LAB 4. Administración gráfica.

I. INTRODUCCIÓN.

Existen numerosas herramientas para acceder al entorno de Apache kafka. Una bastante muy interesante es la siguiente:

http://www.kafkatool.com/

Nos permite configurar accesos a nuestro clúster y ver de forma gráfica Topics, configuraciones .

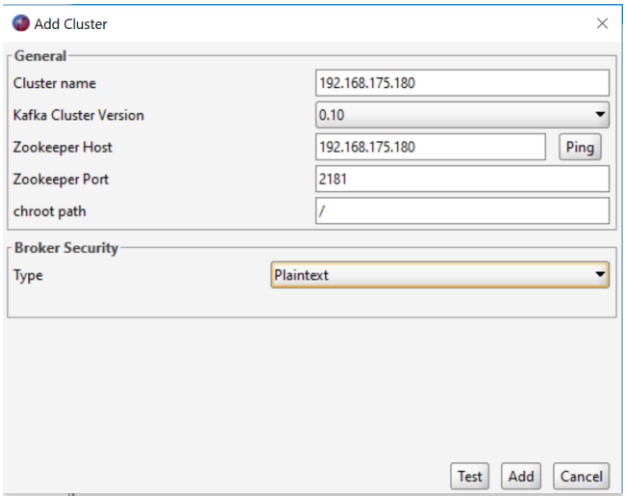


Ilustración 1 Pantalla para añadir clúster a kafkatool.

La aplicación nos permitirá ver numerosa información, e incluso los mensajes que tenemos en las Topic.

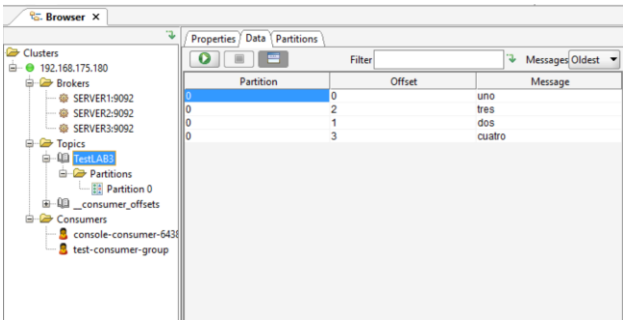


Ilustración 2 Pantalla principal de kafkatool.

Este tipo de herramientas son muy interesantes, pero necesitamos herramientas más centralizadas que estén accesibles por el equipo de administración y soporte. El producto que vamos a instalar es ***kafka-manager***.

II. Kafka manager.

Kafka-manager es un producto de software libre publicado en Github por Yahoo. El proyecto está publicado en:

https://github.com/yahoo/kafka-manager

Para instalarlo deberemos clonarnos el repositorio y compilarlo:

```
kafkauser@SERVER1:~/$ git clone https://github.com/yahoo/kafka-manager.git


kafkauser@SERVER1:~/kafka-manager$ ./sbt clean dist
r ...
[info] Done packaging.
[info]
[info] Your package is ready in /kafka/kafka-manager/target/universal/kafka-manager-1.3.3.7.zip
[info]
[success] Total time: 718 s, completed 07-jun-2017 21:10:48
```

Una vez compilado, deberemos descomprimir el zip que nos ha generado y lo lanzaremos con:

```
$ unzip kafka-manager-1.3.3.7.zip
$ mv kafka-manager-1.3.3.7 kafka-manager

kafka-manager$ ./bin/kafka-manager -Dkafka-manager.zkhosts="192.168.175.180:2181,192.168.175.181:2181,192.168.175.182:2181"
```

Una vez lanzado podremos acceder a la consola vía:

http://$(IP_HOST):9000

Ilustración 3 Consola de inicio de Kafka-manager.

Kafka-manager nos permitirá gestionar diferentes clústeres desde una única consola. En la consola principal sólo podremos añadir clúster o listar. En nuestro caso añadiremos el clúster que montamos en el LAB3.

