

KAFKA – QUEUE



Topic

- ❖ Need of Messaging System
- ❖ What is Kafka?
- ❖ Kafka Features
- ❖ Kafka Components
- ❖ Kafka architecture
- ❖ Installing Kafka
- ❖ Working with Single Node Single Broker Cluster



Data Pipelines

Communication is required between different systems in the real-time scenario, which is done by using data pipelines.



For Example: Chat Server needs to communicate with Database Server for storing messages

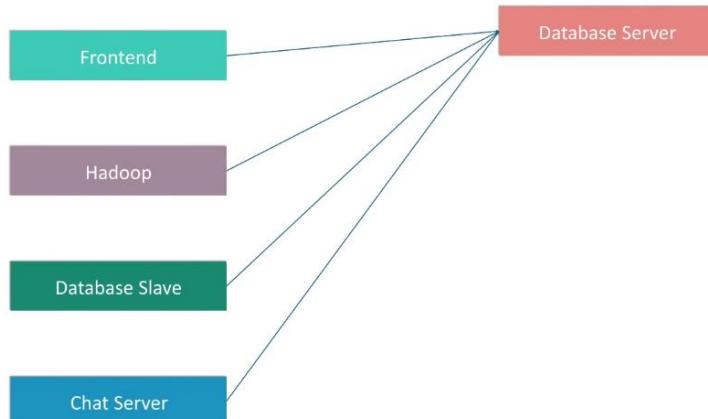
edureka!

Copyright © 2017, edureka and/or its affiliates. All rights reserved.



Increase in number of Nodes

Similarly, there may be many applications wanting to access the Database Server

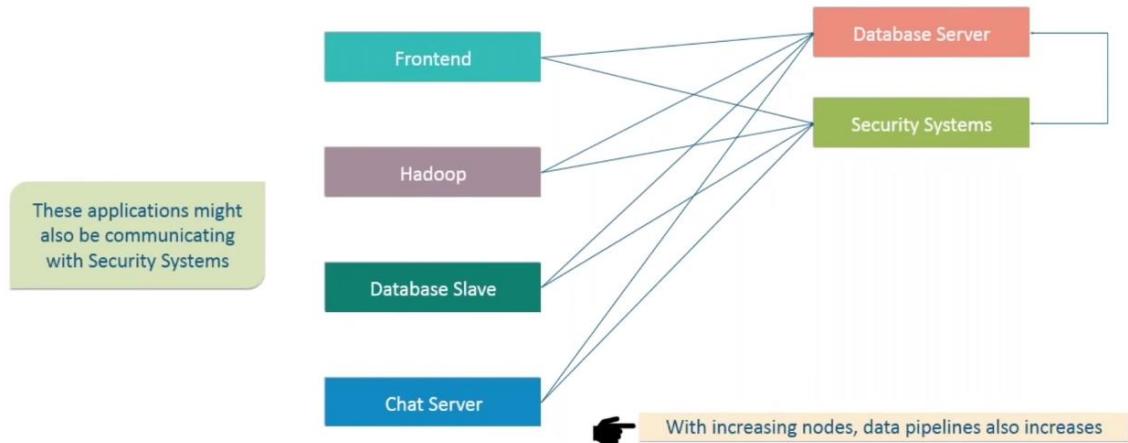


edureka!

Copyright © 2017, edureka and/or its affiliates. All rights reserved.



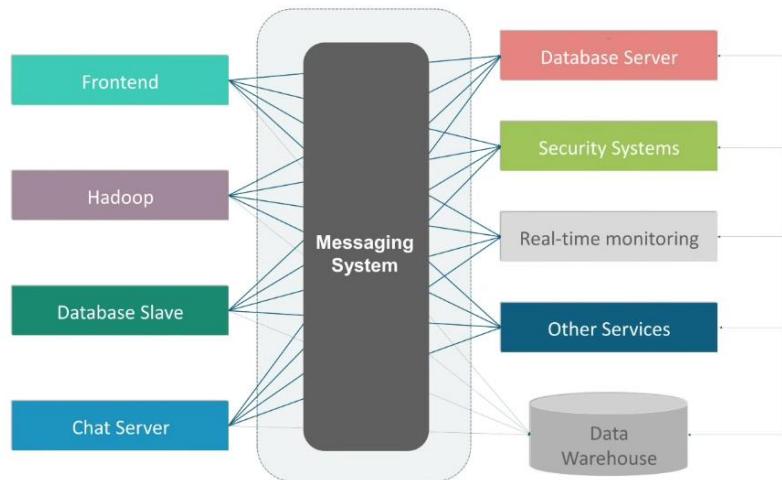
Increase in number of Nodes



With time data pipelines also increases and its difficult for a developer to maintain these pipelines , every pipeline have its own configuration . So organization cant bear any loss of transaction data or any transactions .

