KAFKA: It is a faster streaming and messaging system with msg Delivery Guarante.

other messaging systems in market:

1)websphereMQ

2)RabitMQ

3)Tibco

4)webmethods

all the above systems are not distributeed ,so heavy loads of streaming or messaging cannot be handled

Kafka is a distributed and supports parallel streams or msgs for importing and writing from/into src/target applications.

Drawbacks of flume:

1)No Msg Delivery Guarantee

2)If channel fails, all buffered events will be lost

if multiple channels, then there will be huge delay

------------------------------------------------------------------------------------------------------------------------------------------------------------

KAfka Advantages:

1)Distributed parallel writes and imports

2)Replication facility

3)msg Delivery Guarantee

4)persisted msg feature

5)Heavy loads of reads and writes of messages can be handled.

N ote: In Kafka we can have multiple masters

----------------------------------------------------------------------------------------------------

Kafka Components: 5 components

1)Broker

2)Topics

3)Producers

4)Consumers

5)Zookeeper

1)Broker: It is nothing but kafka server which maintains topics and sever messages .

Broker Responsibilities:

1)Partitions Assignment

2)Load balancing

3)Fault tolerence

4)Monitoring the health of other brokers (heart beating) in the n/w.

---------------------------------------------------------------------------------------------------------------------------------------

2)Topics: Topics are the message holding units

In kafka, messages are stored in topics

Topics are subdivided into partitions,

Partitions are distributed across multiple brokers.

----------------------------------------------------------------------------------------------------------------------------------------

3)Producers :Responsible for writing messages into topics.

Producers can write into one or more topics.

multiple producers can write into a single topic.

----------------------------------------------------------------------------------------------------------------------------------------

4)Consumers : Responsible for reading messages from topics

Consumers can read from multiple topics

multiple consumers can read from a single topic

---------------------------------------------------------------------------------------------------------------------------------------

5)Zookeeper: It acts as a co-ordinating system b/w different Distributed Applications

ex: b/w flume and hadoop

b/w storm and hadoop

b/w hbase and hadoop

b/w kafka and hadoop

Pig------writing to------------>HDFS

--------------------------->No zookeeper required

Hive----writing to------------>HDFS

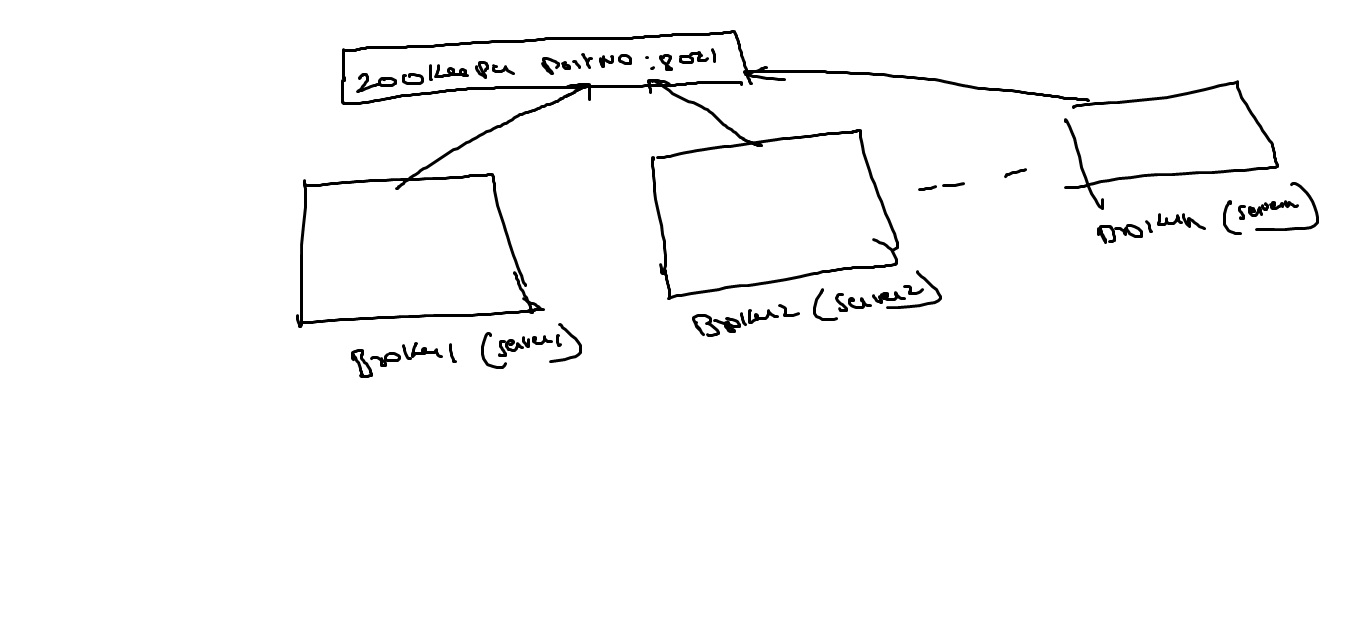
But Hbase/Flume/Kafka---------writing to-------->HDFS--->zookeeper required

