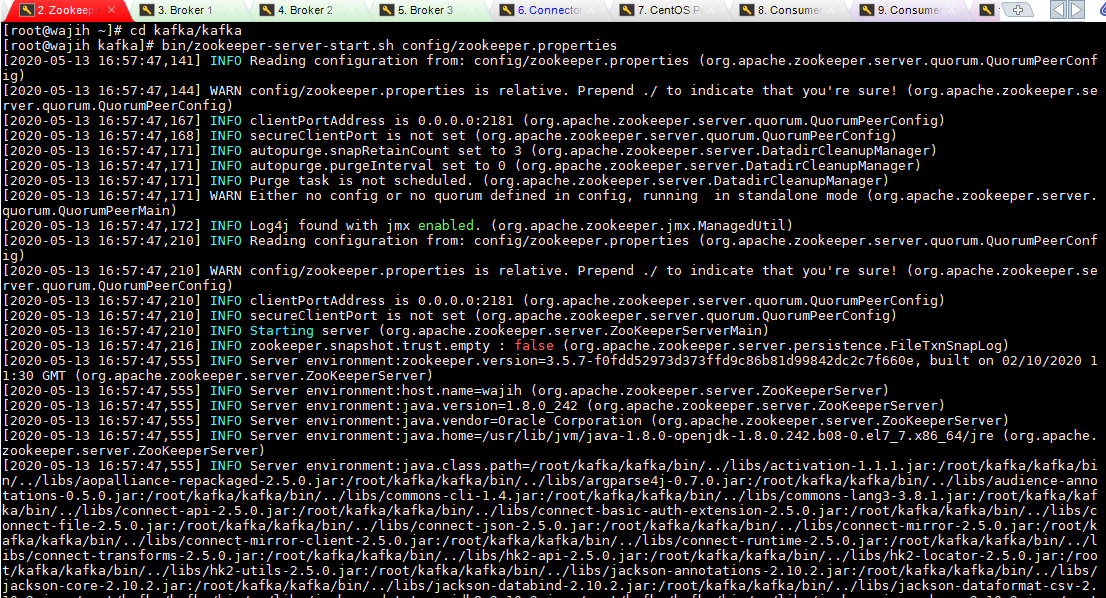
Testing Kafka Behavior

**TEST 2**

We Will Be Doing the following Things

1. Start MS SQL Server (Checking all necessary properties i.e., Enabling CDC etc)
2. Start VMware and Connect MobaXterm
3. Console 1 – Start Zookeeper
4. Console 2,3,4 - Start 3 Brokers respectively
5. Console 5 – Create the Potential Topic That should be created by the Connector when run in the next step
6. Console 4 – Run Connector
7. Console 6,7,8 – Create broker respective Consumers