Solution to the Complex Data Pipelines

Messaging Systems helps managing the complexity of the pipelines



edureka!

Copyright © 2017, edureka and/or its affiliates. All rights reserved.



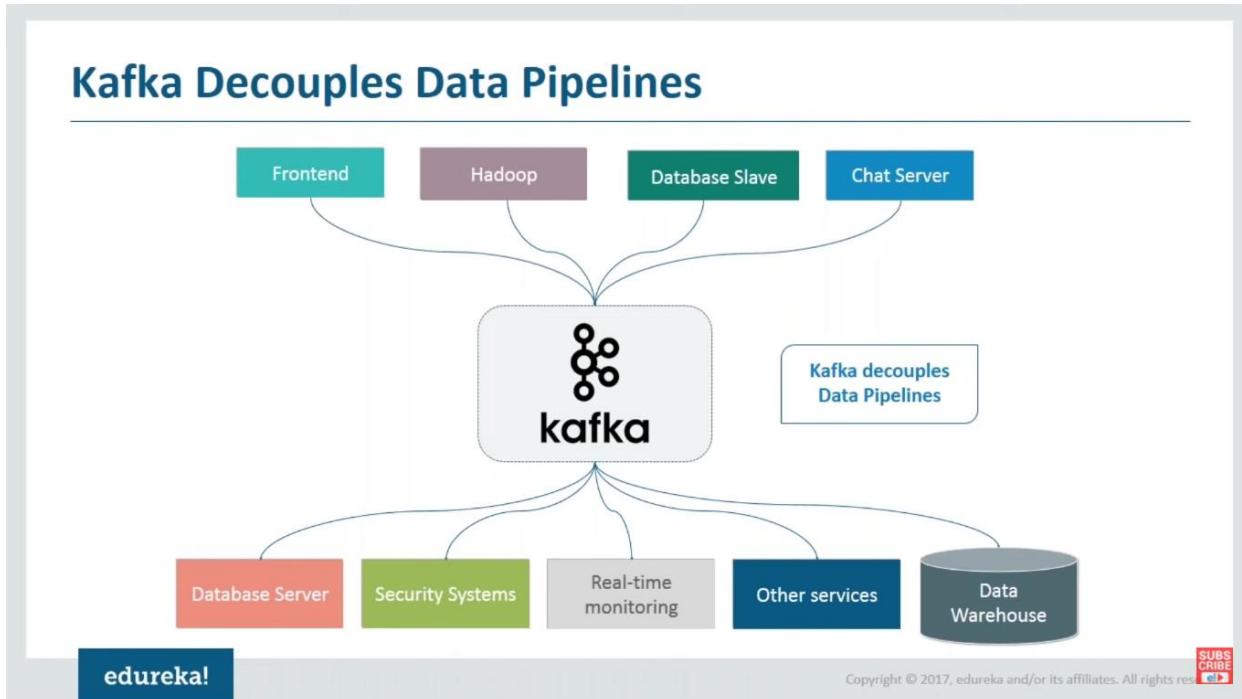
Let's See How Kafka Solves the Problem

edureka!

Copyright © 2017, edureka and/or its affiliates. All rights reserved.



Kafka Decouples Data Pipelines



Applications that sends the data to kafka called producers . And applications those listen the events called consumers.

We can have multiple subscribers for the same set of data .

Ex : Radio Transmission

(Diff people can listen on radio aysnc)

What is Kafka?

- **Apache Kafka** is a distributed *publish-subscribe* messaging system
- It was originally developed at LinkedIn and later became a part of Apache Project
- Kafka is fast, scalable, durable, fault-tolerant and distributed by design



edureka!

Copyright © 2017, edureka and/or its affiliates. All rights reserved.



