**Connecting Confluent to kafka using CLI:**

Create a File(confluent.properties) using vim in the below folder and add the configuration.

C:\kafka\_2.13-3.6.1\bin\windows

**File details:**

C:\kafka\_2.13-3.6.1\bin\windows>type confluent.properties

**security.protocol**=SASL\_SSL

**sasl.jaas.config**=org.apache.kafka.common.security.plain.PlainLoginModule required **username**='GDF4KRC6KC4QRC2E' **password**='KVy728UsKVjtGtkPKB3t7Jk2WITqNn39fWhMdkqk8+P1mTZVJZtF5p6pLC7WGvHD';

**sasl.mechanism**=PLAIN

|  |
| --- |
| **Note:**  Username , password and bootstrap server details you will get from a file(**API keys in confluent which was added by you initially**) which can be downloaded from confluent in web. |

**List topics:**

C:\kafka\_2.13-3.6.1\bin\windows>*kafka-topics.bat --command-config confluent.properties --list --bootstrap-server pkc-921jm.us-east-2.aws.confluent.cloud:9092*

topic\_0

**Create Topic:**

C:\kafka\_2.13-3.6.1\bin\windows>*kafka-topics.bat --command-config confluent.properties --bootstrap-server pkc-921jm.us-east-2.aws.confluent.cloud:9092 --create --topic orderdemo*

Created topic orderdemo.

**Create Topic with n partitions:**

C:\kafka\_2.13-3.6.1\bin\windows>*kafka-topics.bat --command-config confluent.properties --bootstrap-server pkc-921jm.us-east-2.aws.confluent.cloud:9092 --create --topic second-topic*

*--partitions 1*

Created topic second-topic.

Describe Topics:

C:\kafka\_2.13-3.6.1\bin\windows>*kafka-topics.bat --command-config confluent.properties --bootstrap-server pkc-921jm.us-east-2.aws.confluent.cloud:9092 --topic second-topic --describe*

Topic: second-topic TopicId: H3\_bj3azTYazD\_pVjJlI5Q PartitionCount: 1 ReplicationFactor: 3 Configs: min.insync.replicas=2,segment.bytes=104857600,message.format.version=3.0-IV1,max.message.bytes=2097164

Topic: second-topic Partition: 0 Leader: 2 Replicas: 2,4,3 Isr: 2,4,3

**Add Replication Factor while creating topic:**

C:\kafka\_2.13-3.6.1\bin\windows>*kafka-topics.bat --command-config confluent.properties --bootstrap-server pkc-921jm.us-east-2.aws.confluent.cloud:9092 --create --topic second-topic --partitions 1 --replication-factor 3*

Created topic second-topic.

**Delete a topic:**

C:\kafka\_2.13-3.6.1\bin\windows>kafka-topics.bat --command-config confluent.properties --bootstrap-server pkc-921jm.us-east-2.aws.confluent.cloud:9092 --topic second-topic --delete

**Console Producer**

**Using Kafka Console Producer:**

C:\kafka\_2.13-3.6.1\bin\windows>*kafka-console-producer.bat --producer.config confluent.properties --bootstrap-server pkc-921jm.us-east-2.aws.confluent.cloud:9092 --topic topic\_0*

>Hello from kafka CLI

>Message Reached

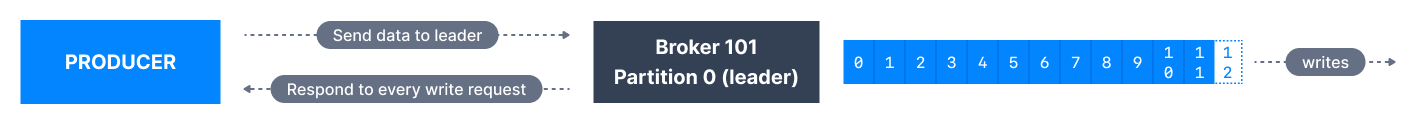
A screenshot of a computer

Description automatically generated

Acks:

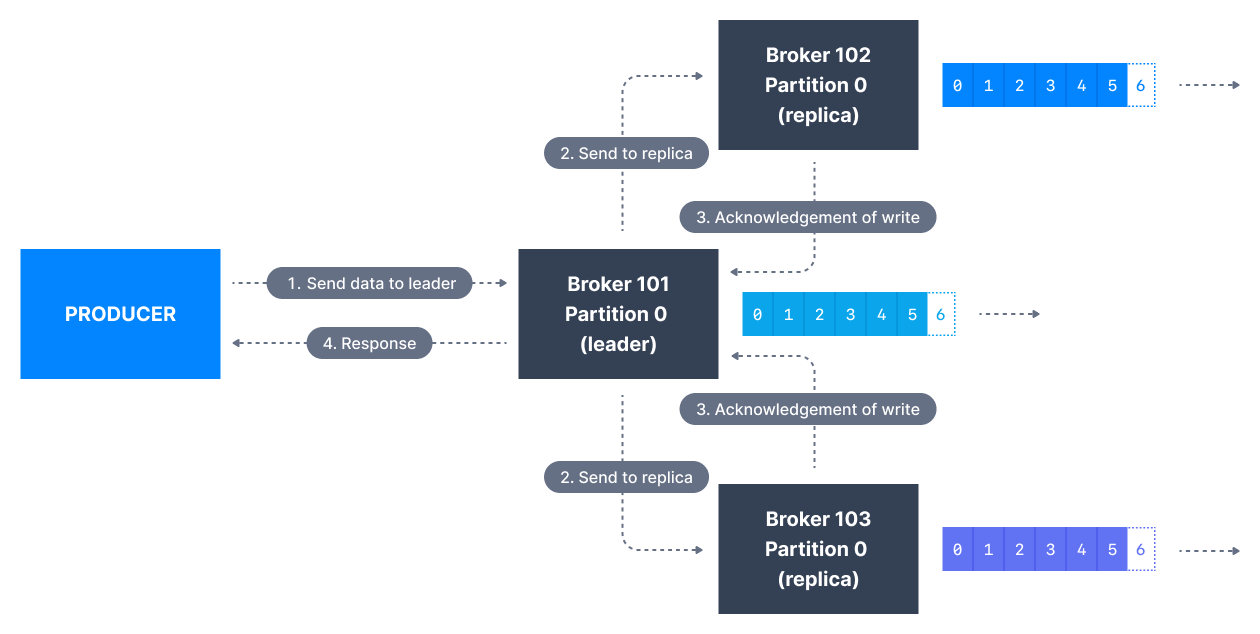
When acks=0 producers consider messages as "written successfully" the moment the message was sent without waiting for the broker to accept it at all.

