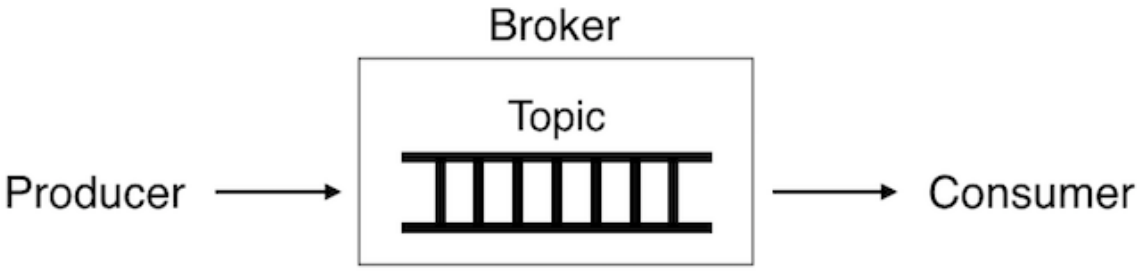
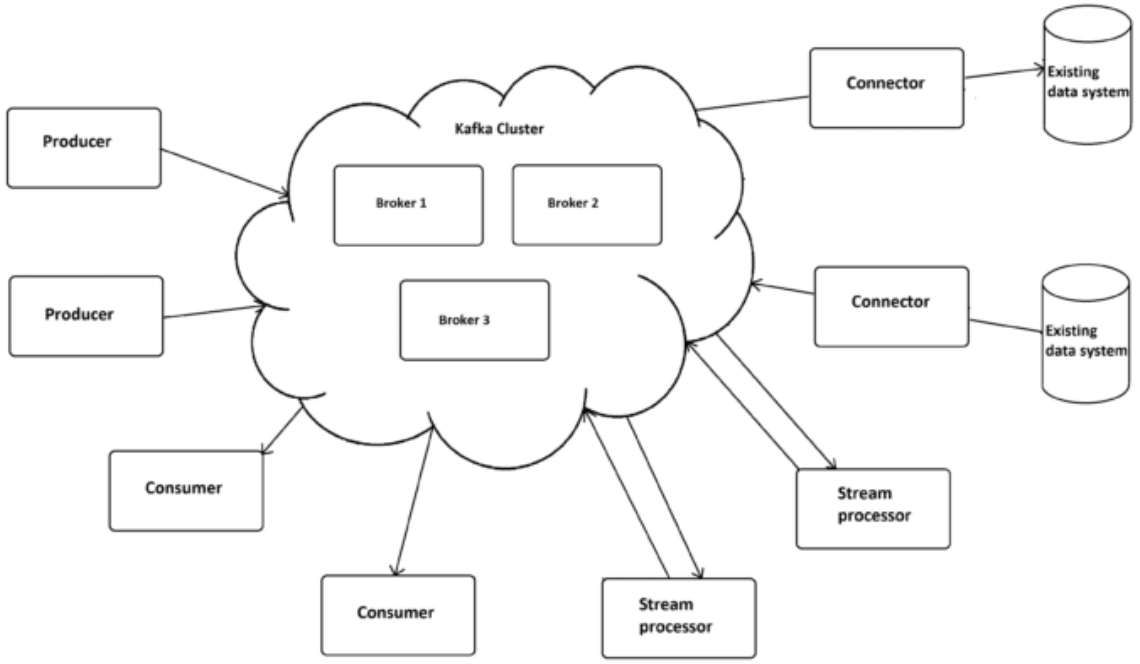
Contents





Kafka Fundamentals :

What is Kafka
Why Kafka ? (Publish Subscribe Based Messaging)
What is Topic ? (Logical Categorisation)
What are Producer and Consumer and Consumer Groups ?
What is Event/Record/Message ?
What is Broker/Bootstrap Server ?(Kafka server)
What is Kafka Cluster ? (Group of Kafka Servers)
What is the Replication Factor ?
What is Zookeeper ?
What is Partition ?
What is a Leader ?

Session 17

Kafka Notes
https://drive.google.com/drive/folders/1EUkcELGjf7jlAXHMpYyJs2XVIfgREj9n?usp=sharing

Kafka Commands
https://kafka.apache.org/quickstart

Challenge 1: If we have 2 apps A and B . A is on host H1 and port P1 while B is on host H2 and port P2 .How can A exchange data with B ?