Kafka @LinkedIn

- 1100+ commodity machines
- 31,000+ topics
- 350,000+ partitions

- 675 billion messages/day
- 150 TB/day in
- 580 TB/day out

- Peak Load**
- 10.5 million messages/sec
 - 18.5 GB/sec Inbound
 - 70.5 GB/sec Outbound

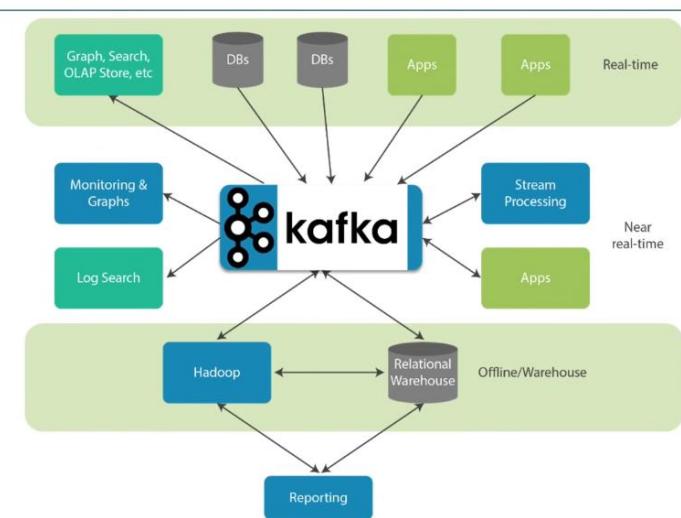


Fig: A modern stream-centric data architecture built around Kafka

edureka!

Copyright © 2017, edureka and/or its affiliates. All rights reserved.



Kafka Growth Exploding

- More than **1/3** of all Fortune **500** companies use **Kafka**.
- These companies includes the top ten travel companies, **7** of top ten banks, **8** of top ten insurance companies, **9** of top ten telecom companies.
- LinkedIn**, **Microsoft** and **Netflix** process billions of messages a day with Kafka (1,000,000,000,000).
- Kafka** is used for **real-time streams** of data & used to collect big data for **real time analysis**.



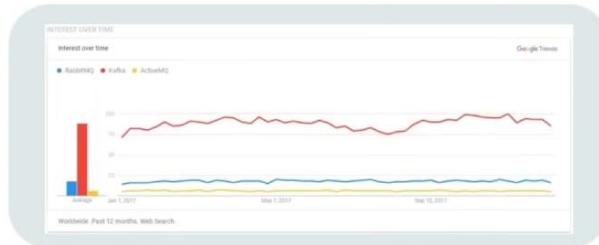
86% of respondents reported that the number of their systems that use Kafka is increasing



20% reported that the number is "growing a lot!"



52% of organizations have at least **6** systems running Kafka



Source: Google Trends

edureka!

Copyright © 2017, edureka and/or its affiliates. All rights reserved.



Kafka Terminologies

Producer
A **producer** can be any application who can publish messages to a topic

Consumer
A **consumer** can be any application that subscribes to a topic and consume the messages

Partition
Topics are broken up into ordered commit logs called **partitions**

Broker
Kafka cluster is a set of servers, each of which is called a **broker**

Topic
A **topic** is a category or feed name to which records are published

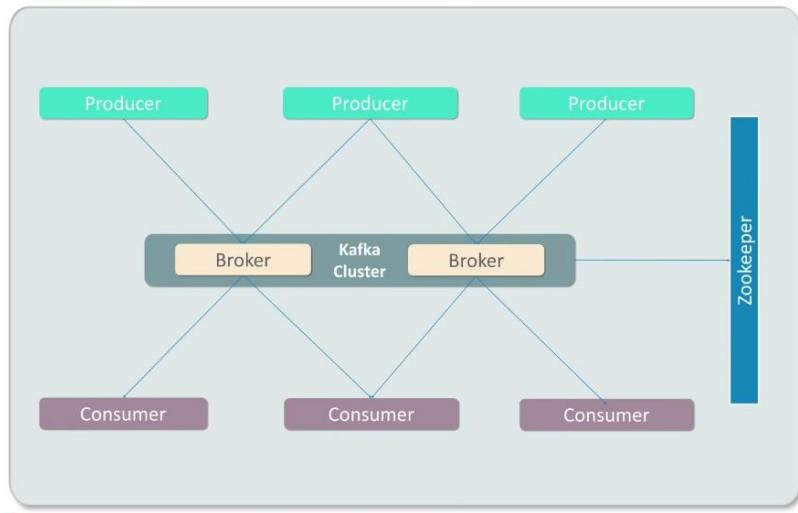
Zookeeper
ZooKeeper is used for managing and coordinating Kafka broker



Topic where – events are published and from where consumers can consume those events .

Topics can be further divided to partitions let suppose if its divided to 3 partitions , then 3 consumers can consume messages parllely .

Kafka Cluster



edureka!

Copyright © 2017, edureka and/or its affiliates. All rights reserved.