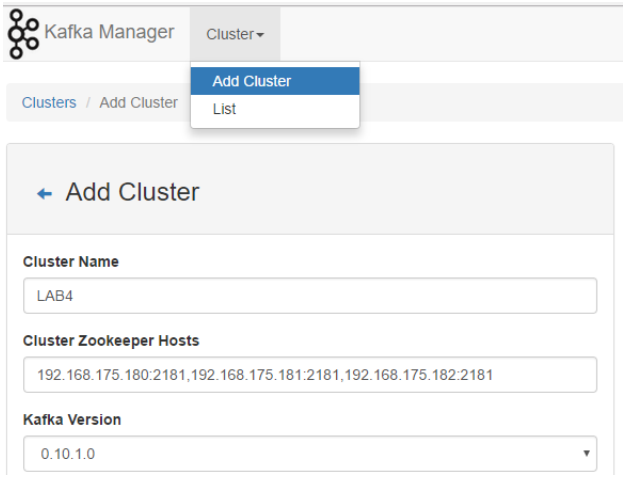


Ilustración 4 Add clúster a Kafka-manager.

Deberemos rellenar las casillas esenciales que son:

- ***Cluster Name***: LAB4
- ***Cluster Zookeeper Hosts***: 192.168.175.180:2181, 192.168.175.181:2181, 192.168.175.182:2181
- ***Kafka Version***: 0.10.1.0

Estas son las esenciales, pero tendremos muchas más que podemos configurar:

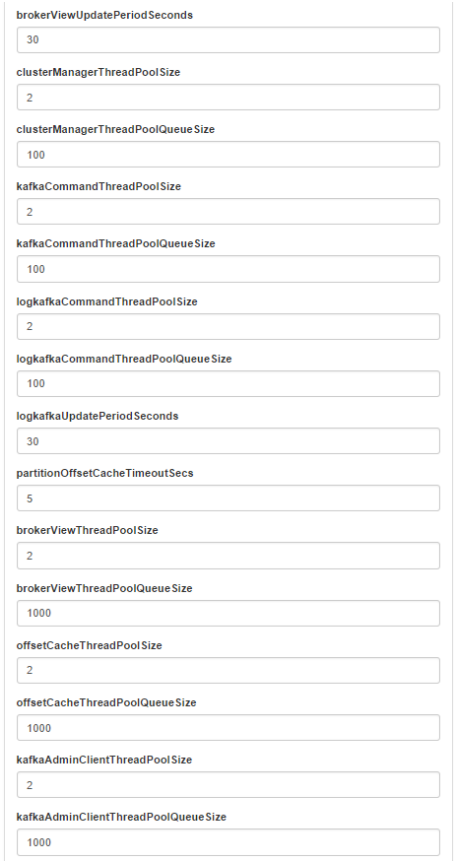


Ilustración 5 Parámetros de configuración para añadir clúster.

Si todo ha ido bien nos aparecerá nuestro nuevo clúster:

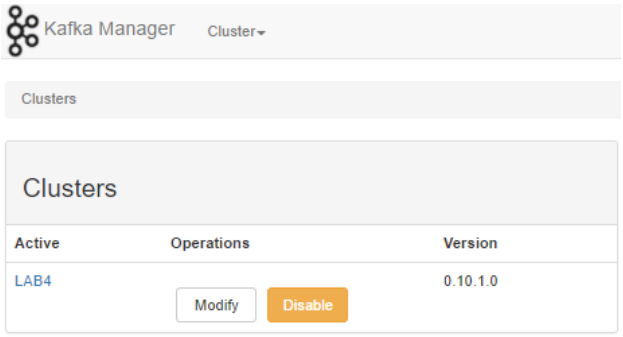
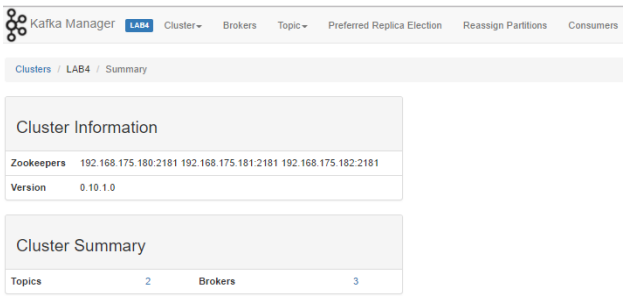


Ilustración 6 Vista de lista de clústers.