Challenge 2: Is Kafka promoting tight coupling or loose coupling between producers and consumers ?

Challenge 3: Can a consumer for one topic , be the producer from another topic ?
HINT : Netflix's Recommendation System consumes the movie details from Netflix and produces a recommended movie list which gets consumed by Netflix .It occurs in different topics .

Challenge 4: Does Kafka server start if there is no zookeeper ?

Challenge 5: What is Retention Time and where is Kafka server.properties file ?

Challenge 6: What does Polling mean when we say "Broker does not send messages to consumers , consumers constantly polls the broker" ?
HINT : The poll() method is the function a Kafka consumer calls to retrieve records from a given topic.

Challenge 7: What is the meaning of the statement "Consumer always reads from the leader" ?

HINT : Leader is at the partition level .Suppose Replication factor is 1 and Partition Count is 5 , then each partition will be leader .But if Replication factor is 2 and Partition Count is 5 , then across both kafka brokers there will be a leader among P1 and P1' , P2 and P2' and so on.

Challenge 8: What is clientPort in zookeeper.properties ? How is it different from admin.serverPort in zookeeper.properties?

Challenge 9: What is the default port of Kafka server ?

Challenge 10: What is the broker id of our server and where do we get it ?
HINT : You can see it in /usr/local/etc/kafka/server.properties

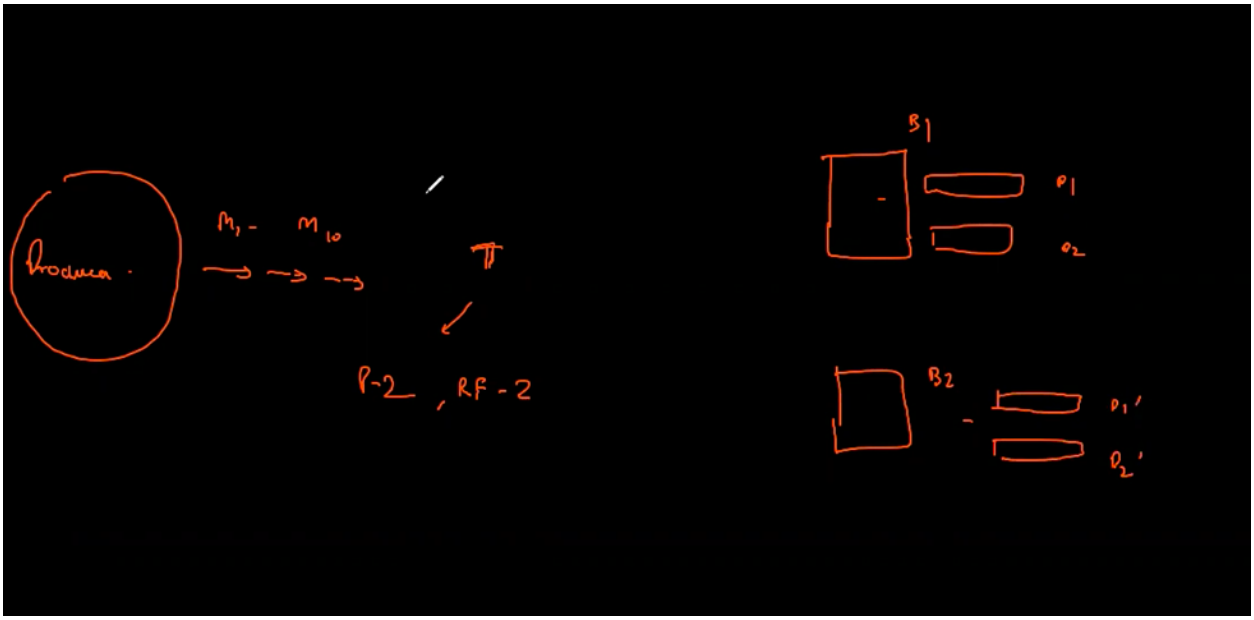
Challenge 11: What is the use of __*consumer_offsets* topic in the list of topics of your broker server ?

Challenge 12: What is the significance if the replication factor of the topic is 1 ?
HINT : It means that this topic is stored in only one broker.

Challenge 13: Why do we have multiple partitions in a topic? Why not have a single Queue /Partition and store all messages there itself ?
HINT : To increase the rate of consumption because if there is only partition , then only one consumer of a consumer group can consume at one point of time.

