# Getting started with Kafka

## Install the VM

Running the VM requires 2Gb of RAM and approximately 7Gb of free HD space

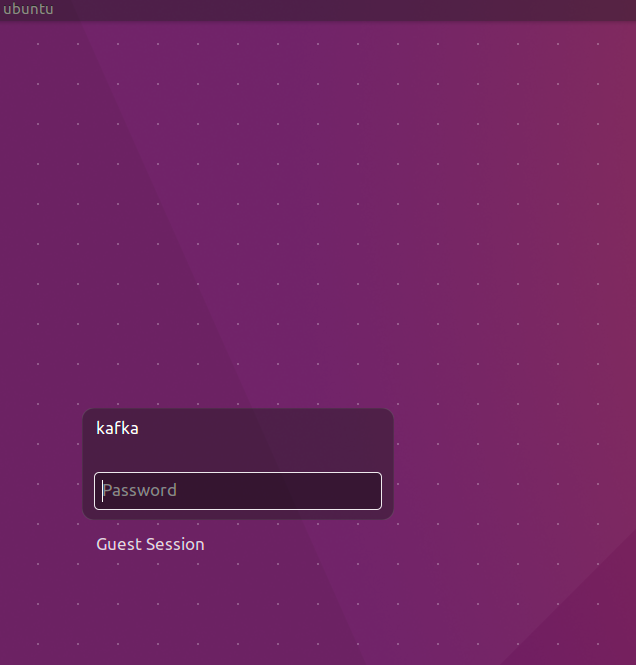
Download VirtualBox from <https://www.virtualbox.org>.

Start VirtualBox

Import the VM image by going to File, Import Virtual Appliance and browse to the image file. The VM requires 2Gb of RAM to run.

Start the VM

The VM will boot to a graphical login screen



The VM has two users: root and kafka. Both have password Welcome01.

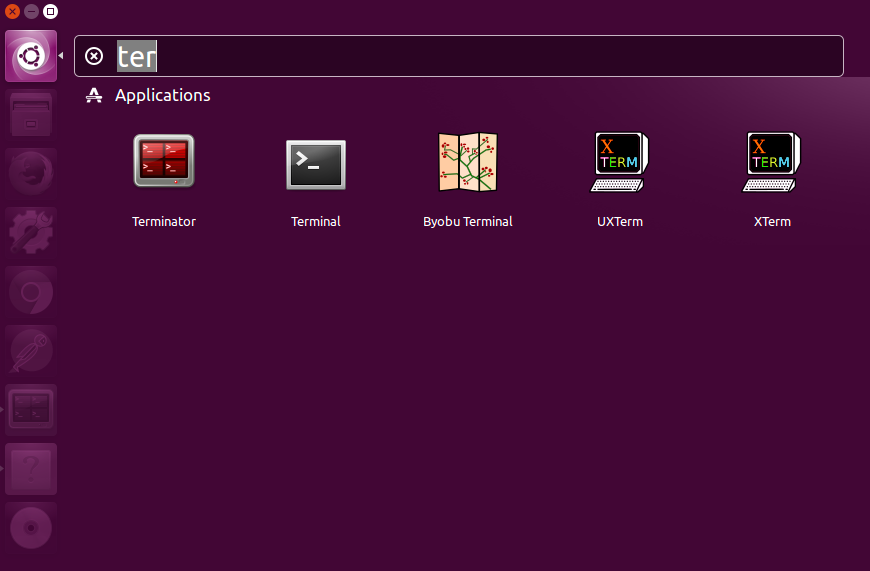
The VM contains:

* Ubuntu 16.04 LTS
* Oracle JDK 8
* Eclipse Neon.2
* Node 7.4
* Zookeeper
* Firefox
* Chrome
* Postman
* confluent-platform-oss-2.11
* kafka-manager 1.3.2.1. When started runs on <http://localhost:9000>
* kafkatool 1.0.1

