

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 .

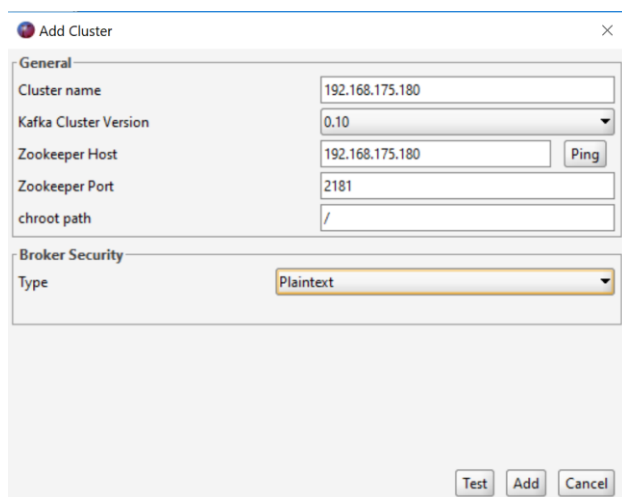


Ilustración 1 Pantalla para añadir clúster a kafkatool.

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

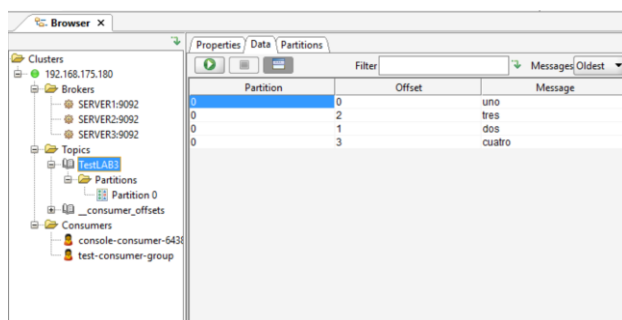


Ilustración 2 Pantalla principal de kafkatool.