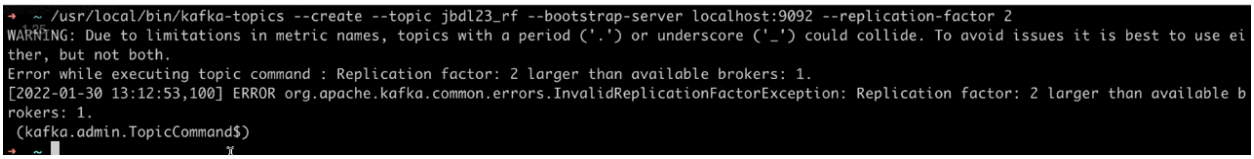
Challenge 14: Let's try to understand the situation when we will have a Topic T and partition count as 2 and replication factor as 1 ?

Challenge 15: Let's try to understand the situation when we will have a Topic T and partition count as 2 and replication factor as 2 ? What is the use of replication factor as 2 ?
HINT : RF 2 will ensure there will be 2 partitions each in both server nodes .Data will be the same in the partitions across the 2 servers . We use rf2 as a backup for if one server fails .



Session 17

Challenge 16: If we have created one broker , one topic , 2 partitions ,RF as 2 . Why did we get error on our command below :



TASK 1: Install Kafka
https://www.apache.org/dyn/closer.cgi?path=/kafka/3.1.0/kafka_2.13-3.1.0.tgz

TASK 2: Start zookeeper

```
# Start the ZooKeeper service # Note: Soon, ZooKeeper will no longer be required by Apache Kafka.

$ bin/zookeeper-server-start.sh config/zookeeper.properties
```

TASK 3: Start Kafka server

```
# Start the Kafka broker service

$ bin/kafka-server-start.sh config/server.properties
```

TASK 4 : Create a Kafka Topic in your current broker server running on 9092 port

```
$ bin/kafka-topics.sh --create --topic topic1 --bootstrap-server localhost:9092
```

TASK 5 : Please list all topics in your current broker server running on 9092 port

```
$ bin/kafka-topics.sh --bootstrap-server=localhost:9092 --list
```

TASK 6 : Please describe the topic that you recently created in your current broker server running on 9092 port

```
$ bin/kafka-topics.sh --bootstrap-server=localhost:9092 --describe --topic
```

TASK 7 : How to create a new topic with replication factor 2 and default partitions ?

HINT : +Make sure zookeeper is already running.
+create a new broker by defining a new server2.properties file .Change the broker id , port and log.dirs path .
+start this server too as we did the earlier one .
+Now create a topic
`bin/kafka-topics.sh --create --topic topic3 --bootstrap-server localhost:9092 --replication-factor 2`
+You can verify RF by describing the topic(Isr is in sync replicas.)

Challenge 1: What will happen if I give --replication-factor 3 in the above case without creating the third broker ?
HINT : Error

TASK 8 : Create a new topic with 2 partitions and with replication factor 2 ?

HINT : Create the topic
`bin/kafka-topics.sh --create --topic topic3 --bootstrap-server localhost:9092 --partitions 2 --replication-factor 2`
Now let's describe the above created topic.
We can conclude that the leader is at the partition level .Every partition is present on both servers . Leader 0 for Partition 1 means that for our topic's Partition 1 , we have the leader on broker 0 . The Partiton 1 present on broker 1 is follower/slave .

TASK 9 :Task is to create a new topic with 3 partitions and with replication factor 2 when brokers are 2 .

Now spawn another broker3 (kafka server node) .Task is to create a new topic with 3 partitions and with replication factor 2 ?