Zookeeper is not only for hadoop

ex:java <------->Zookeepeer <--------------> kafka

Zookeeper stores all the coordinating activities in a file called znodes

zookeeper maintains connections and locks and also co-ordinating information required for multiple servers

All the brokers in kafka cluster must use the same zookeeper portno ,then only communication b/w the brokers are possible

Broker------------->means server which maintains topics

Topics------------>stores messages

Producer--------->writes msgs into topics

Consumer------->reads msgs from topics

zookeeper-------->for connections and co-ordinating activities.

----------------------------------------------------------------------------------------------------

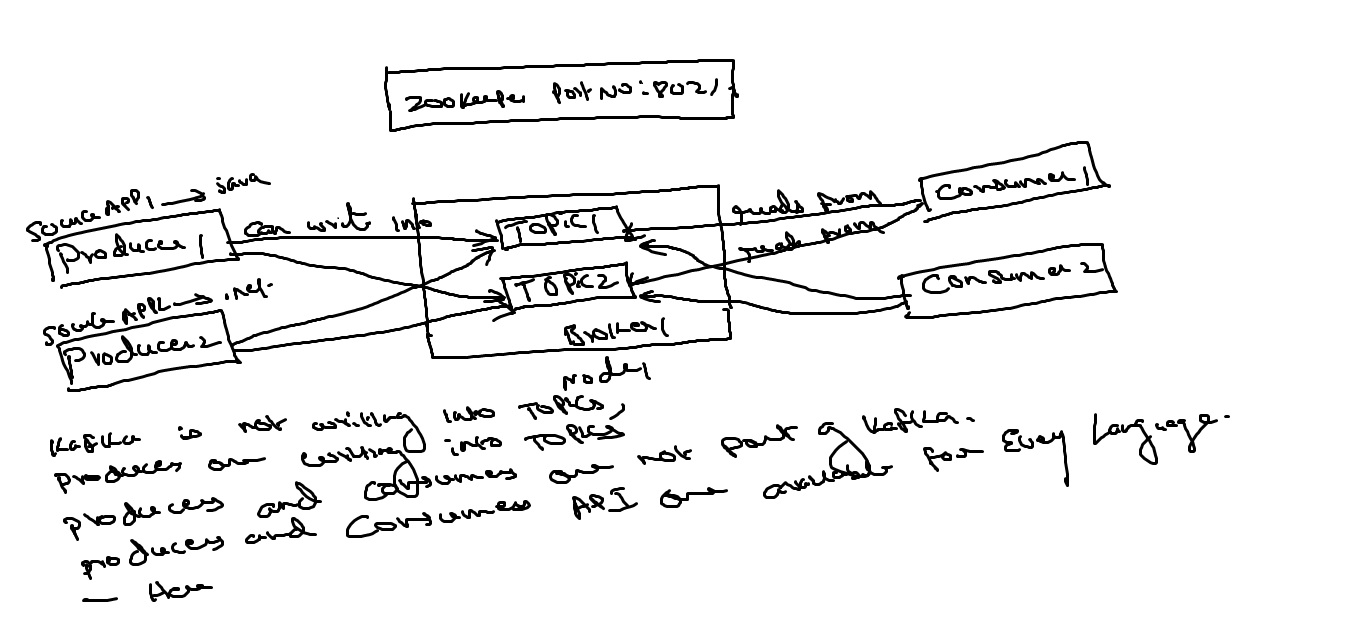
Kafka Architecture:

1)Single node single broker Architecture

2)Single node multi broker Architecture

3)Multi node multi broker Architecture

1)Single node single broker Architcture: see fig2 -kafka2



KAfka is not writing into topics, Producers are writing into topics.

Producers and consumers are not part of kafka.

PRoducers and consumers API is available for every language.

Here Topics can be divided into partitions

1000's of producers(Applications) can write into a single topic

1000's of consumers(Applications) can read from a sigle topic

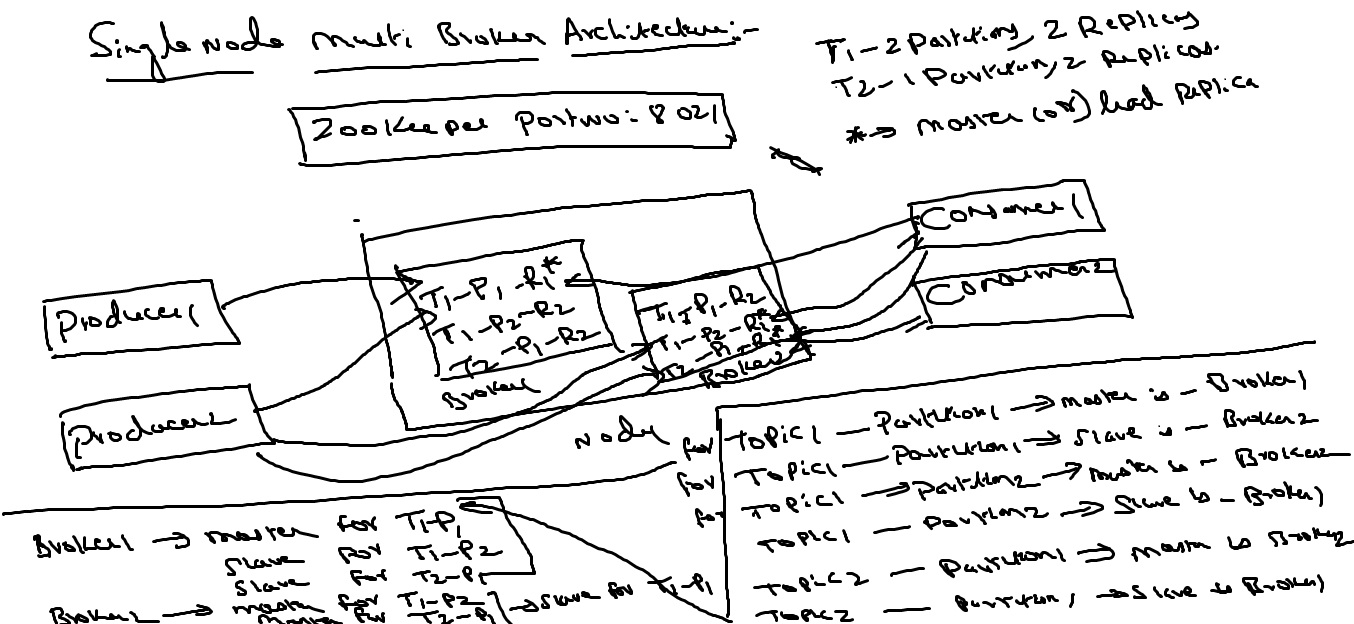
To maintain replications, we need to have multiple brokers.

here if broker is down, then the data is lost ,

so, to overcome this we go with single node multibroker Architecture.