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] 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="192.168.175.180:2181,192.168.175.181:218  
1,192.168.175.182:2181"
```

Una vez lanzado podremos acceder a la consola vía:

`http://$(IP_HOST):9000`

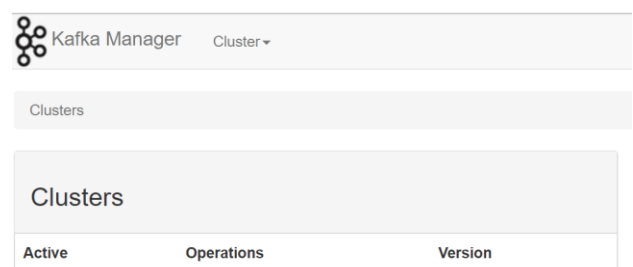
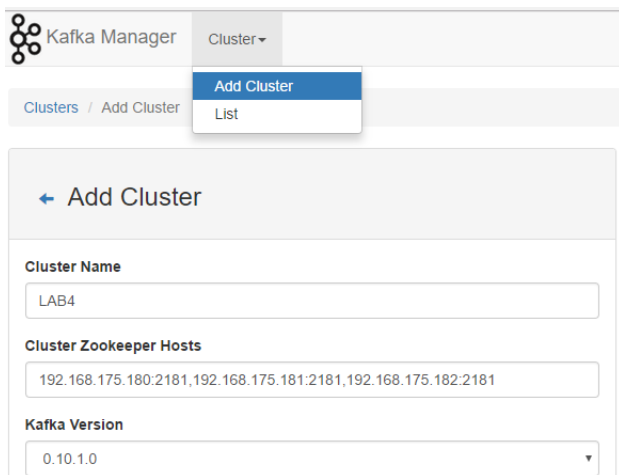


Ilustración 3 Consola de inicio de Kafka-manager.

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.





Kafka Manager Cluster ▾

Clusters / Add Cluster

← Add Cluster

Cluster Name

LAB4

Cluster Zookeeper Hosts

192.168.175.180:2181,192.168.175.181:2181,192.168.175.182:2181

Kafka Version

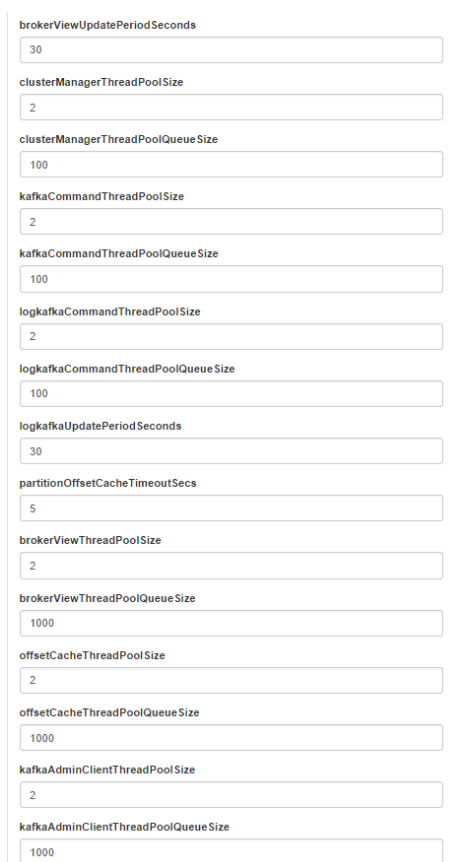
0.10.1.0 ▾

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:



brokerViewUpdatePeriodSeconds

30

clusterManagerThreadPoolSize

2

clusterManagerThreadPoolQueueSize

100

kafkaCommandThreadPoolSize

2

kafkaCommandThreadPoolQueueSize

100

logkafkaCommandThreadPoolSize

2

logkafkaCommandThreadPoolQueueSize

100

logkafkaUpdatePeriodSeconds

30

partitionOffsetCacheTimeoutSecs

5

brokerViewThreadPoolSize

2

brokerViewThreadPoolQueueSize

1000

offsetCacheThreadPoolSize

2

offsetCacheThreadPoolQueueSize

1000

kafkaAdminClientThreadPoolSize

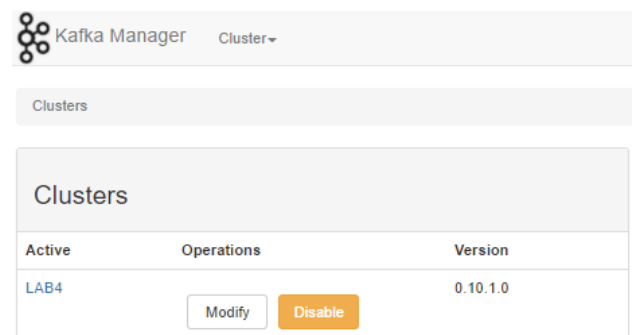
2

kafkaAdminClientThreadPoolQueueSize

1000

Ilustración 5 Parámetros de configuración para añadir clúster.

Si todo ha ido bien nos aparecerá nuestro nuevo clúster:



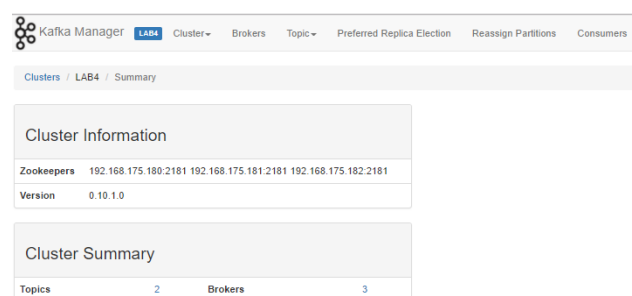
Kafka Manager Cluster ▾

Clusters

Active	Operations	Version
LAB4	Modify Disable	0.10.1.0

Ilustración 6 Vista de lista de clústers.

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



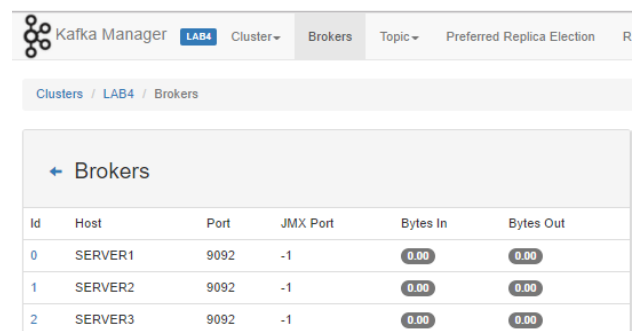
Kafka Manager LAB4 Cluster ▾ Brokers Topic ▾ Preferred Replica Election Reassign Partitions Consumers

Clusters / LAB4 / Summary

Cluster Information	
Zookeepers	192.168.175.180:2181 192.168.175.181:2181 192.168.175.182:2181
Version	0.10.1.0

Cluster Summary	
Topics	2
Brokers	3

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.



Kafka Manager LAB4 Cluster ▾ Brokers Topic ▾ Preferred Replica Election R

Clusters / LAB4 / Brokers

← Brokers

Id	Host	Port	JMX Port	Bytes In	Bytes Out
0	SERVER1	9092	-1	0.00	0.00
1	SERVER2	9092	-1	0.00	0.00
2	SERVER3	9092	-1	0.00	0.00

Ilustración 7 Kafka-manager Broker.

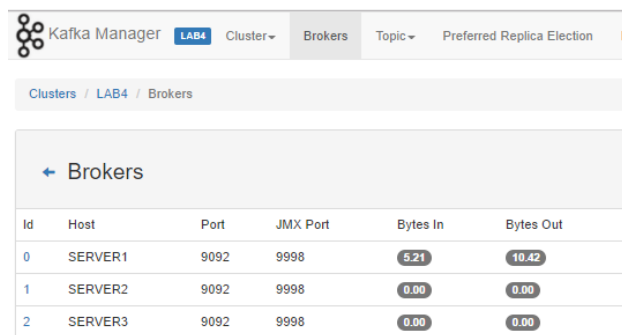
Confluent tiene algunos problemas para poder exportar el puerto de forma sencilla. Debemos añadir las siguientes líneas en el script de arranque: **kafka-server-start**