HINT :
Even if the replication factor is 2 , Kafka will equally distribute the leaders among all the nodes .Every node becomes a leader of 1 partition each .This time we have three brokers . Three partitions scale across 3 nodes and each node becomes the leader for one partition .

```
→  ~ /usr/local/bin/kafka-topics --create --topic jbdl23_partitions_rf --bootstrap-server localhost:9092 --replication-factor 2 --partitions 3
WARNING: Due to limitations in metric names, topics with a period ('.') or underscore ('_') could collide. To avoid issues it is best to use ei
ther, but not both.
Created topic jbdl23_partitions_rf.
→  ~ /usr/local/bin/kafka-topics --bootstrap-server localhost:9092 --describe --topic jbdl23_partitions_rf
Topic: jbdl23_partitions_rf     TopicId: hHfW0cCXS3C9DqzfCztAkg PartitionCount: 3       ReplicationFactor: 2    Configs: segment.bytes=10737418
24
        Topic: jbdl23_partitions_rf     Partition: 0    Leader: 1       Replicas: 1,0   Isr: 1,0
        Topic: jbdl23_partitions_rf     Partition: 1    Leader: 0       Replicas: 0,1   Isr: 0,1
        Topic: jbdl23_partitions_rf     Partition: 2    Leader: 1       Replicas: 1,0   Isr: 1,0
→  ~ /usr/local/bin/kafka-topics --create --topic jbdl23_final --bootstrap-server localhost:9092 --replication-factor 2 --partitions 3
WARNING: Due to limitations in metric names, topics with a period ('.') or underscore ('_') could collide. To avoid issues it is best to use ei
ther, but not both.
Created topic jbdl23_final.
→  ~ /usr/local/bin/kafka-topics --bootstrap-server localhost:9092 --describe --topic jbdl23_final
Topic: jbdl23_final     TopicId: VlgsEWapQg6z_X0RgkarJg PartitionCount: 3       ReplicationFactor: 2    Configs: segment.bytes=1073741824
        Topic: jbdl23_final     Partition: 0    Leader: 0       Replicas: 0,1   Isr: 0,1
        Topic: jbdl23_final     Partition: 1    Leader: 2       Replicas: 2,0   Isr: 2,0
        Topic: jbdl23_final     Partition: 2    Leader: 1       Replicas: 1,2   Isr: 1,2
→  ~
```

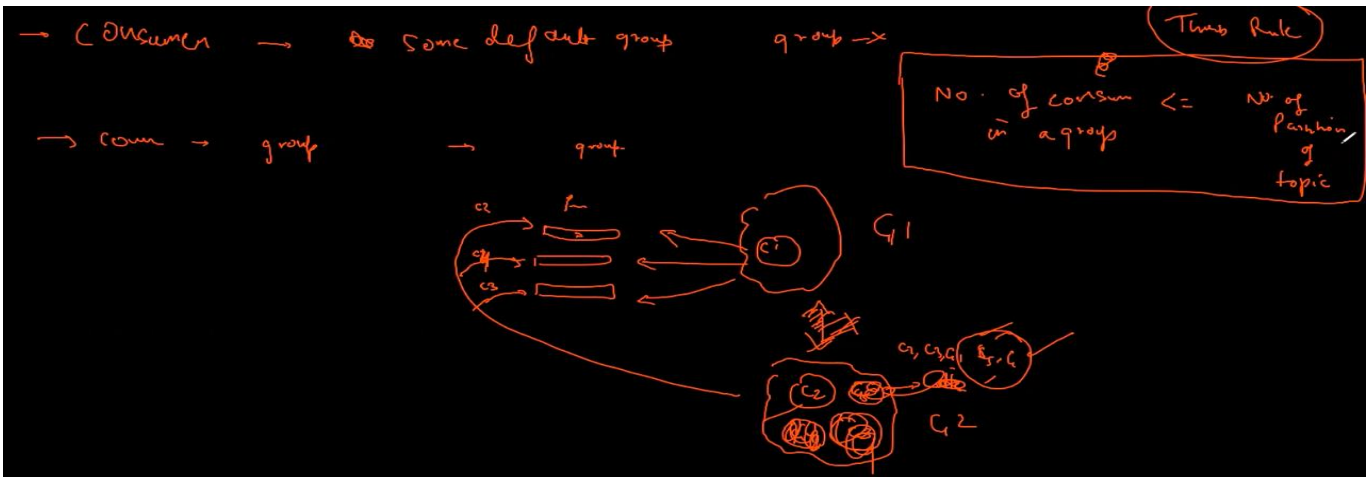
Let's analyze the above image .The one on top shows topic with 3 partitions and with replication factor 2 when we had only 2 brokers.The bottom one shows topic with 3 partitions and with replication factor 2 when we had 3 brokers.

TASK 10: Run some consumers now .

HINT : Let's start 3 consumers now in the same group .

One partition will be read from one consumer only in a group .This means that suppose A and B are consumers in a group , A and B can't read from one partition .Also when in a group there is only one consumer , it will read from all the consumers .

```
$ bin/kafka-console-consumer.sh --topic topic1 --bootstrap-server localhost:9092 —-group jbdl_group
```



```
Last login: Sun Jan 30 13:24:49 on ttys005
➜  ~ /usr/local/bin/kafka-console-consumer --bootstrap-server localhost:9092 --topic jbdl23_final --group jbdl23_def_group
hi
```

TASK 11: Start a producer now and start producing messages .