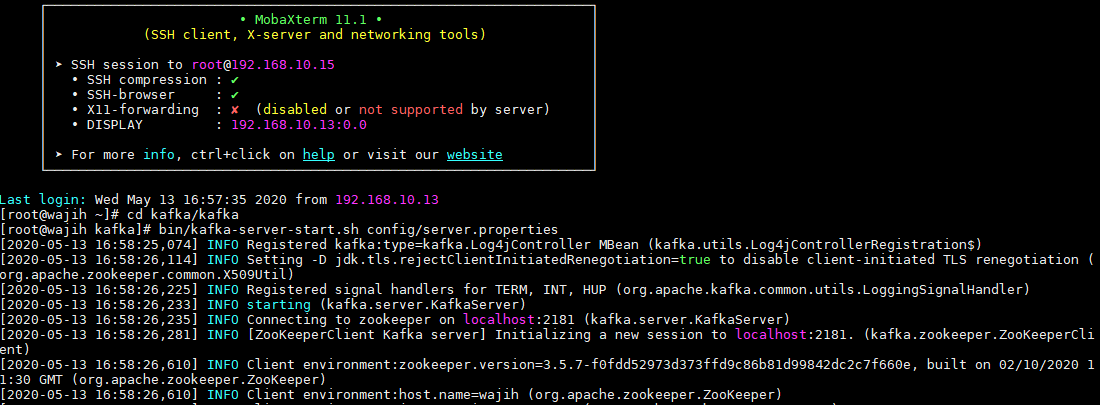
**Zookeeper**



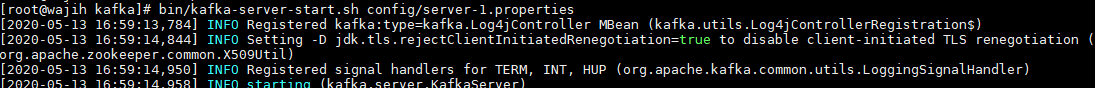
**Brokers**

When Zookeeper is running successfully, now we can run Brokers

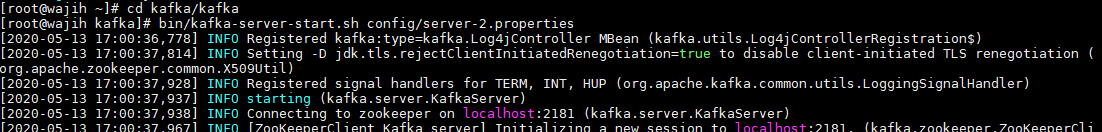
Broker 1



Broker 2



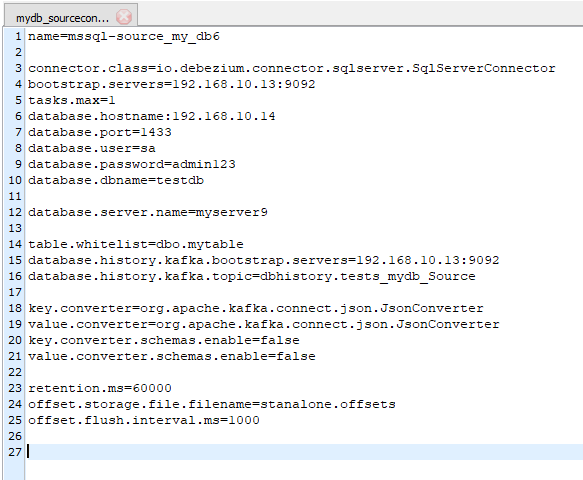
Broker 3



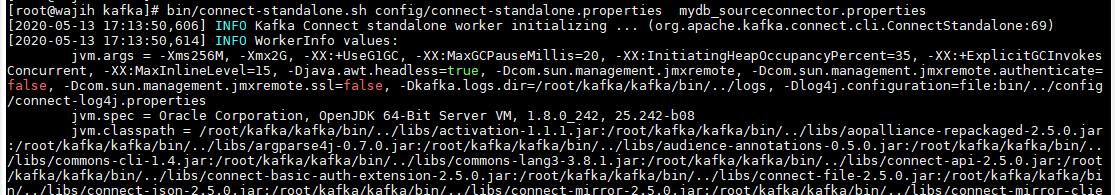
**Connector**

Since we need to connect with My SQL Server, we will create a connector as shown in the picture.

