

9.3 ESCALADO LOGSTASH

Si por ejemplo se va a configurar Kibana para que se conecte al clúster a través de un **nodo coordinador** será necesario instalar Elasticsearch y configurarlo como *client node*.

Instalación y configuración de Zookeeper y Kafka

Funciona sobre Java por lo que su instalación será el primer paso:

```
add-apt-repository ppa:webupd8team/java
apt-get update
apt-get install oracle-java8-installer
nano /etc/environment
    JAVA_HOME="/usr/lib/jvm/java-8-oracle"
source /etc/environment
```

Para la instalación de ambos servicios deberán ejecutarse los siguientes comandos:

```
apt install zookeeperd
adduser --system --no-create-home --disabled-password --disabled-login kafka
cd ~
wget "http://www-eu.apache.org/dist/kafka/1.1.0/kafka_2.12-1.1.0.tgz"
sudo mkdir /opt/kafka
sudo tar -xvzf kafka_2.12-1.1.0.tgz --directory /opt/kafka --strip-components 1
rm -rf kafka_2.12-1.1.0.tgz kafka_2.12-1.1.0.tgz.asc
sudo mkdir /var/lib/kafka
sudo mkdir /var/lib/kafka/data
```

Se deberá modificar el fichero de configuración de Kafka para permitir el borrado de topics y cambiar el *path* donde se almacena los datos `/opt/kafka/config/server.properties`:

```
delete.topic.enable = true
log.dirs=/var/lib/kafka/data
```

Modificamos los permisos de los directorios:

```
sudo chown -R kafka:nogroup /opt/kafka
sudo chown -R kafka:nogroup /var/lib/kafka
```

Para iniciar los servicios:

```
service zookeeperd start
sudo /opt/kafka/bin/kafka-server-start.sh /opt/kafka/config/server.properties
```

- **Prueba de funcionamiento**

Creamos el topic de test:

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

Y se levanta un productor y consumidor para poder escribir por uno y comprobar que el consumidor está leyendo lo escrito:

```
/opt/kafka/bin/kafka-console-producer.sh --broker-list localhost:9092 --topic test
/opt/kafka/bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic test --from-beginning
```

Creación de topics y descarga/configuración del productor

Con los siguientes comandos se podrán crear los topics necesarios y listarlos:

```
/opt/kafka/bin/kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1 --topic sourcetopic
/opt/kafka/bin/kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 1 --partitions 2 --topic udemytopic
/opt/kafka/bin/kafka-topics.sh --list --zookeeper localhost:2181
```

- **Instalación del productor de Netflow**

Clonado del proyecto:

```
apt install git
git clone https://github.com/redBorder/synthetic-producer.git
apt install maven
mvn package
```

Y se cambia en el fichero de configuración el topic en el que se van a meter los datos

`configProducer.yml` :

```
topic: sourcetopic
```

Y se puede arrancar el productor:

```
java -jar /root/synthetic-producer/target/synthetic-producer-1.4.1-SNAPSHOT-selfcontained.jar -c /root/synthetic-producer/configProducer.yml -z localhost:2181 -t 1 -r 1
```

Instalación de Logstash y configuración

Para la instalación de Logstash en las máquinas correspondientes se deben seguir los pasos del módulo de Logstash. Además será necesario añadir en los hosts la resolución de nombre de todas las máquinas involucradas en la práctica `nano /etc/hosts` :

```
192.168.1.30    apacheServer
192.168.1.31    elastic01
192.168.1.32    elastic02
...
```

Y se genera el archivo de configuración de Logstash para los indexers que leerán del topic `udemytopic` e indexará en Elasticsearch:

```
input {
  kafka {
    topics => "udemytopic"
    codec => "json"
    bootstrap_servers => "192.168.1.30:9092"
    client_id => "elastic02"
    group_id => "logstash"
  }
}

output {
  elasticsearch {
    hosts => ["192.168.1.31:9200", "192.168.1.32:9200", "192.168.1.33:9200"]
    index => "netflow-%{+YYYY.MM.dd}"
  }
  stdout { codec => rubydebug }
}
```

Y se podrá ejecutar con el siguiente comando:

```
/usr/share/logstash/bin/logstash -f /etc/logstash/conf.d/kafka.conf --path.setting
s=/etc/logstash
```

En el caso de la máquina Logstash que se encuentra entre los dos topics deberá tener la siguiente configuración:

```
input {
  kafka {
    topics => "sourcetopic"
    codec => "json"
    bootstrap_servers => "192.168.1.30:9092"
    client_id => "apacheServer"
    group_id => "source"
  }
}

output {
  stdout { codec => rubydebug }

  kafka {
    topic_id => "udemytopic"
    codec => "json"
    bootstrap_servers => "192.168.1.30:9092"
    client_id => "apacheServer"
  }
}
```