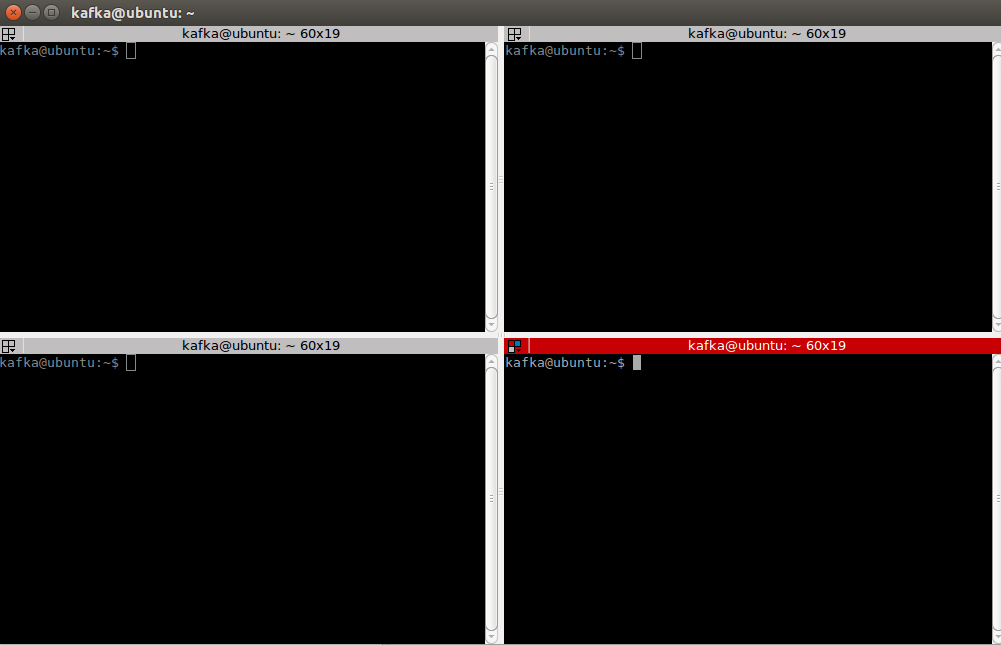
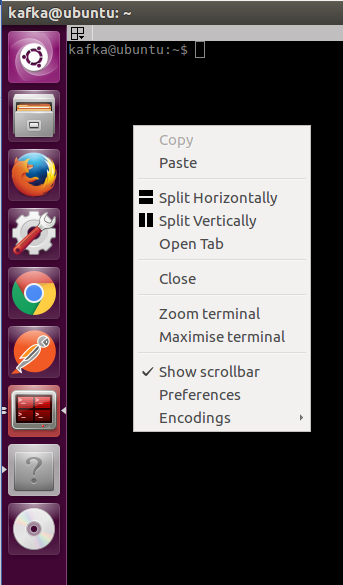
## Getting to know the VM

Login to the VM: kafka/Welcome01

Start Terminator from the Unity menu



Split the screen horizontally and every screen vertically by right clicking in the window



# Getting started with Kafka

In the first terminal window start a Kafka broker:

kafka-server-start /etc/kafka/server.properties

Start in the second terminal window the kafka manager:

kafka-manager

Start in the third terminal window kafkatool:

kafkatool

Minimize the tool. You will use it at a later time.

Continue in the fourth terminal window

### Create a topic

Create a topic:

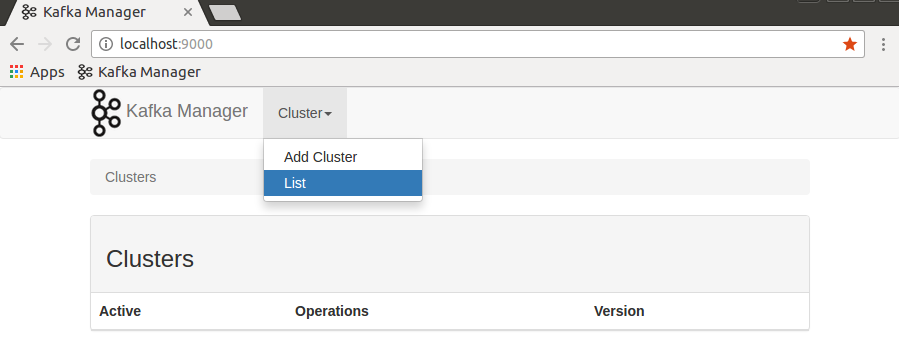
kafka-topics --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1 --topic test

Confirm the topic is created:

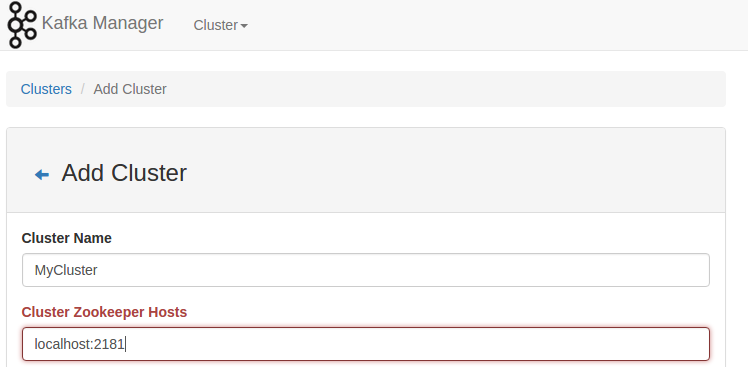
kafka-topics --list --zookeeper localhost:2181

Open a browser and go to <http://localhost:9000> and confirm the topic is created using a webinterface.