```
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.

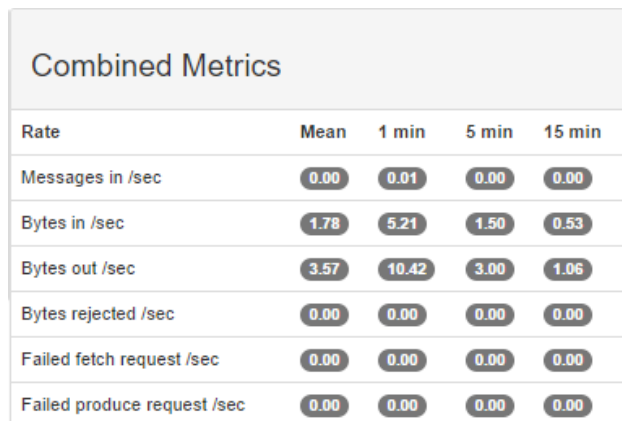


Una vez reiniciados empezaremos a ver estadísticas:



The screenshot shows the 'Brokers' tab in Kafka Manager. It displays a table of brokers with columns: Id, Host, Port, JMX Port, Bytes In, and Bytes Out. Broker 0 (SERVER1) shows 5.21 MB/s in and 10.42 MB/s out, while brokers 1 and 2 show 0.00 MB/s.

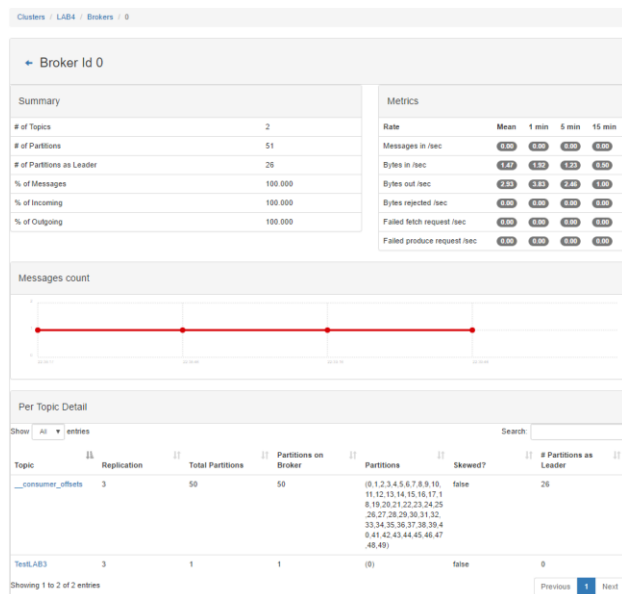
Id	Host	Port	JMX Port	Bytes In	Bytes Out
0	SERVER1	9092	9998	5.21	10.42
1	SERVER2	9092	9998	0.00	0.00
2	SERVER3	9092	9998	0.00	0.00



The screenshot shows the 'Combined Metrics' section. It displays a table with columns: Rate, Mean, 1 min, 5 min, and 15 min. Metrics include Messages in /sec, Bytes in /sec, Bytes out /sec, Bytes rejected /sec, Failed fetch request /sec, and Failed produce request /sec.