Si seleccionamos LAB4 nos aparecerá un nuevo menú con todas las opciones de nuestro clúster:



Si vemos la pestaña de los ***Brokers*** nos mostrara todos los que tenemos en el clúster y una serie de estadísticas. En principio nos aparecen todas a cero porque debemos habilitar el puerto JMX en cada uno de los Brokers.

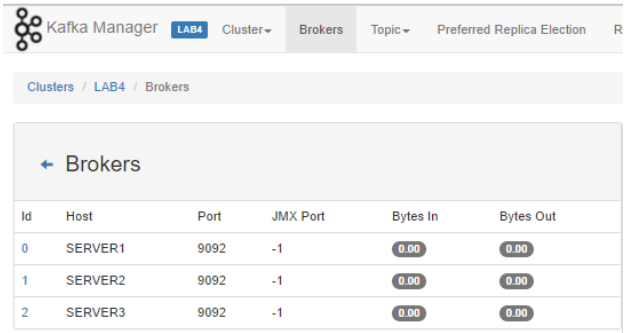


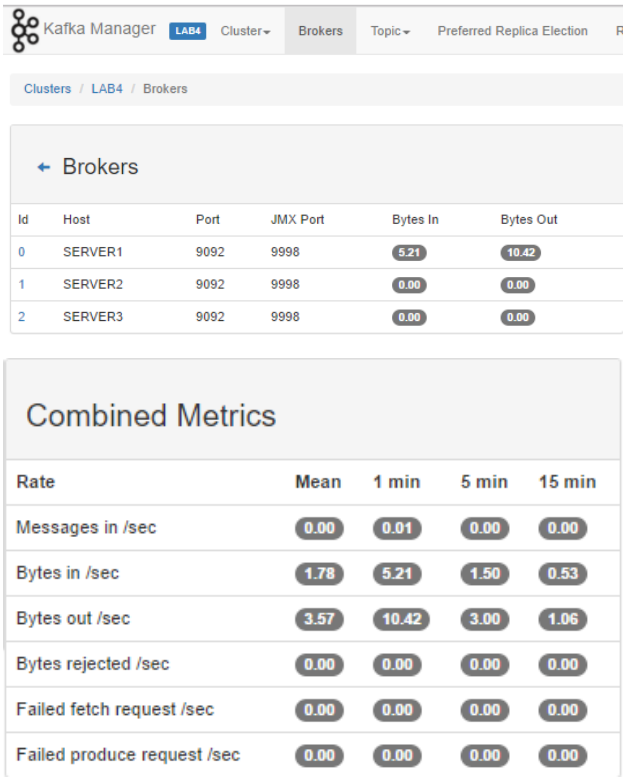
Ilustración 7 Kafka-manager Broker.

Confluent tiene algunos problemas para poder exportar el puerto de forma sencilla. Deberemos añadir las siguientes líneas en el script de arranque: ***kafka-server-start***

```
export JMX_PORT=9998
export KAFKA_JMX_OPTS="-Dcom.sun.management.jmxremote -Dcom.sun.management.jmxremote.authenticate=false -Dcom.sun.management.jmxremote.ssl=false -Djava.rmi.server.hostname=10.191.82.2"
```

Deberemos reiniciar los servidores de Apache Kafka para que cojan los cambios.

Una vez reiniciados empezaremos a ver estadísticas:



Si pinchamos cualquiera de los id de los Brokers nos llevará a la consola extendida del Broker donde veremos información de las Topics que contiene.