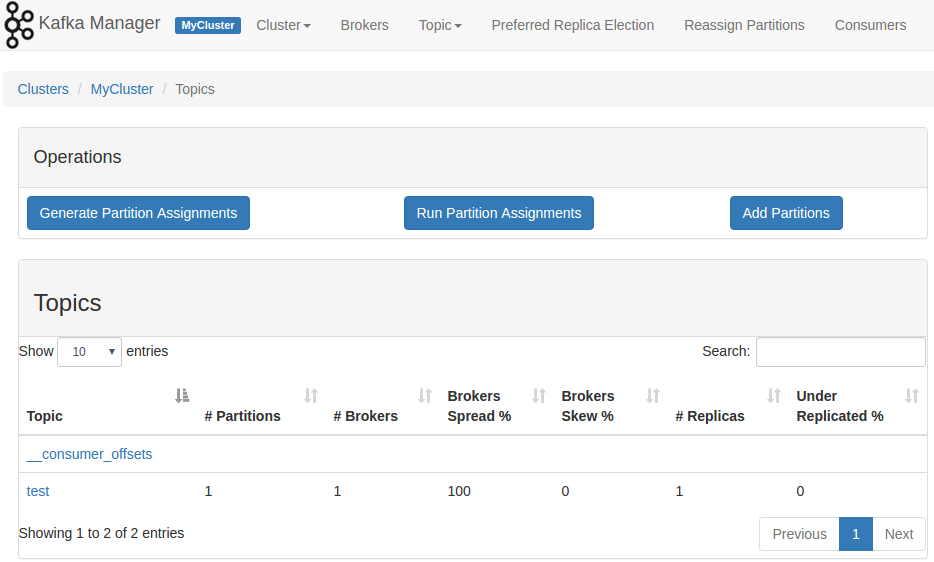
First create a cluster



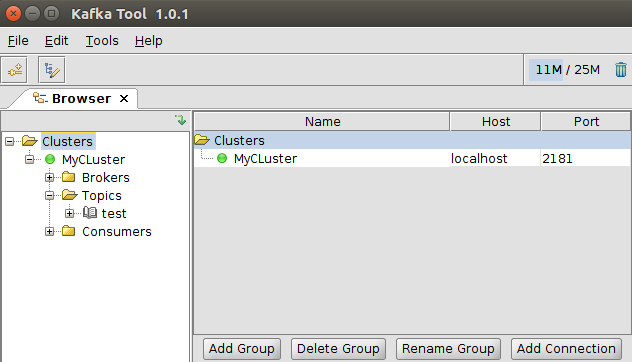
Name the cluster MyCluster and enter localhost:2181 as Zookeeper cluster. Scroll down and click save



Go to the cluster view and determine the topic has been created



Open kafkatool and confirm the topic is created



### Produce a message

Produce a message:

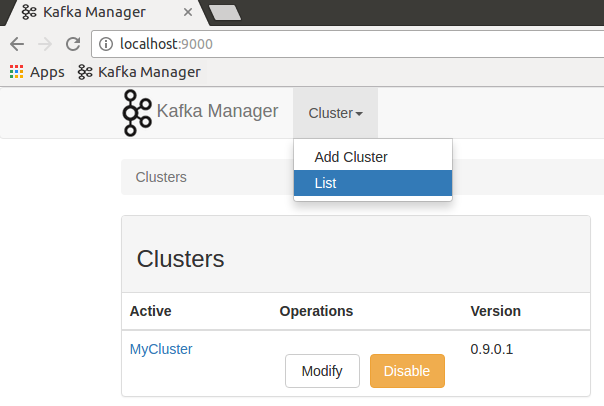
kafka-console-producer --broker-list localhost:9092 --topic test

Type a message and press enter:

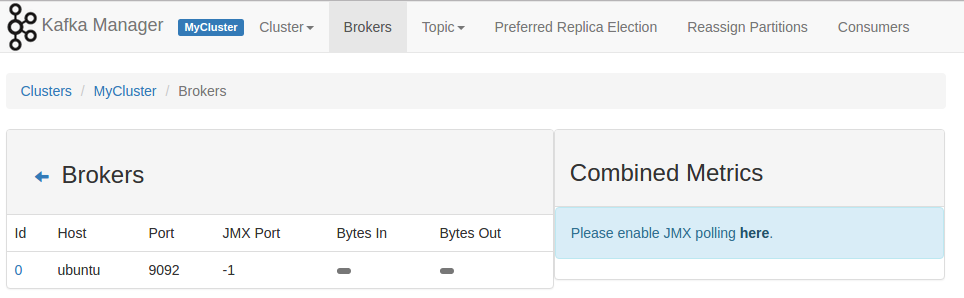
Hello World!

Check the message is created using kafka-manager and kafkatool.

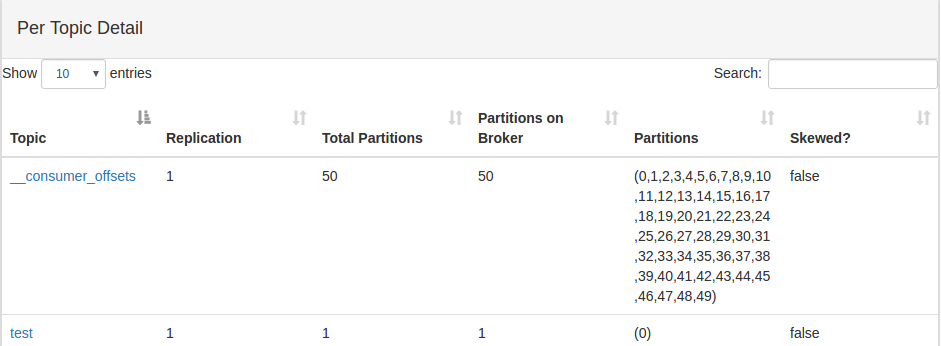
In the kafka-manager, go to Cluster, List and click the cluster you previously created