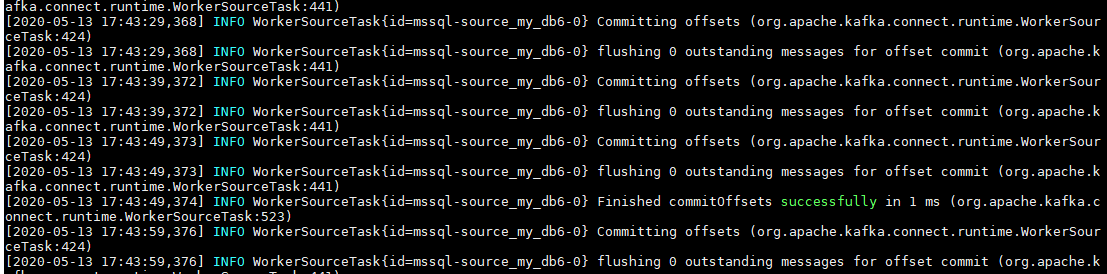
We will run this after creating Topic



Running Connector



Connector Run Successfully

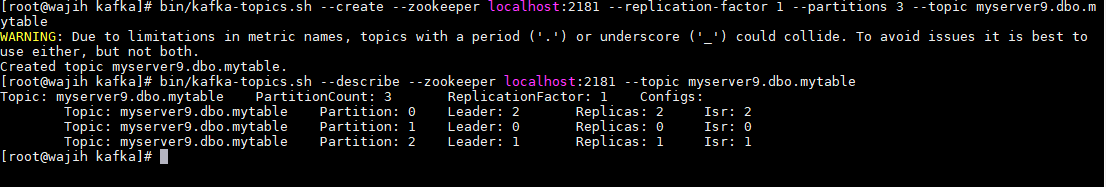


**Topic**

Now we created a topic myserver7.dbo.mytable with

* Replication factor = 1
* Partitions = 3

Topics with descriptions



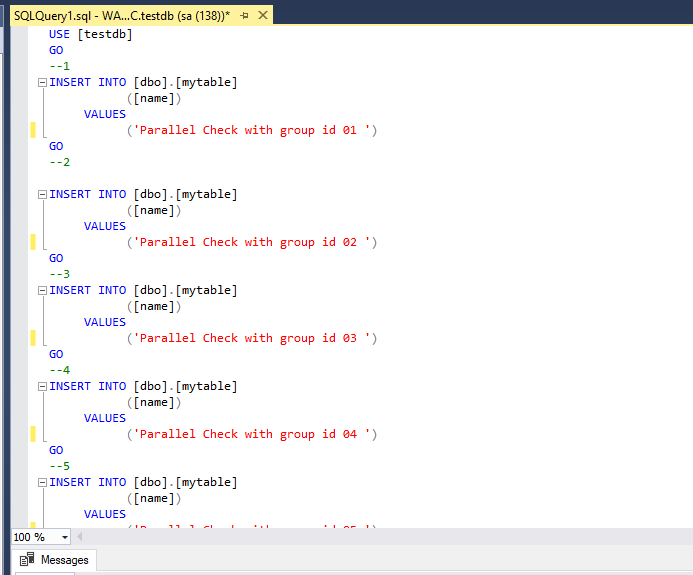
**Consumers**

To get the data that is fetched into the topic by the connector, we will run the consumer.

Since we are testing with three brokers and three consumers, we will run three consumers in three different consoles.

Initially we will get all data from topic to consumer and then we will check instantly put data.

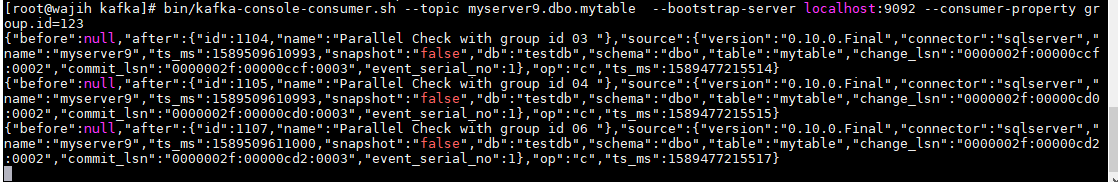
We will push some data from the My Sql Server to the kafka simultaneously.



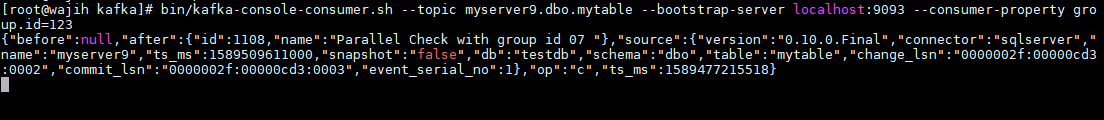
Data is like this till 07.

The SQL query sheet is run all at once and the results in kafka are as follows

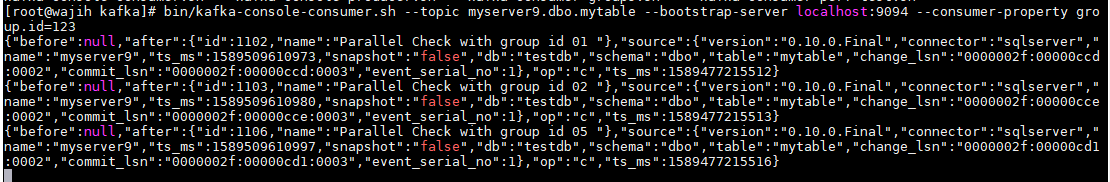
Consumer 1



Consumer 2



Consumer 3



**Analyzing Outputs**

Now, we can clearly see that there are no duplicate data. Each record is consumed only once. Once a message is consumed, its no longer available for other consumer.

Group id 03,04,06 are consumed by consumer 1

Group id 07 is consumed by consumer 2

Group id 01,02,05 are consumed by consumer 3

**Conclusion:**

When we compare the results, We get the following results

* No entry is missing and all data is instantly pushed and has been received by consumers
* Data is Partitioned which is clear from the different order of data in all consumers
* Kafka is working as per theory
* Different partitioned are assigned to different consumers
* This is achieved by having these consumers grouped into one consumer group
* If consumer group is not assigned, kafka creates a new different group id for each consumer that is why those consumers are taken as independent and hence these get all the data

Below picture illustrates the theory of out test.