When acks=1 , producers consider messages as "written successfully" when the message was acknowledged by only the leader.



**acks = all**

When acks=all, producers consider messages as "written successfully" when the message is accepted by all in-sync replicas (ISR).



**Usage:**

C:\kafka\_2.13-3.6.1\bin\windows>*kafka-console-producer.bat --producer.config confluent.properties --bootstrap-server pkc-921jm.us-east-2.aws.confluent.cloud:9092 --topic topic\_0 --producer-property acks=all*

>Check for acks from leader

>as well as from all In sinc replicas

|  |
| --- |
|  |

Produce with keys:

C:\kafka\_2.13-3.6.1\bin\windows>*kafka-console-producer.bat --producer.config confluent.properties --bootstrap-server pkc-921jm.us-east-2.aws.confluent.cloud:9092 --topic topic\_0 --property parse.key=true --property key.separator=:*

>name:saikiran

>age:25

A screenshot of a computer

Description automatically generated

**Kafka Console Consumer CLI**

Consuming From Beginning:

--from-beginning

C:\kafka\_2.13-3.6.1\bin\windows>*kafka-console-consumer.bat --consumer.config confluent.properties --bootstrap-server pkc-921jm.us-east-2.aws.confluent.cloud:9092 --topic topic\_0 --from-beginning*

saikiran

25

**Display Key, Value and timestamp:**

Use –formatter kafka.tools.DefaultMessageFormatter

C:\kafka\_2.13-3.6.1\bin\windows>*kafka-console-consumer.bat --consumer.config confluent.properties --bootstrap-server pkc-921jm.us-east-2.aws.confluent.cloud:9092 --topic topic\_0 --formatter kafka.tools.DefaultMessageFormatter --property print.timestamp=true --property print.key=true --property print.value=true --property print.partition=true --from-beginning*

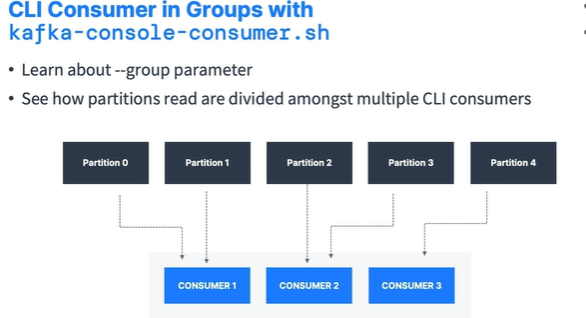
CreateTime:1704273597318 Partition:1 order2 order2-DISPATCHED

CreateTime:1704273597346 Partition:1 order2 order2-OUT\_FOR\_DELIVERY

CreateTime:1704355093027 Partition:0 name saikiran

CreateTime:1704355096167 Partition:0 age 25

Consumer Groups



This approach we can see in production. But to make it happen in local we need to use Round robin algorithm so that messages will go to different partitions in different consumers in a consumer group.

Partitions read are divided amongst multiple consumers in a consumer group.

Consumer Group creation:

Use below property in consumers which you want to make a group.

--group group\_name

Eg:

I have created 1 producer with 1 consumer group(having 3 consumers).

Producer:

C:\kafka\_2.13-3.6.1\bin\windows>*kafka-console-producer.bat --producer.config confluent.properties --bootstrap-server pkc-921jm.us-east-2.aws.confluent.cloud:9092 --producer-property partitioner.class=org.apache.kafka.clients.producer.RoundRobinPartitioner --topic orderdemo*

>order2

>order3

>order4

>o5

>o6

>o7

>08

Consumer Group:

1. C:\kafka\_2.13-3.6.1\bin\windows>kafka-console-consumer.bat --consumer.config confluent.properties --bootstrap-server pkc-921jm.us-east-2.aws.confluent.cloud:9092 --topic orderdemo --group mine --property print.partition=true

Partition:5 order3

Partition:5 o6

1. C:\kafka\_2.13-3.6.1\bin\windows>*kafka-console-consumer.bat --consumer.config confluent.properties --bootstrap-server pkc-921jm.us-east-2.aws.confluent.cloud:9092 --topic orderdemo --group mine --property print.partition=true*

Partition:3 order4

Partition:3 o7

1. C:\kafka\_2.13-3.6.1\bin\windows>*kafka-console-consumer.bat --consumer.config confluent.properties --bootstrap-server pkc-921jm.us-east-2.aws.confluent.cloud:9092 --topic orderdemo --group mine --property print.partition=true*

Partition:1 o5

Partition:1 08

Consumer Group management CLI

A diagram of a group

Description automatically generated