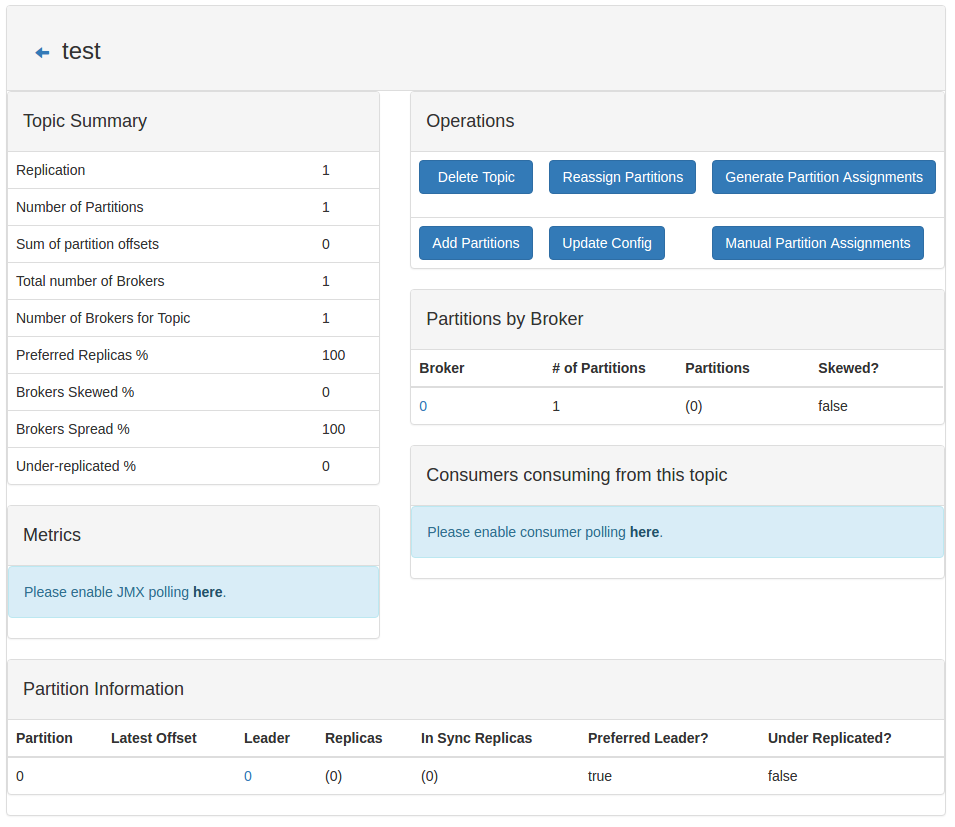
Click brokers, click 0



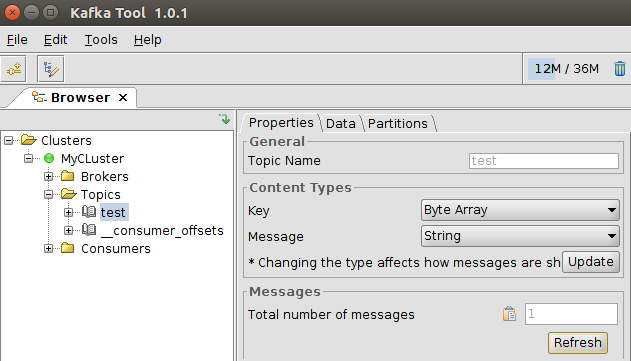
Scroll down and click the test topic



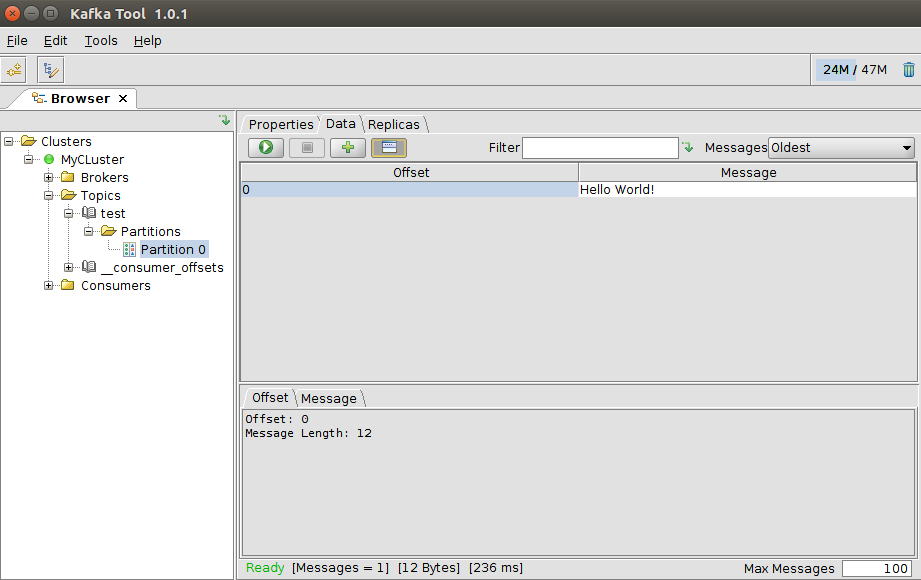
Scroll down again and confirm the latest offset is empty. No consumer has picked up messages yet.