$ bin/kafka-console-producer.sh --topic topic1 --bootstrap-server localhost:9092

TASK 12 : Let's describe the consumer group now .We can see each consumer has one partition of the topic allotted . What will happen if we close one of our 3 consumers ?

HINT :

```
Last login: Sun Jan 30 13:46:06 on ttys005
➜  ~ /usr/local/bin/kafka-consumer-groups --bootstrap-server localhost:9092 --group jbdl23_def_group --describe --members

GROUP           CONSUMER-ID                                                 HOST            CLIENT-ID                   #PARTITIONS
jbdl23_def_group consumer-jbdl23_def_group-1-304a9ecc-8771-4ea2-8bfa-9c3b19a0bc88 /127.94.0.2     consumer-jbdl23_def_group-1 2
jbdl23_def_group consumer-jbdl23_def_group-1-31472e31-465f-4d0d-a345-ad1d2607bdec /127.94.0.2     consumer-jbdl23_def_group-1 1
➜  ~
```

Challenge 2: In the above scenario, what happens if all the remaining consumers ?

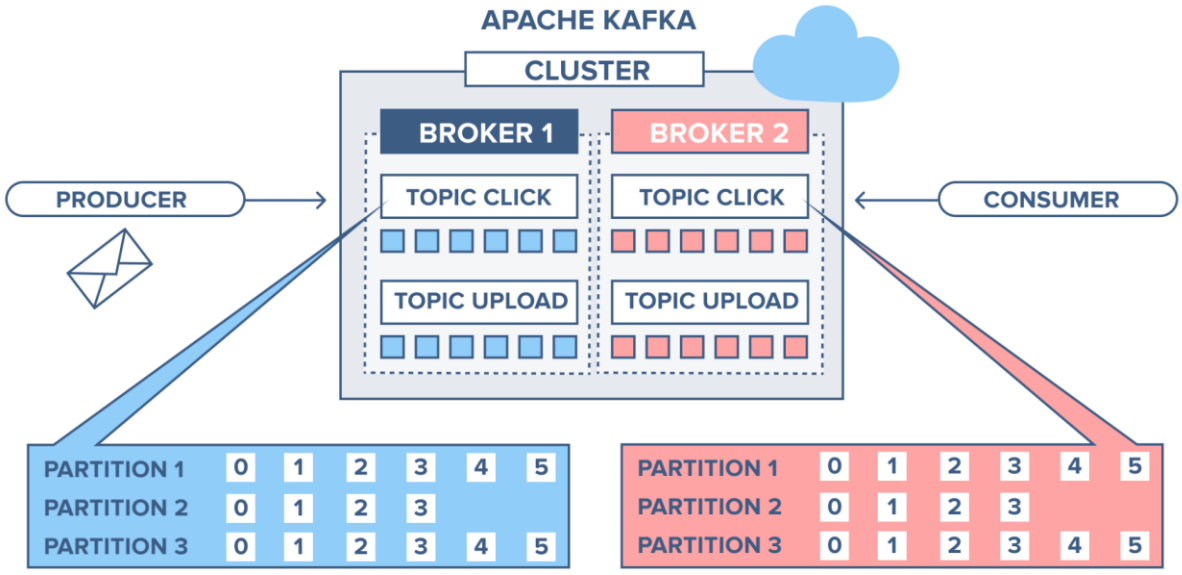
Challenge 3: How to change the default value for partition count in Broker config ?

References :
https://www.geeksforgeeks.org/how-to-install-and-run-apache-kafka-on-windows/

https://medium.com/event-driven-utopia/understanding-kafka-topic-partitions-ae40f80552e8

https://www.geeksforgeeks.org/apache-kafka/

https://www.geeksforgeeks.org/why-apache-kafka-is-so-fast/


APACHE KAFKA
CLUSTER
BROKER 1
BROKER 2
PRODUCER
CONSUMER
TOPIC CLICK
TOPIC CLICK
TOPIC UPLOAD
TOPIC UPLOAD
PARTITION 1 0 1 2 3 4 5
PARTITION 2 0 1 2 3
PARTITION 3 0 1 2 3 4 5
PARTITION 1 0 1 2 3 4 5
PARTITION 2 0 1 2 3
PARTITION 3 0 1 2 3 4 5