Rate	Mean	1 min	5 min	15 min
Messages in /sec	0.00	0.01	0.00	0.00
Bytes in /sec	1.78	5.21	1.50	0.53
Bytes out /sec	3.57	10.42	3.00	1.06
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

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.

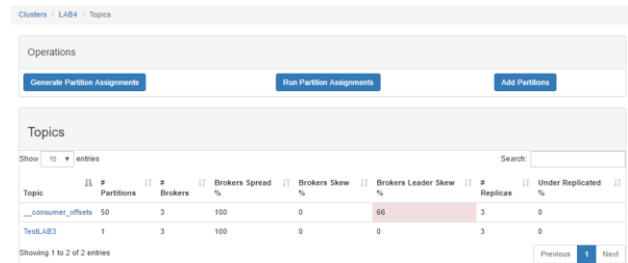


The screenshot shows the extended view for Broker Id 0. It includes a 'Summary' section with statistics like # of Topics, # of Partitions, and % of Messages. It also features a 'Metrics' table similar to the one in the previous screenshot, a 'Messages count' line graph, and a 'Per Topic Detail' table listing topics and their replication status.

Topic	Replication	Total Partitions	Partitions on Broker	Partitions	Skewed?	# Partitions as Leader
__consumer_offsets	3	50	50	(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49)	false	26
TestLAB3	3	1	1	(0)	false	0

Ilustración 8 Visión extendida del Broker.

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



The screenshot shows the 'Topics' tab in Kafka Manager. It displays a table of topics with columns: Topic, # Partitions, # Brokers, Brokers Spread %, Brokers Skew %, Brokers Leader Skew %, # Replicas, and Under Replicated %. Topics listed include __consumer_offsets and TestLAB3.

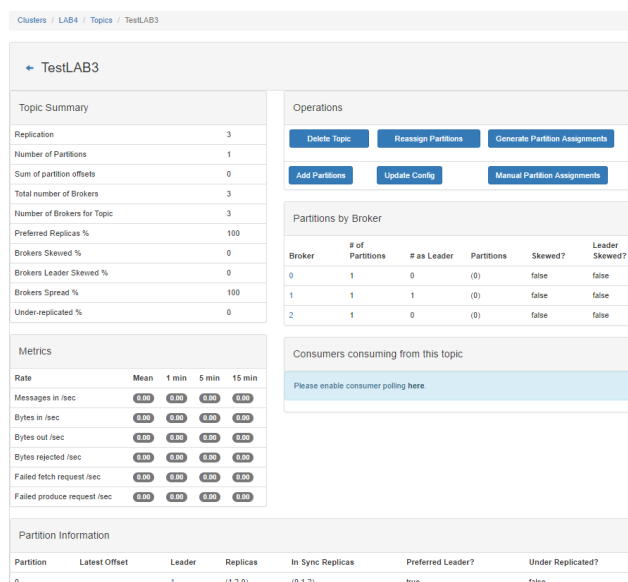
Topic	# Partitions	# Brokers	Brokers Spread %	Brokers Skew %	Brokers Leader Skew %	# Replicas	Under Replicated %
__consumer_offsets	50	3	100	0	66	3	0
TestLAB3	1	3	100	0	0	3	0

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 ampliada con mucha más información.



The screenshot shows the extended view for Topic TestLAB3. It includes a 'Topic Summary' section with statistics like Replication, Number of Partitions, and Sum of partition offsets. It also features an 'Operations' section with buttons for Delete Topic, Reassign Partitions, Generate Partition Assignments, Add Partitions, Update Config, and Manual Partition Assignments. A 'Partitions by Broker' table shows the distribution of partitions across brokers, and a 'Metrics' table shows performance metrics. A 'Partition Information' table lists the details of the single partition in the topic.

Partition	Latest Offset	Leader	Replicas	In Sync Replicas	Preferred Leader?	Under Replicated?
0		1	(1, 2, 0)	(0, 1, 2)	true	false

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.



Update Config

Topic

TestLAB3

max.message.bytes

segment.index.bytes

segment.jitter.ms

min.cleanable.dirty.ratio

retention.bytes

file.delete.delay.ms

compression.type

flush.ms

cleanup.policy

unclean.leader.election.enable

flush.messages

retention.ms

min.insync.replicas

delete.retention.ms

preallocate

index.interval.bytes

segment.bytes

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.

Ilustración 10 Parámetros Topic.