Open kafkatool. Indicate the messages on the topic are string. Click the update button. Click the refresh button to view how many messages are present on the topic. Confirm the number is 1.



Open the partition under the test topic. Click the green play button. Confirm the Hello World! message has arrived.



### Consume a message

Consume a message

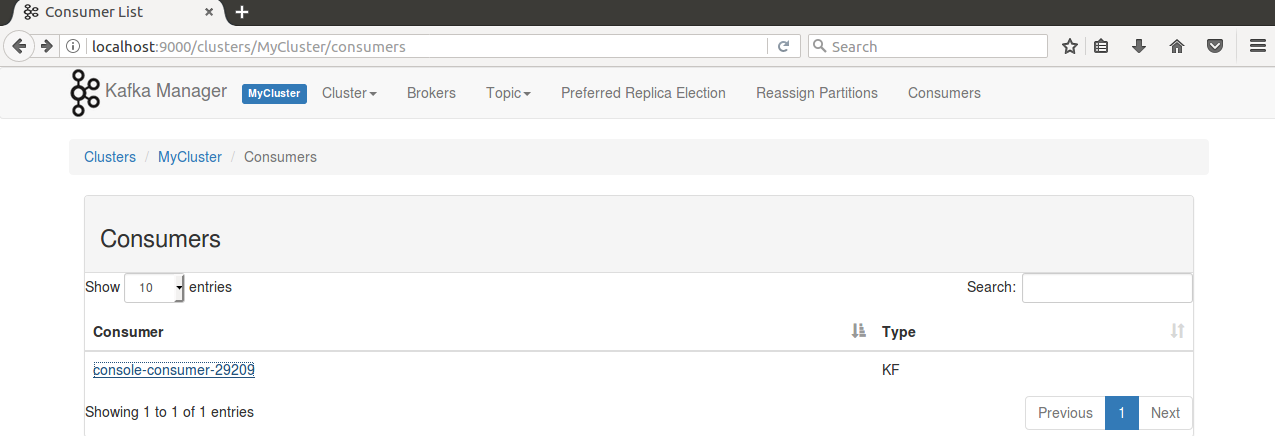
kafka-console-consumer --bootstrap-server localhost:9092 --topic test --from-beginning



Confirm that the previously posted message has been consumed.

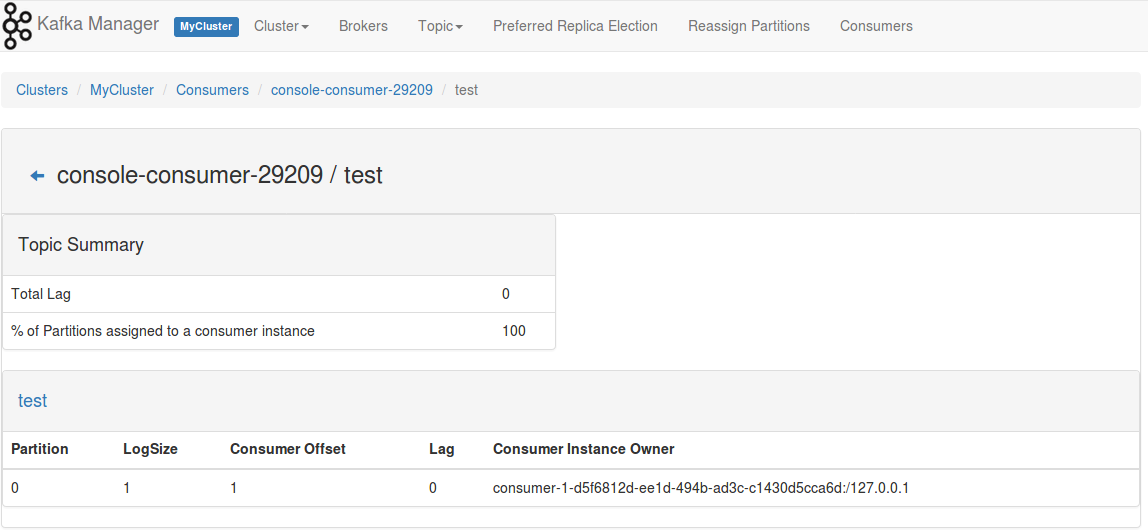
Look at kafka-manager: <http://localhost:9000>

Click on MyCluster, Consumers



Confirm the console consumer is visible

Click the consumer name. Next click the topic test.



Confirm that this consumer has an offset of 1.

End the console consumer by pressing ctrl-C in the terminal window where the consumer is running. End the console producer by pressing ctrl-C in the terminal window where the producer is running.

### Availability

Start a consumer

kafka-console-consumer --consumer-property group.id=TestConsumer --bootstrap-server localhost:9092 --topic test

Start a producer

kafka-console-producer --broker-list localhost:9092 --topic test

Send a message from the producer

Hello World!

Confirm the consumer receives the message.

End the consumer by pressing ctrl-C in the consumer console window

Produce a new message

Hello World!

