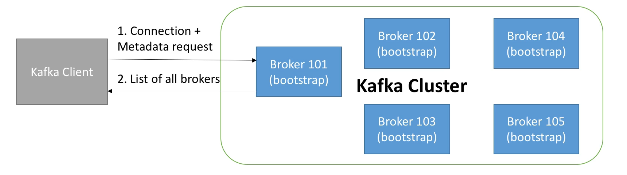
**Kafka Broker Discovery:**

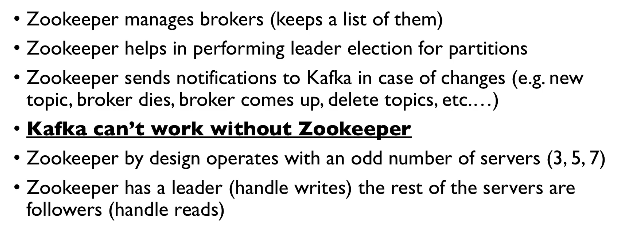
Every kafka broker is also called a “boot strap server”.

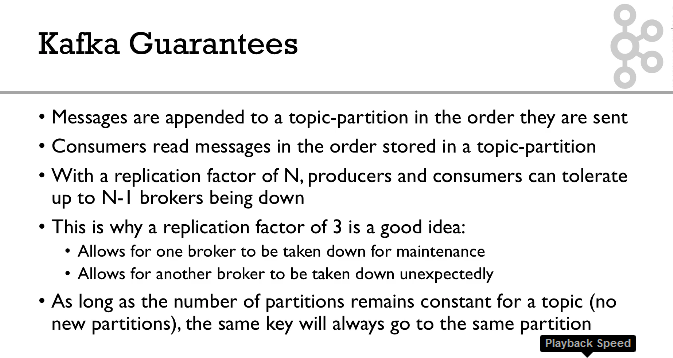
That means that you need to connect to only one broker that connect to the entire cluster.

Each Broker know about the all broker, topic and partitions.

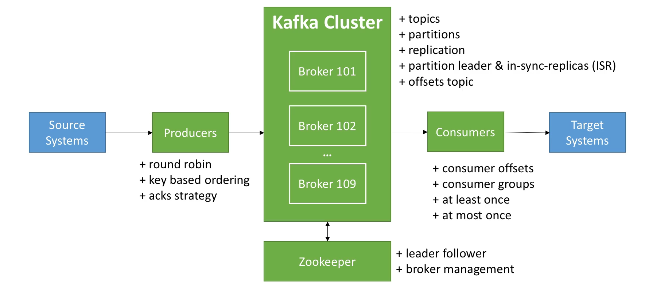


**Zookeeper**

****

****

**Kafka**

****

**Start Zookeeper:**

We can start 1. zookeeper-server-start

2nd way:

Zookeeper-server-start/zookeeper.properties (this start will display all the information of zookeeper)

In the config change the DataDir to our customised dir.

**Kafka:**

Mkdir data/kafka

Change the config/server.properties log basics

Log.dirs to our customized

**Kafka Command Link:**

Kafka-topics : to get all the commands

Kafka-topics –zookeeper 127.0.0.1:2181 –topic first\_topics –create –create –partitions 3 –replication-factor 1

\*\* replication factor must equal or less then the broker, If we use replication factor 2 here it will not create a topics.

To know the list of topics created then:

**Kafka-topics –zookeeper 127.0.0.1:2181 –list**

**Kafka-topics –zookeeper 127.0.0.1:2181 –topics first\_topics –describe**

****

**Kafka-topics –zookeeper 127.0.0.1:2181 –topic second\_topic –create –partitions6 –replication-factor 1**

**Kafka-topics –zookeeper 127.0.0.0.1:2181 –topic second\_topic –delete**

****

\*\* Do not delete topic in windows it will lead to crash the kafka.

**Kafka-console-producer:**

Kafka-console-producer –broker-list 127.0.0.1:9092 –topic –first\_topic

>Hai

>First kafka message

Kafka-console-producer –broker-list 127.0.0.1:9092 –topic first\_topic –producer-property acks=all

>Some message

>just for fun

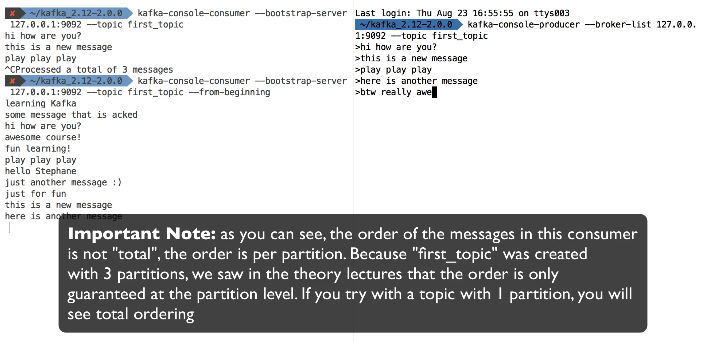
Kafka-console-producer –broker-list 127.0.0.1:9092 –topic new\_topic

This will create a default topic: we can change the default settings in server.properties

Kafka-console-consumer –bootstrap-server 12.0.0.1:9092 –topic first\_topic

To read the message from the beginning then will use the flag at the end of the command **–from-beginning**

**Kafka-console-producer –broker-list 127.0.0.1 –topic first\_topic**



Producer produce the message and consumer consume the message.

**Kafka consumer in group:**

kafka-console-consumer –bootstrap-server 127.0.0.1:9092 –topic first\_topic –group my-first-application.

(for example, if we have 3 partitions then the message goes to the 3 partitions like round robin manner).

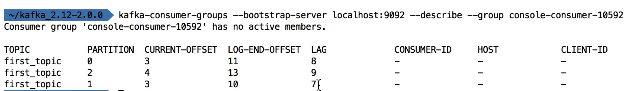
kafka-console-producer –broker-list 127.0.0.1:9092 –topic first\_topic

**Kafka Consumer Groups CLI:**

Kafka-consumer-group –bootstrap-server localhost:9092 –list

When we don’t specify any group then it will automatically producer random consumer group for us.

Kafka-consumer-groups –bootstrap-server localhost:9092 –describe –group console-consumer-10592



**Resetting Offsets:**

Kafka-consumer-group –bootstrap-server localhost:9092 --group my-first-application –reset-offsets –to-earliest –execute --topic first\_topic