the **kafka-python** package for Python to create Producers and Consumers

1. Log in to you Sandbox (**ssh maria_dev@<your_VM_IP> -p 2222**)
2. Install pip
    a. **sudo yum -y install python-pip**
3. Install the kafka package
    a. **sudo pip install kafka-python**

Kafka and Python: Create the Producer

Open a Python shell and type the following commands:

```
from kafka import KafkaProducer


producer = KafkaProducer(bootstrap_servers='sandbox-hdp.hortonworks.com:6667')
producer.send('<topic_name>', '<message>')
```

Kafka and Python: Create the Consumer

Open a Python shell and type the following commands:

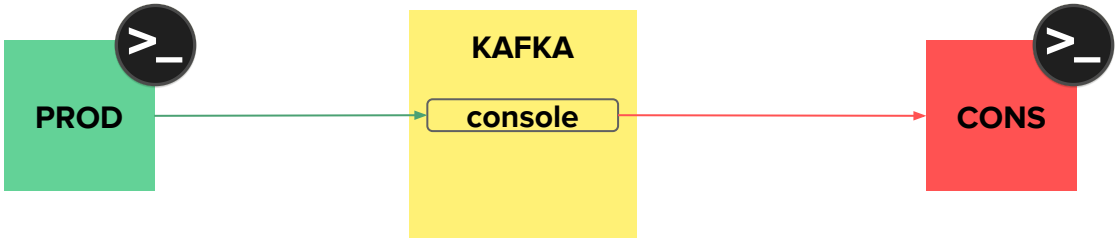
```
from kafka import KafkaConsumer
import time

consumer = KafkaConsumer(bootstrap_servers='sandbox-hdp.hortonworks.com:6667',
                         auto_offset_reset='earliest',
                         consumer_timeout_ms=1000)
consumer.subscribe(['<topic_name>'])
while(True):
    for message in consumer:
        print(message)
    time.sleep(5) # wait 5 secs
```

Demo

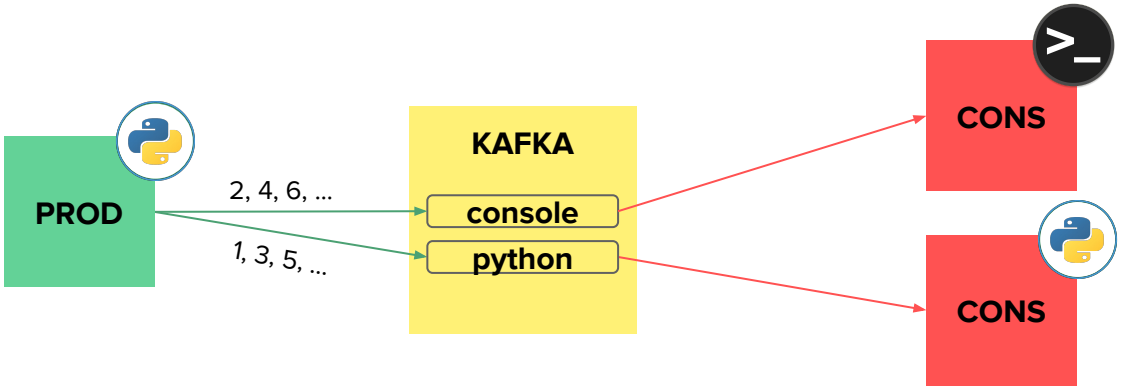
Exercise 1: Try on your own

1) Start Kafka
2) Create a new topic named “console”
3) Launch the producer script (console 1)
4) Launch the consumer script (console 2)
5) Pass messages using two different consoles



Exercise 2: Python+Kafka

1) Create a new topic named “python” (now you should have two topics)
2) Send numbers (range(1,100)) from a Python shell to the two different topics, even numbers should go to the “console” topic, odd numbers to the “python” topic
3) Messages sent to the “console” topic should be printed in a console, while messages sent to the “python” topic should be printed in a second Python shell



Exercise 3: Python+Kafka+Twitter

1) Install tweepy in your Sandbox (sudo pip install tweepy)
2) Get data from Twitter
3) Send tweet text to Kafka (from Python)
4) Read Kafka messages (from Python) and store it somewhere (e.g., to a file)