**Apache Kafka L2 Class Room Assignments**

[

**Note: - All the assignments are based on Window platform however, participants can execute on Linux platform as well.**

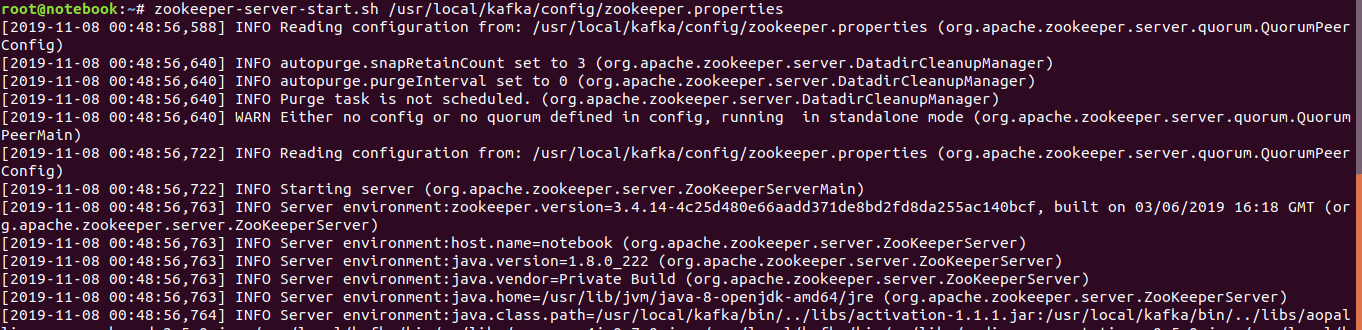
**Participants have to upload the steps/commands and screenshots for the below assignment**

1. Download Apache Kafka server package and configure along with the zookeeper server. Start both the server.

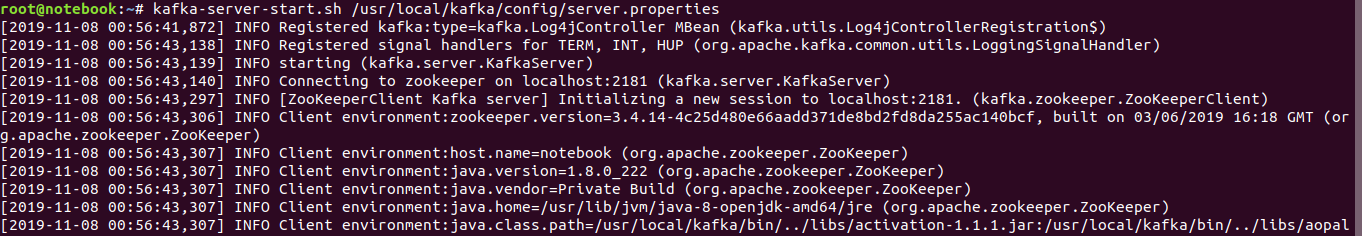
**Note:** **share the steps to configure properties and snapshot of the configured file wherever Applicable.**

**Ans.**

Started Zookeeper Server



Started Kafka Server



1. Create a Topic by name **“Wipro”** using a command line interface with replication factor and partitions as 1.

**Note: Share the command and screenshot.**

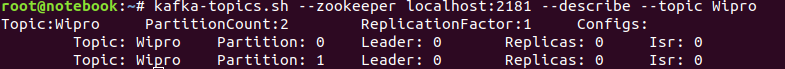
**Ans.**



1. Execute the command to describe topic “Wipro” and list all the topics

**Note: share the command and screenshot**

**Ans.**

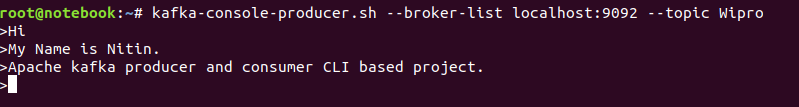




4) Publish-Subscribe Messaging

1. Open the command line interface and execute the command to publish a message on Specific topic. (Topic that you created).

