

UNIVERSIDAD EAFIT
DEPARTAMENTO DE INFORMÁTICA Y SISTEMAS
ST1612 SISTEMAS INTENSIVOS EN DATOS

2021-2

Lab 4 - Amazon MFSK y Apache Kafka

Alcance:

1. gestión de tópicos en MSK desde cli
2. ejecución de al menos 2 producers desde cli (MSK)
3. ejecución de al menos 2 consumers desde cli (MSK)
4. gestión de tópicos en LOCALHOST desde cli
5. ejecución de al menos 2 producers desde cli (LOCALHOST)
6. ejecución de al menos 2 consumers desde cli (LOCALHOST)
7. ejecutar y entender producers y consumers en python hacia MSK y LOCALHOST
8. RETO: crear una API de twitter para leer mensajes, enviarlos vía Kafka (elegir MSK o LOCALHOST) y en el consumidor almacenarlos en una base de datos mongodb.
9. realizar ejemplos de consultas en mongodb de algunos tuits.

Ayuda en el github: https://github.com/st1612eafit/st1612_20212.git (sección kafka)

1. Labs en una máquina standalone descargando Kafka:

Crear una VM linux EC2 o en su PC

Instalar java y Kafka:

```
$ sudo yum install java-1.8.0
$ wget https://archive.apache.org/dist/kafka/2.8.0/kafka_2.12-2.8.0.tgz
$ tar -xzf kafka_2.12-2.8.0.tgz
$ cd kafka_2.12-2.8.0
```

Editar los archivos:

config/zookeeper.properties

```
$ vim
```

Actualizar la variable: (lo puede dejar en el /tmp)

```
dataDir=/tmp/zookeeper
```

a:

```
dataDir=/home/emontoya/bin/kafka_2.12-2.8.0/data/zookeeper
```

```
$ vim config/server.properties
```

Actualizar la variable: o tener en cuenta donde deja los logs en el /tmp

```
log.dirs=/tmp/kafka-logs
```

a:

```
dataDir=/home/emontoya/bin/kafka_2.12-2.8.0/data/kafka-logs
```

Iniciar zookeeper:

```
$ bin/zookeeper-server-start.sh -daemon config/zookeeper.properties
```

Iniciar el servidor de Kafka:

```
$ bin/kafka-server-start.sh -daemon config/server.properties
```

crear un tópico:

```
$ bin/kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1 --topic sample-topic
```

Listar los tópicos:

```
$ bin/kafka-topics.sh --list --zookeeper localhost:2181
```

Borrar un tópico:

```
$ bin/kafka-topics.sh --delete --zookeeper localhost:2181 --topic sample-topic
```

PRODUCERS:

```
$ bin/kafka-console-producer.sh --broker-list localhost:9092 --topic sample-topic
```

config/zookeeper.properties

CONSUMERS:

```
$ bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic sample-topic --from-beginning
```

Detener el servidor de Kafka:

```
$ bin/kafka-server-stop.sh
```

Detener zookeeper:

```
$ bin/zookeeper-server-stop.sh
```

2. Labs en Amazon MSK:

REF: <https://docs.aws.amazon.com/msk/latest/developerguide/getting-started.html>

Crear cluster: st1612-cluster-1

Crear una máquina virtual para gestionar canales, crear producers y consumers.

Create a Topic

ClusterArn

```
arn:aws:kafka:us-east-1:193953349228:cluster/st1612-cluster-1/bc59420e-4bcb-47ed-95f8-7d0576255ea2-8
```