Zookeeper stores the metadata information of kafka cluster .

Ex : Broker info , kafka topics e.t.c

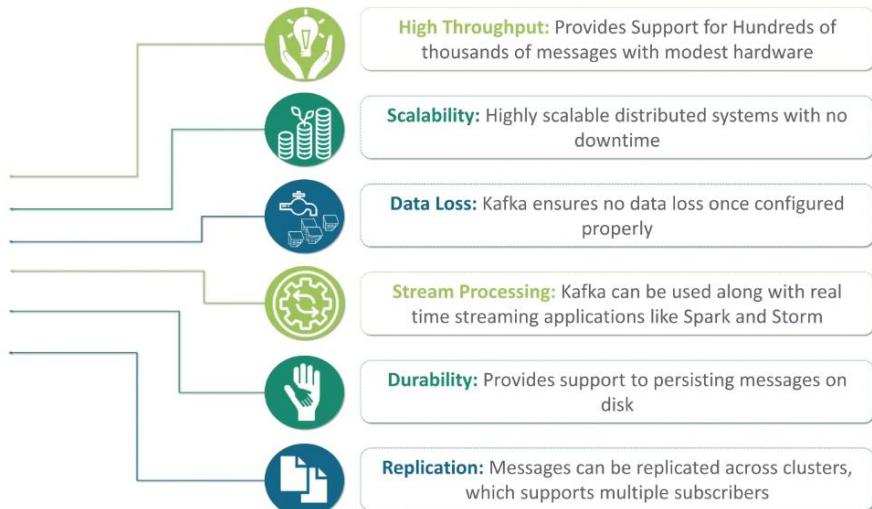
This kafka cluster is managed by Zookeeper

Kafka Features



kafka

edureka!



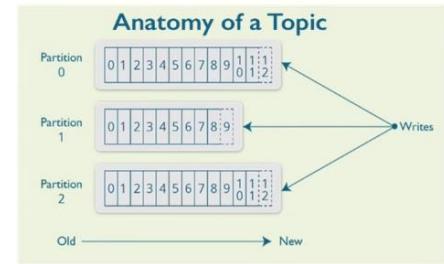
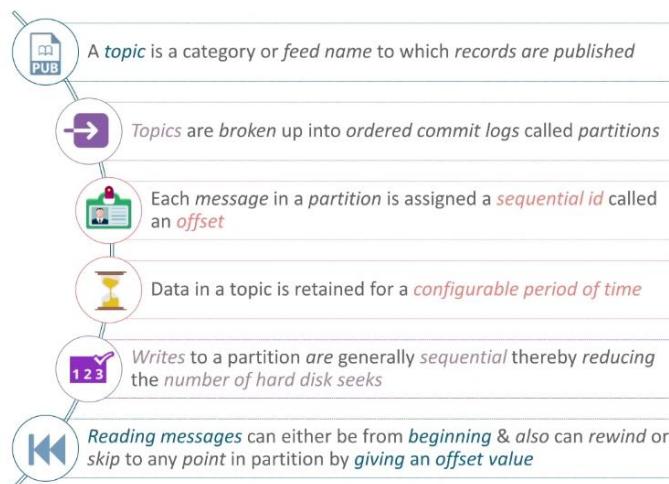
Copyright © 2017, edureka and/or its affiliates. All rights reserved.



Note : we can have a single consumer can consume the messaged from a single partition. Else we are not able to figure out the sequence of messages .

Durability – We can store that data for backup purpose as well it will not only in memory data.

Kafka Components - Topics and Partitions

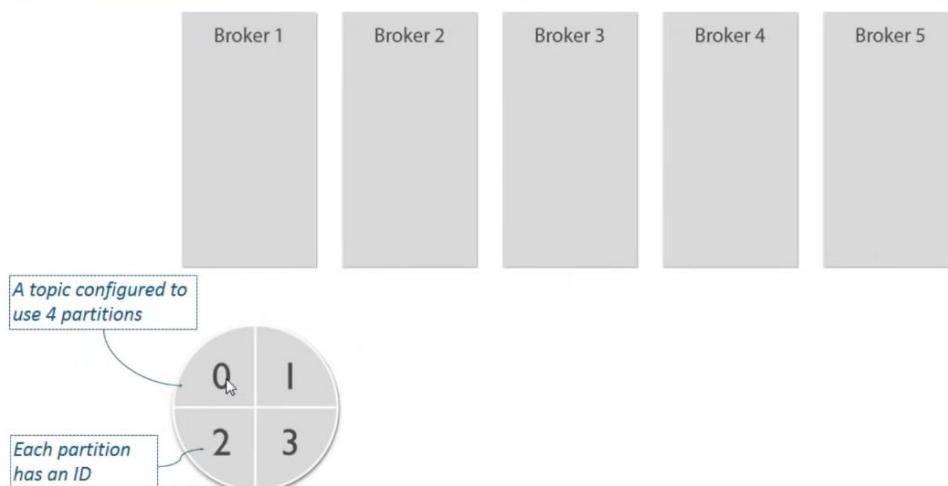


edureka!