---------------------------------------------------------------------------------------------------------------

2)Single node Multi broker Architecture: see fig3 -kafka3



Note:

-In Hadoop, we have only single master

-In KAfka , we can have multiple masters

-in kafka, each broker(server) can act as a master and slave

Producers or consumers will always read/write from lead replicas or masterreplicas

slave replicas will copy the msgs from lead replicas later

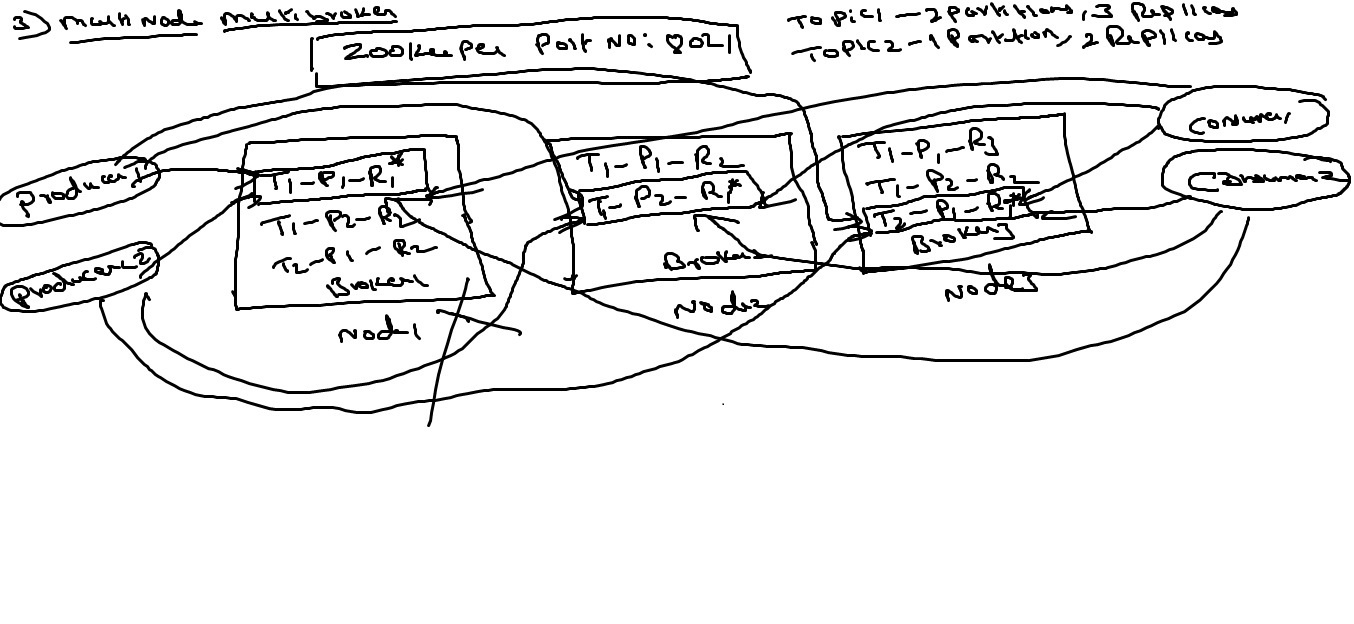
If Broker1 is down----->still topics and partitions are available in Broker2 --->so there is no data loss

but if Node1 is down --->then data is lost ,

so to overcome this ,we go with multinode and multi-broker Architecture

----------------------------------------------------------------------------------------------------

3)Multi-node and Multi-broker Architeecture: see fig4 (kafka4)



Here if one node down,still we have other nodes, so here no data loss

if master replica node is down ,then any of the slave replica becomes master

the same master replica will be created automatically in another node if available , i.e

i.e in kafka, dynamic replica creation is possible

but in hadoop we doesnt have dynamic replica creation