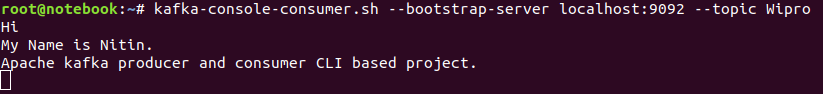
**Ans.**



1. Open another command line interface and execute the command to subscribe to a specific topic where you published the message.

Test by publishing message.

**Ans.**



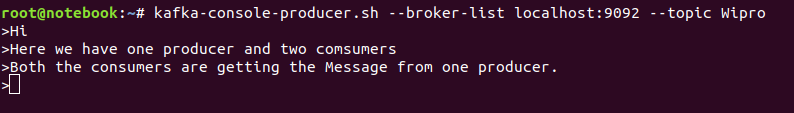
**Note: Share the both publish - subscribe commands along with the screenshots.**

1. Create a two consumer for the same topic using command line interface. Observe whether both the consumers are receiving, the messages or not.

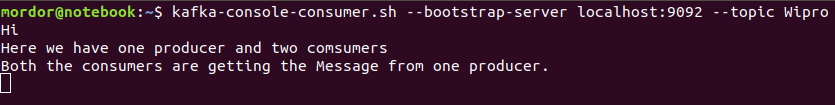
**Note: Share the screenshots.**

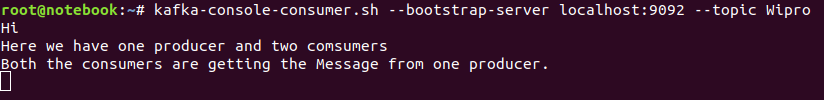
**Ans.**

**Producer**



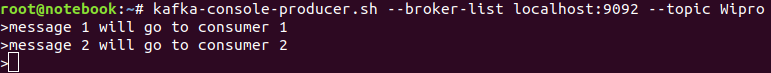
**Consumers**





1. Create a two consumer in a Group **“GroupTest”** for the topic **“Wipro”**. Using another command line interface Publish the message to the topic **“Wipro”**. Observe the behavior.

Ans.

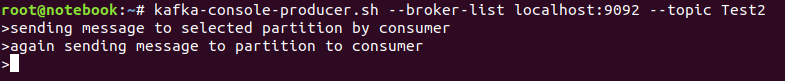




**Note: Share the commands for the above activity along with the screenshot.**

1. Create a New Topic **“Test2”** with two partitions and single replication factor.
2. Using command line interface publish a message to the new Topic **“Test2”.**

**Ans.**



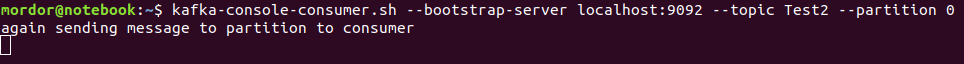
1. Open a another command line interface and execute the command to subscribe to the partition 1 of the topic **‘Test1’**

**Ans.**



1. Open a another command line interface and execute the command to subscribe to the partition 2 of the topic ‘Test1’

**Ans.**



1. Observe whether published messages are stored in two partitions or Not

**Ans.** Yes

**Note: Share all the commands executed for the above activity along with the screenshot.**

1. **Creating a Replication using command line interface**
2. Stop all the servers to set up a new cluster environment.
3. Configure **server.properties** file in config folder.

**Hint:** copy the existing server.properties file in the same folder (3 files) and configure as per requirements. For each server there should be one server.properties file. Name should be unique (**Ex:** **server-1.properties, server-2.properties, and server-3.properties**)

1. Open each properties file and change the **broker id, listening port number and log folder name**

**NOTE:** the broker.id property is the unique and permanent name of each node in the cluster. We have to override the port and log directory only because we are running these all on the same machine and we want to keep the brokers from all trying to register on the same port or overwrite each other's data.

1. Open **3 command line interface**. In each command line interface start one server by specifying appropriate “**server.properties”** file. For example, command line 1 start the server that is configured in server1.properties file. Similarly start other two server using different command line interface.
2. Open another command line interface and create a new topic “replicationtest” with 3 replication factor and 1 partitions.
3. Publish the message to a new topic “replicationtest”

Ans.