Copyright © 2017, edureka and/or its affiliates. All rights reserved.



Kafka Components - Topics, Partitions & Replicas

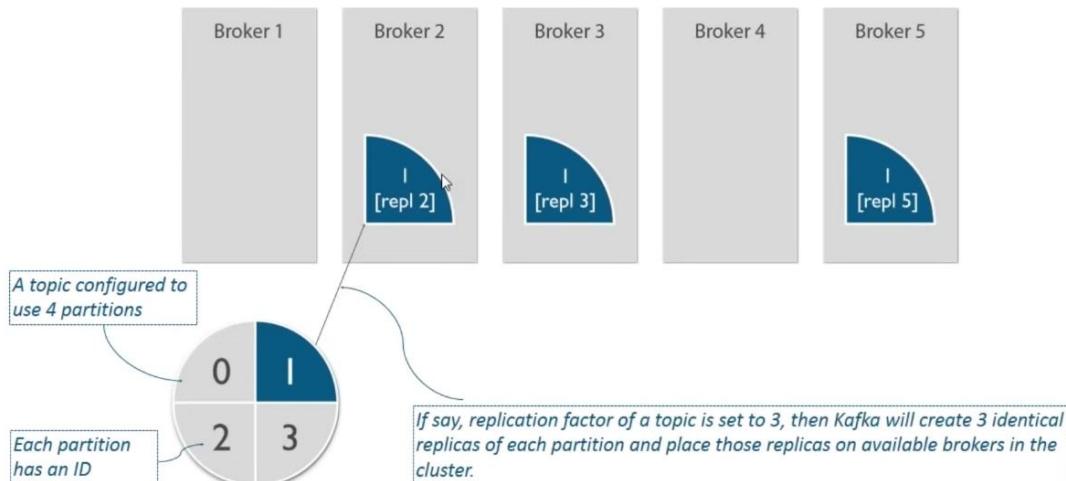


Copyright © 2017, edureka and/or its affiliates. All rights reserved.

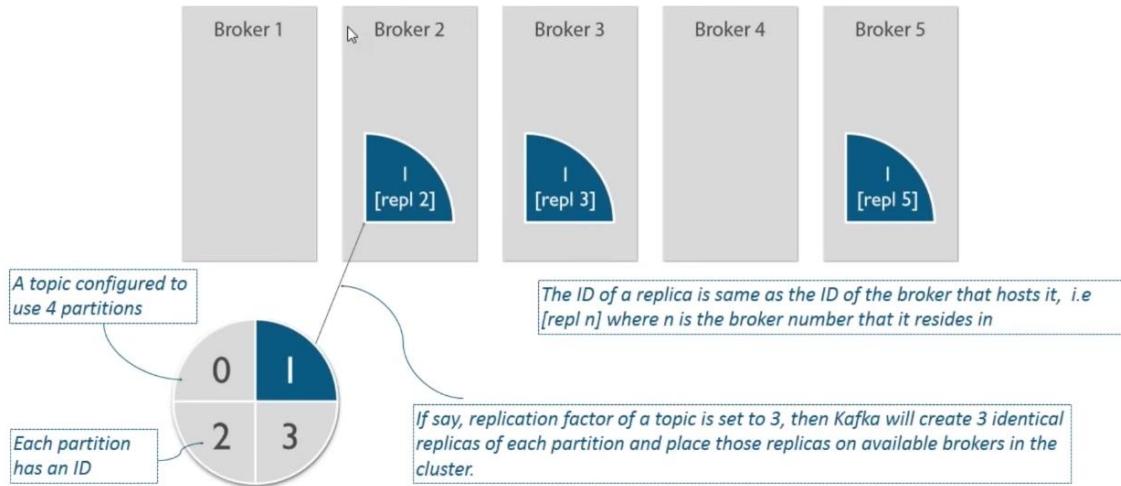


edureka!