AWSKafkaTutorialTopic

```
z-1.st1612-cluster-1.e0djg5.c8.kafka.us-east-1.amazonaws.com:2181,z-2.st1612-cluster-1.e0djg5.c8.kafka.us-east-1.amazonaws.com:2181,z-3.st1612-cluster-1.e0djg5.c8.kafka.us-east-1.amazonaws.com:2181
```

```
$ aws kafka describe-cluster --region us-east-1 --cluster-arn "ClusterArn"
```

```
$ aws kafka describe-cluster --region us-east-1 --cluster-arn arn:aws:kafka:us-east-1:193953349228:cluster/st1612-cluster-1/bc59420e-4bcb-47ed-95f8-7d0576255ea2-8
```

```
$ bin/kafka-topics.sh --create --zookeeper ZookeeperConnectString --replication-factor 2 --partitions 1 --topic AWSKafkaTutorialTopic
```

crear un tópico:

```
$ bin/kafka-topics.sh --create --zookeeper z-1.st1612-cluster-1.e0djg5.c8.kafka.us-east-1.amazonaws.com:2181,z-2.st1612-cluster-1.e0djg5.c8.kafka.us-east-1.amazonaws.com:2181,z-3.st1612-cluster-1.e0djg5.c8.kafka.us-east-1.amazonaws.com:2181 --replication-factor 2 --partitions 1 --topic AWSKafkaTutorialTopic2
```

config/zookeeper.properties

Listar los tópicos:

```
$ bin/kafka-topics.sh --list --zookeeper z-1.st1612-cluster-1.e0djg5.c8.kafka.us-east-1.amazonaws.com:2181,z-2.st1612-cluster-1.e0djg5.c8.kafka.us-east-1.amazonaws.com:2181,z-3.st1612-cluster-1.e0djg5.c8.kafka.us-east-1.amazonaws.com:2181
```

Borrar un tópico:

```
$ bin/kafka-topics.sh --delete --zookeeper z-1.st1612-cluster-1.e0djg5.c8.kafka.us-east-1.amazonaws.com:2181,z-2.st1612-cluster-1.e0djg5.c8.kafka.us-east-1.amazonaws.com:2181,z-3.st1612-cluster-1.e0djg5.c8.kafka.us-east-1.amazonaws.com:2181 --topic AWSKafkaTutorialTopic2
```

Producers and Consumers

Settings:

```
cp /usr/lib/jvm/JDKFolder/jre/lib/security/cacerts  
/tmp/kafka.client.truststore.jks
```

vim client.properties:

```
security.protocol=SSL  
ssl.truststore.location=/tmp/kafka.client.truststore.jks  
aws kafka get-bootstrap-brokers --region us-east-1 --cluster-arn  
ClusterArn
```

find: BootstrapBrokerStringTls

BootstrapBrokerStringTls

b-1.st1612-cluster-1.e0djg5.c8.kafka.us-east-1.amazonaws.com:9094

```
$ aws kafka get-bootstrap-brokers --region us-east-1 --cluster-arn  
arn:aws:kafka:us-east-1:193953349228:cluster/st1612-cluster-1/bc59420e-4bcb-  
47ed-95f8-7d0576255ea2-8
```

PRODUCERS:

```
$ bin/kafka-console-producer.sh --broker-list BootstrapBrokerStringTls --  
producer.config client.properties --topic AWSKafkaTutorialTopic
```

```
$ bin/kafka-console-producer.sh --broker-list b-1.st1612-cluster-1.e0djg5.c8.kafka.us-east-1.amazonaws.com:9094 --producer.config  
client.properties --topic AWSKafkaTutorialTopic
```

CONSUMERS:

config/zookeeper.properties

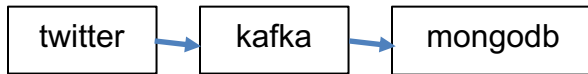
```
$ bin/kafka-console-consumer.sh --bootstrap-server BootstrapBrokerStringTls -  
-consumer.config client.properties --topic AWSKafkaTutorialTopic --from-  
beginning
```

```
$ bin/kafka-console-consumer.sh --bootstrap-server b-1.st1612-cluster-  
1.e0djg5.c8.kafka.us-east-1.amazonaws.com:9094 --consumer.config  
client.properties --topic AWSKafkaTutorialTopic --from-beginning
```

config/zookeeper.properties

RETO: crear una desde la API de twitter lectura de tuits, enviarlos vía Kafka (elegir MSK o LOCALHOST) y en el consumidor almacenarlos en una base de datos mongodb.

Arquitectura propuesta:



realizar ejemplos de consultas en mongodb de algunos tuits.