Start the consumer

Confirm the consumer receives the message it had missed while being down.

### Bonus

Remove the consumer offset registrations.

kafka-consumer-groups --delete --zookeeper localhost:2181 --topic test

Remove the topic

kafka-topics --delete --zookeeper localhost:2181 --topic test

### Questions:

* Why does the consumer not pickup missed messages when the group.id is not specified?
* Where is the offset for the consumers stored? Zookeeper, the consumer, the producer or the broker?

# Oracle Stream Analytics

Start Stream Explorer

cd ~/Oracle/Middleware12212osa/Oracle\_Home/user\_projects/domains/defaultdomain/defaultserver

startwlevs.sh

Start a Kafka broker

kafka-server-start /etc/kafka/server.properties

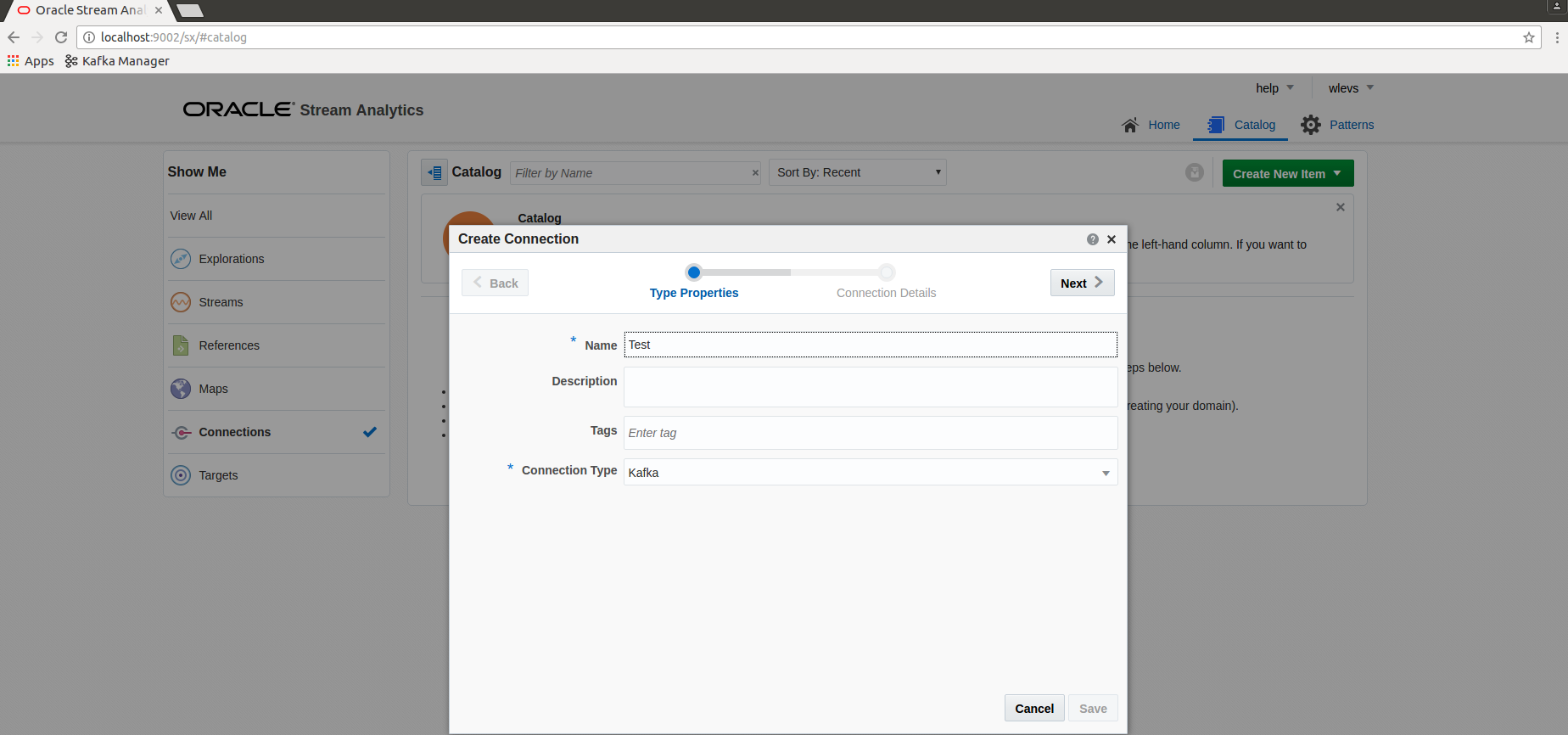
Start a console producer

kafka-console-producer --broker-list localhost:9092 --topic test

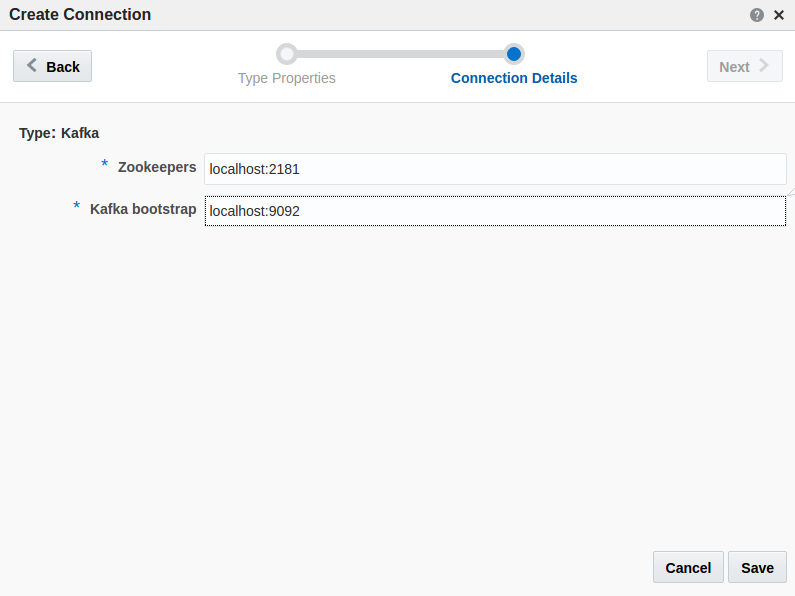
Start a browser and go to <http://localhost:9002/sx>. Login with user wlevs, Welcome01

## Create a connection

Go to Catalog, create New Item, Connection and fill in the fields as shown below



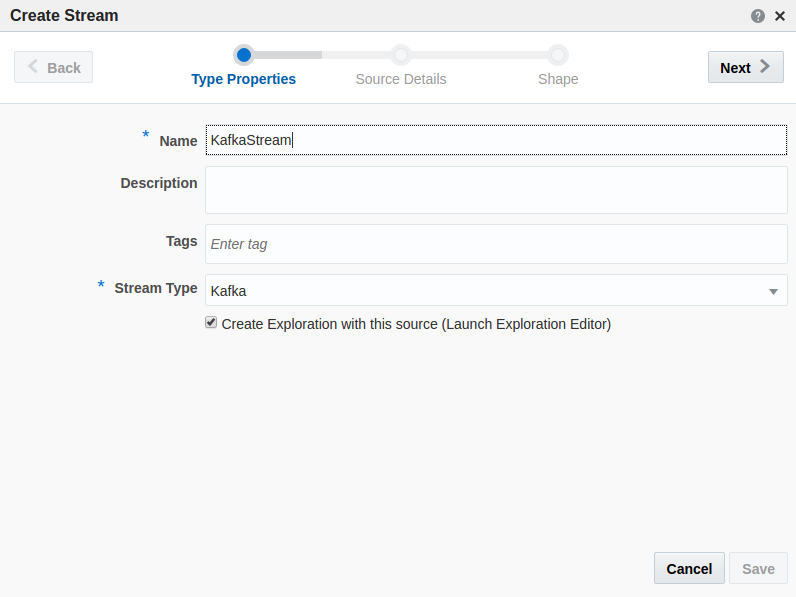
Click next and fill in the Zookeeper URL and bootstrap broker.