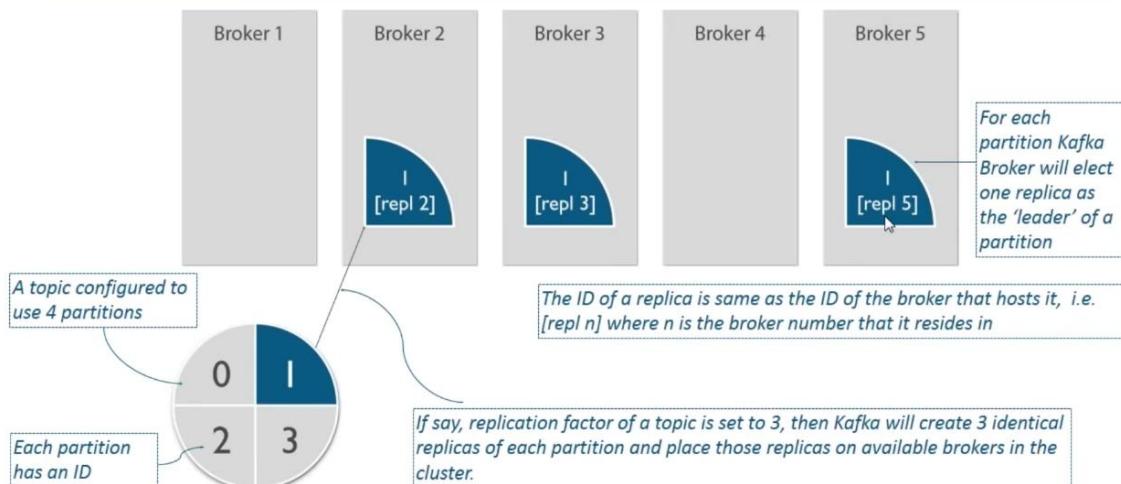
Kafka Components - Topics, Partitions & Replicas



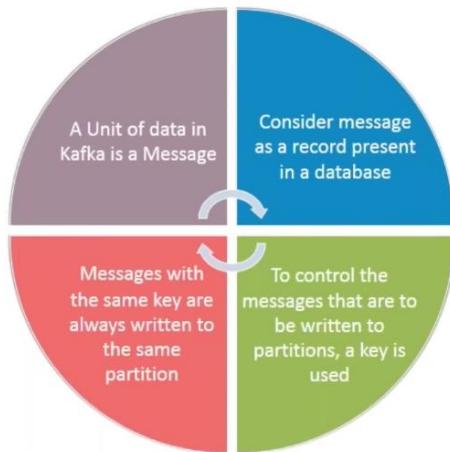
Kafka Components - Topics, Partitions & Replicas



Kafka Components - Topics, Partitions & Replicas

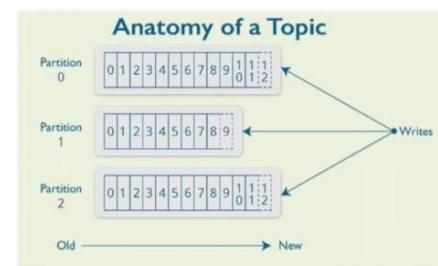


Kafka Components - Messages



Kafka Components - Topics and Partitions

- A *topic* is a category or *feed name* to which *records* are published
- Topics are broken up into ordered commit logs called *partitions*
- Each message in a partition is assigned a *sequential id* called an *offset*
- Data in a topic is retained for a *configurable period of time*
- Writes to a partition are generally *sequential* thereby reducing the number of hard disk seeks
- Reading messages can either be from *beginning* & also can *rewind* or skip to any point in partition by giving an *offset value*



Kafka Components - Producer

1

Producer (publisher or writer) publishes a new message to a specific topic

3

Directing messages to a partition is done using the message key and a partitioner, this will generate a hash of the key and map it to a partition

2

The producer does not care what partition a specific message is written to and will balance messages over every partition of a topic evenly

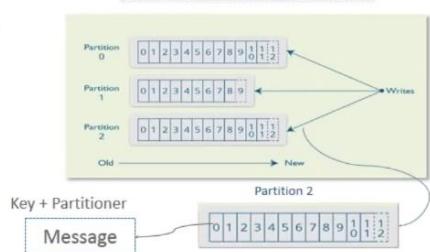
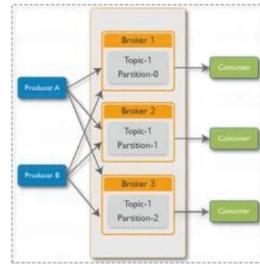
4

Every message a producer publishes in the form of a key : value pair

edureka!