1. Execute the command to **describe** the topic **“replicationtest”** to know each broker details. Details like which broker is a l**eader** for the given topic, number of **replicas** and **in-sync replicas**.

Ans.



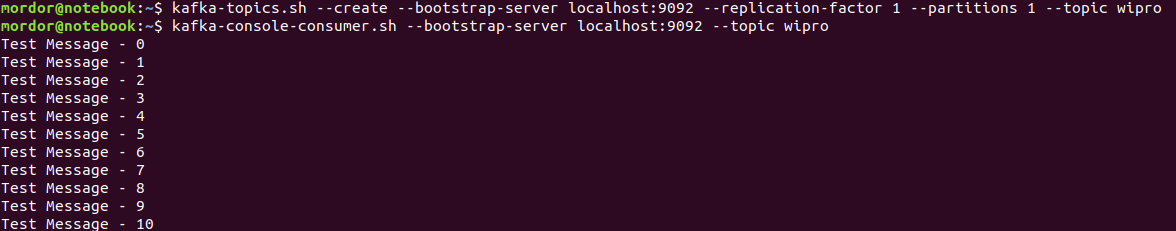
1. To test **Fault-tolerance**, stop the Node, which is acting as **leader**. Go to appropriate command line interface and execute stop command. Execute the command to describe the topic “replicationtest” and observe who is the **next leader** and other details. Repeat the steps by publishing few more messages.

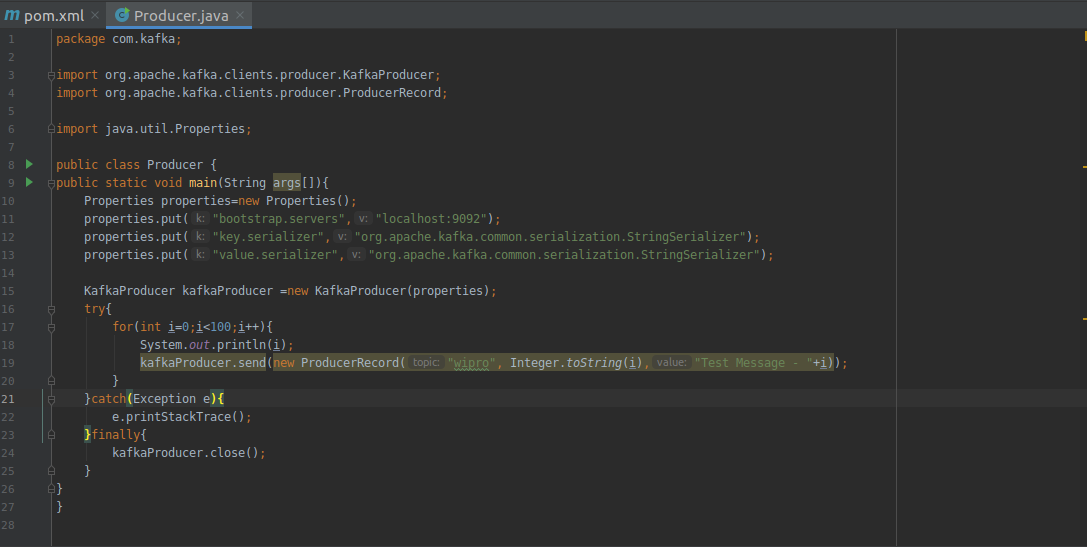
Ans.

**NOTE: Share the steps and commands to complete the above activity along with the screenshot**

1. Write a simple **Java Application** to publish (**Producer**) a message to a specific topic. Open a command line interface to consume a message that was published from java application.

Ans. Java code is Provided in the Repo





**Note: you can use any IDE/Editor to write Java Code. Download the required libraries to write a java code.**

**Share the Java code to publish the message along with the screenshot**

10) Write a simple java Application to consume (**Consumer**) a message from the Kafka node. You can test the application by executing both Producer (exercise 9) and consumer application.

**Note: you can use any IDE/Editor to write java code. Download required libraries to write a code.**

**Share the Java code to consume Message from Kafka Node along with the screenshot.**