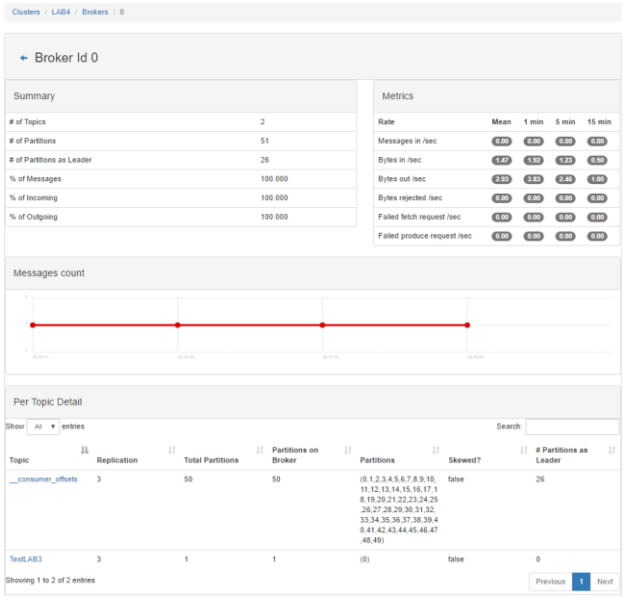


Ilustración 8 Visión extendida del Broker.

La siguiente pestaña nos permitirá ver un listado de las Topic que tenemos creadas.

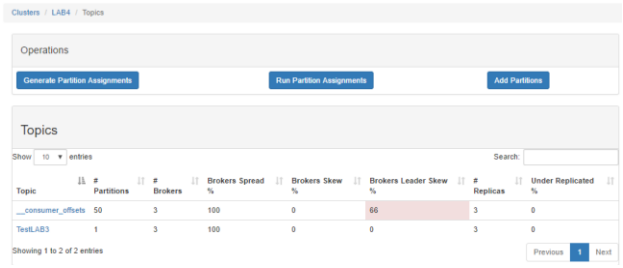
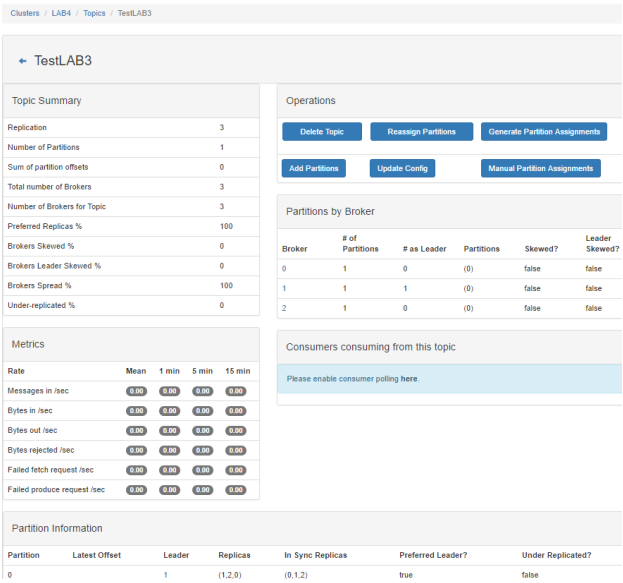


Ilustración 9 Listado de Brokers Kafka-manager.

Desde esta visión podremos realizar las siguientes tareas:

- Generate Partition Assignments.
- Run Partition Assignments.
- Add partitions.

Si pinchamos en la Topic de Prueba TestLAB nos aparecerá la visión ampliado con mucha más información.



Desde esta pantalla podremos gestionar la mayoría de las operatorias referentes a la Topic:

- Delete Topic
- Reassign Partitions.
- Generation Partitions Assignments.
- Add Partitions.
- Update Config.
- Manual Partition Assignments.

En la opción de update podemos modificar todos los parámetros de la cola Topic.

Ilustración 10 Parámetros Topic.

Vemos que todos los contadores están a cero vamos a generar carga para ver cómo evolucionan.