SUBSCRIBE

Copyright © 2017, edureka and/or its affiliates. All rights reserved.



Kafka Components - Consumer

Consumers(subscribers or readers) read messages

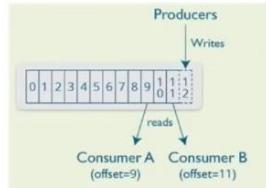
The consumer subscribes to one or more topics and reads the messages sequentially

The consumer keeps track of the messages it has consumed by keeping track on the offset of messages

The offset is bit of metadata(an integer value that continually increases)that Kafka adds to each message as it is produced

Each partition has a unique offset which is stored

With the offset of the last consumed message, a consumer can stop and restart without losing its current state



edureka!

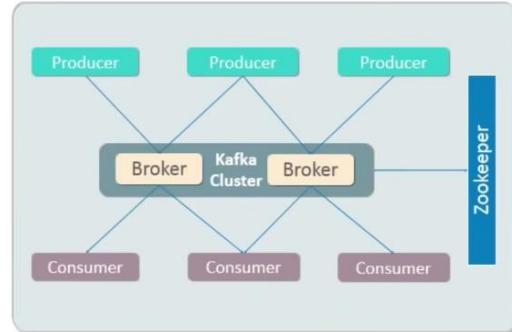
SUBSCRIBE

Copyright © 2017, edureka and/or its affiliates. All rights reserved.

Kafka Components - ZooKeeper

ZooKeeper is used for managing and coordinating Kafka broker

- Zookeeper service is mainly used for co-ordinating between brokers in the Kafka cluster
- Kafka cluster is connected to ZooKeeper to get information about any failure nodes



Zoopkeeper elects the leader from the various partition , if any leader got down , then it automatically remove it from its directory and elect a new leader .

Fault Tolerance



Deep Dive into Kafka Architecture

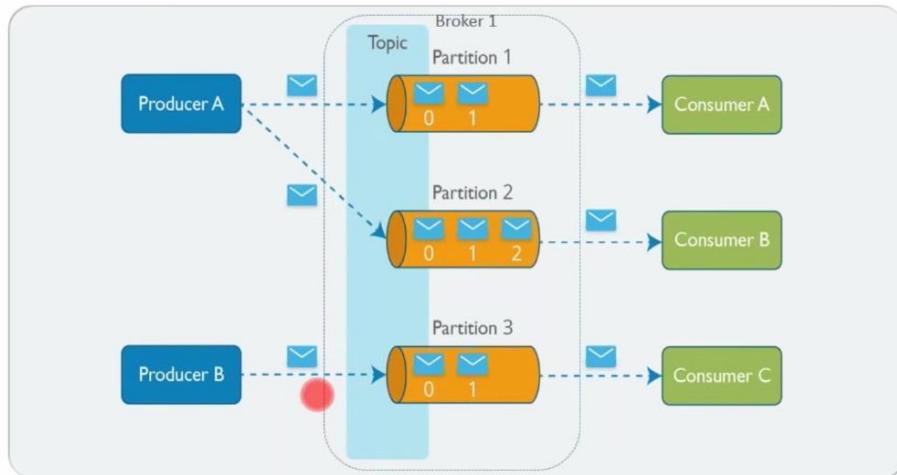
edureka!



Copyright © 2017, edureka and/or its affiliates. All rights reserved.



Kafka Architecture



edureka!

Copyright © 2017, edureka and/or its affiliates. All rights reserved.

SUBSCRIBE

Let's see some Use Cases of Kafka

edureka!

Copyright © 2017, edureka and/or its affiliates. All rights reserved.

SUBSCRIBE

Kafka - Use Cases



- Applications can produce messages using Kafka, without being concerned about the format of the messages
- Messages are sent and handled by a single application that can read all of them consistently, including :
 - A common formatting of messages using a common look
 - Send multiple messages in a single notification
 - Receive messages in a way that meets the users preferences

edureka!

Copyright © 2017, edureka and/or its affiliates. All rights reserved.

SUBSCRIBE

Kafka - Use Cases



- Originally Kafka was designed at LinkedIn, to track user activity
- When a user interacts with frontend applications, which generates messages regarding actions the user is taking
- Kafka keeps track of simple information like click tracking to complex information like data in a user's profile

edureka!

Copyright © 2017, edureka and/or its affiliates. All rights reserved.