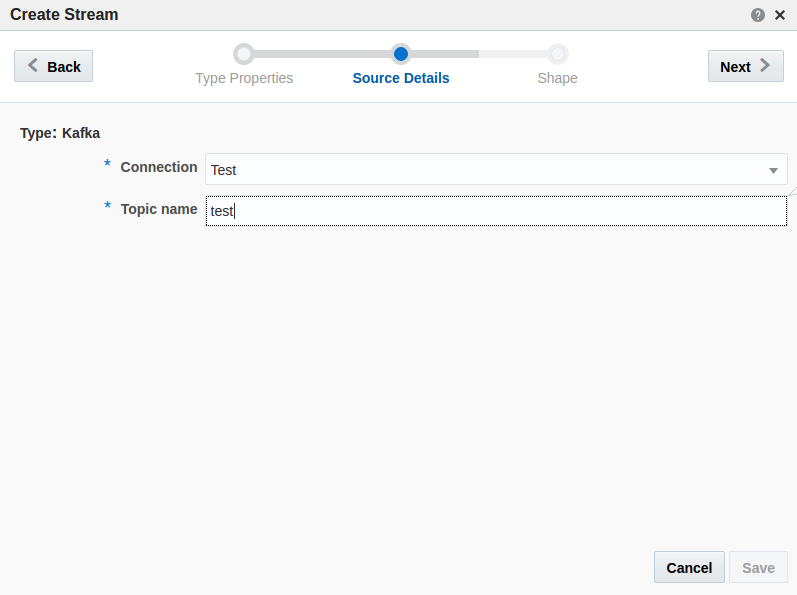


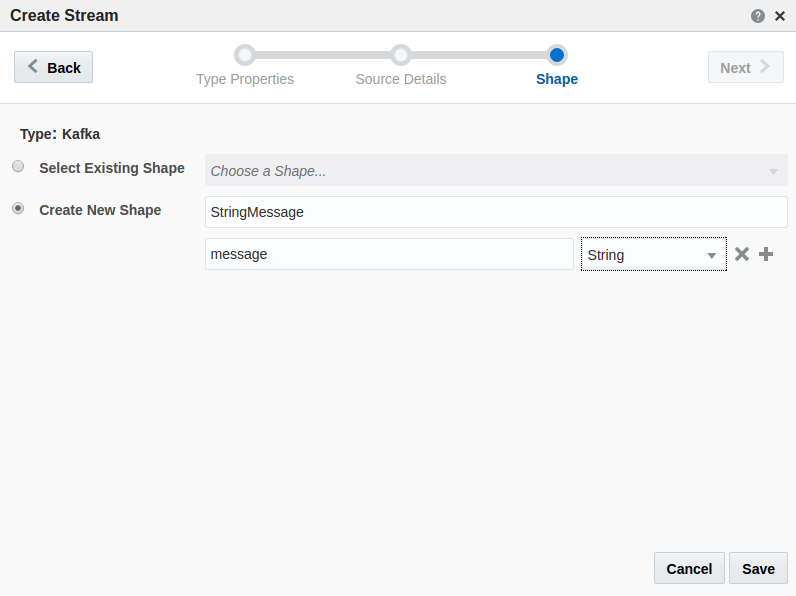
Click Save.

## Create a Stream

Create a stream using the connection.

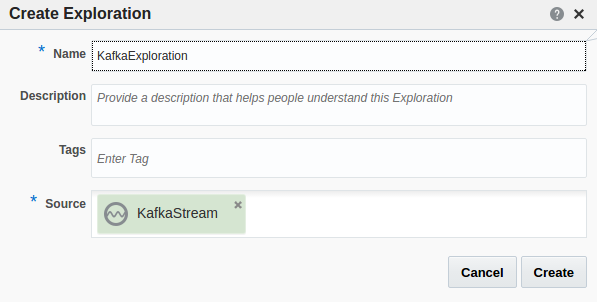






## Create an exploration

Create an exploration based on the stream



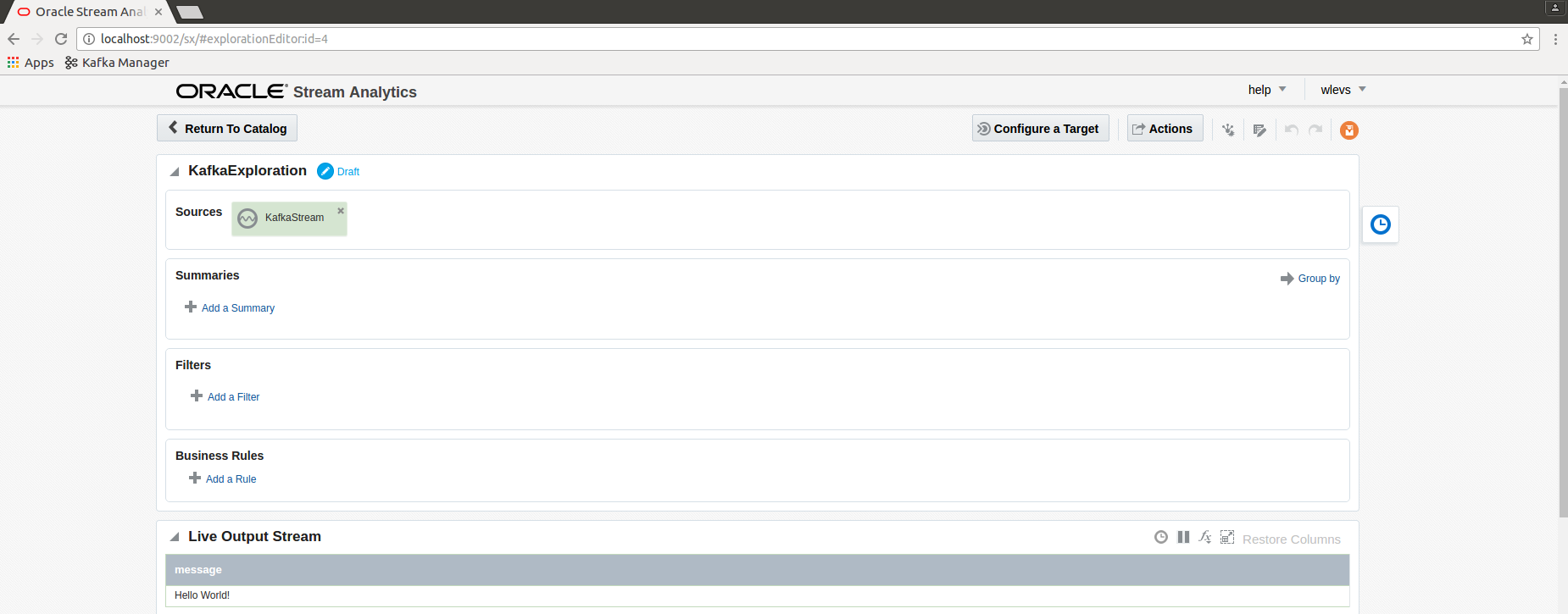
Click the create button

## Confirm OSA receives messages

Next produce a message. Go to the console of the producer and type:

{"message":"Hello World!"}

Confirm you can see the message in your exploration.



# References

Kafka OSA:

https://www.rittmanmead.com/blog/2016/07/stream-analytics-processing-kafka-oracle-stream-analytics/