kafkauser@SERVER1:~/confluent$ **./bin/kafka-producer-perf-test --num-records 1000000 --record-size 100 --topic TestLAB3 --throughput 10000 --producer-props bootstrap.servers=192.168.175.180:9092,192.168.175.181:9092,192.168.175.182:9092 max.in.flight.requests.per.connection=1 batch.size=10000**
49962 records sent, 9988,4 records/sec (0,95 MB/sec), 8,7 ms avg latency, 122,0 max latency.
50150 records sent, 10028,0 records/sec (0,96 MB/sec), 5,0 ms avg latency, 48,0 max latency.
50030 records sent, 10006,0 records/sec (0,95 MB/sec), 4,4 ms avg latency, 60,0 max latency.
50010 records sent, 10000,0 records/sec (0,95 MB/sec), 2,5 ms avg latency, 27,0 max latency.
50030 records sent, 10006,0 records/sec (0,95 MB/sec), 2,0 ms avg latency, 39,0 max latency.
50000 records sent, 10000,0 records/sec (0,95 MB/sec), 2,4 ms avg latency, 38,0 max latency.
50060 records sent, 10008,0 records/sec (0,95 MB/sec), 2,4 ms avg latency, 42,0 max latency.
50020 records sent, 10002,0 records/sec (0,95 MB/sec), 1,8 ms avg latency, 29,0 max latency.
50020 records sent, 10002,0 records/sec (0,95 MB/sec), 1,9 ms avg latency, 24,0 max latency.
49990 records sent, 9994,0 records/sec (0,95 MB/sec), 2,1 ms avg latency, 23,0 max latency.
50070 records sent, 10010,0 records/sec (0,95 MB/sec), 2,6 ms avg latency, 42,0 max latency.
50000 records sent, 9996,0 records/sec (0,95 MB/sec), 1,7 ms avg latency, 26,0 max latency.
50050 records sent, 10010,0 records/sec (0,95 MB/sec), 2,0 ms avg latency, 34,0 max latency.
50020 records sent, 9972,1 records/sec (0,95 MB/sec), 1,4 ms avg latency, 21,0 max latency.
50210 records sent, 10038,0 records/sec (0,96 MB/sec), 1,9 ms avg latency, 30,0 max latency.
50010 records sent, 10002,0 records/sec (0,95 MB/sec), 1,9 ms avg latency, 22,0 max latency.
50030 records sent, 10004,0 records/sec (0,95 MB/sec), 7,4 ms avg latency, 164,0 max latency.
50010 records sent, 10002,0 records/sec (0,95 MB/sec), 1,6 ms avg latency, 31,0 max latency.
50010 records sent, 10002,0 records/sec (0,95 MB/sec), 1,9 ms avg latency, 31,0 max latency.
1000000 records sent, 9999,000100 records/sec (0,95 MB/sec), 2,91 ms avg latency, 164,00 ms max latency, 2 ms 50th, 8 ms 95th, 30 ms 99th, 83 ms 99.9th.

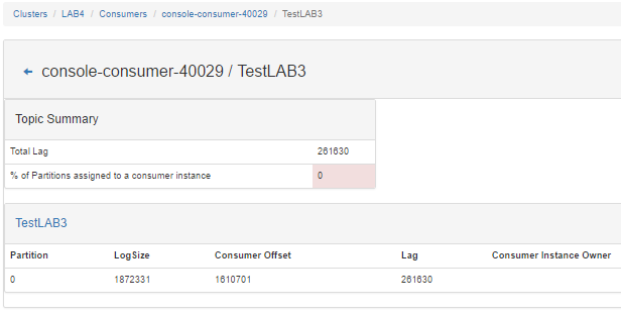
kafkauser@SERVER3:~/confluent$ ./bin/kafka-console-consumer --bootstrap-server 192.168.175.180:9092,192.168.175.181:9092,192.168.175.182:9092 --topic TestLAB3

Con este productor y consumidor funcionando vemos como los contadores comienzan a funcionar correctamente.

Metrics

Rate	Mean	1 min	5 min	15 min
Messages in /sec	34.23	47.33	9.78	3.28
Bytes in /sec	4.4k	6.3k	1.3k	0.4k
Bytes out /sec	9k	12k	2.6k	0.8k
Bytes rejected /sec	0.00	0.00	0.00	0.00
Failed fetch request /sec	0.00	0.00	0.00	0.00
Failed produce request /sec	0.00	0.00	0.00	0.00

Algo interesante que podemos ver es el estado del consumidor y como va consumiendo. En este caso vemos que tiene un LAG bastante grande pero que consume con mucha velocidad. El LAG es la diferencia entre el número total de mensajes que tenemos en la cola y el offset por donde va consumiendo.



Si le dejamos un rato vemos como el LAG se coloca en cero indicando que ha consumido todos los mensajes.