SUBSCRIBE

Kafka - Use Cases



- Kafka is also ideal for collecting application's and system metrics and logs
- Applications publish metrics on a regular basis to a Kafka topic, and those metrics can be consumed by systems for monitoring and alerting
- Log messages can be published in the same way and routed to dedicated log search systems like Elasticsearch or security analysis applications

edureka!

Copyright © 2017, edureka and/or its affiliates. All rights reserved.



Kafka - Use Cases



- Database changes can be published to Kafka and applications can easily monitor this stream to receive live updates as they happen
- Kafka replicates database updates to a remote system for consolidating changes from multiple applications in a single database view
- Durable retention becomes useful providing a buffer for the changelog, meaning it can be replayed in the event of a failure of the consuming applications
- Log-compacted topics can be used to provide longer retention by only retaining a single change per key

edureka!

Copyright © 2017, edureka and/or its affiliates. All rights reserved.



Kafka - Use Cases



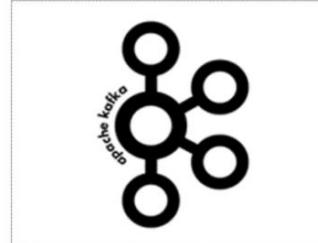
- Stream processing term is typically used to refer applications that provide similar functionality to map/reduce processing in Hadoop
- Stream processing operates on data in real-time, as quickly as messages are produced :
 - Write small applications to operate on Kafka messages,
 - Performing tasks such as counting metrics
 - Partitioning messages for efficient processing by other applications

Getting Started with Kafka

- Prerequisites :



- Components :

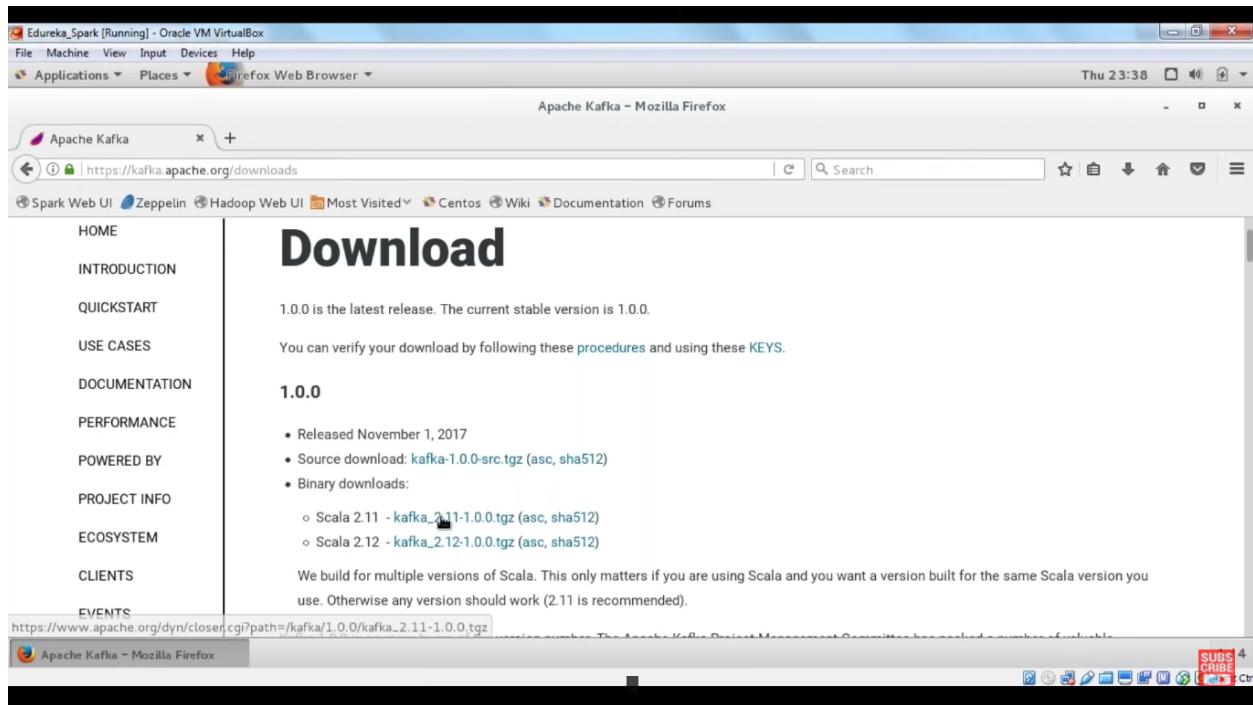


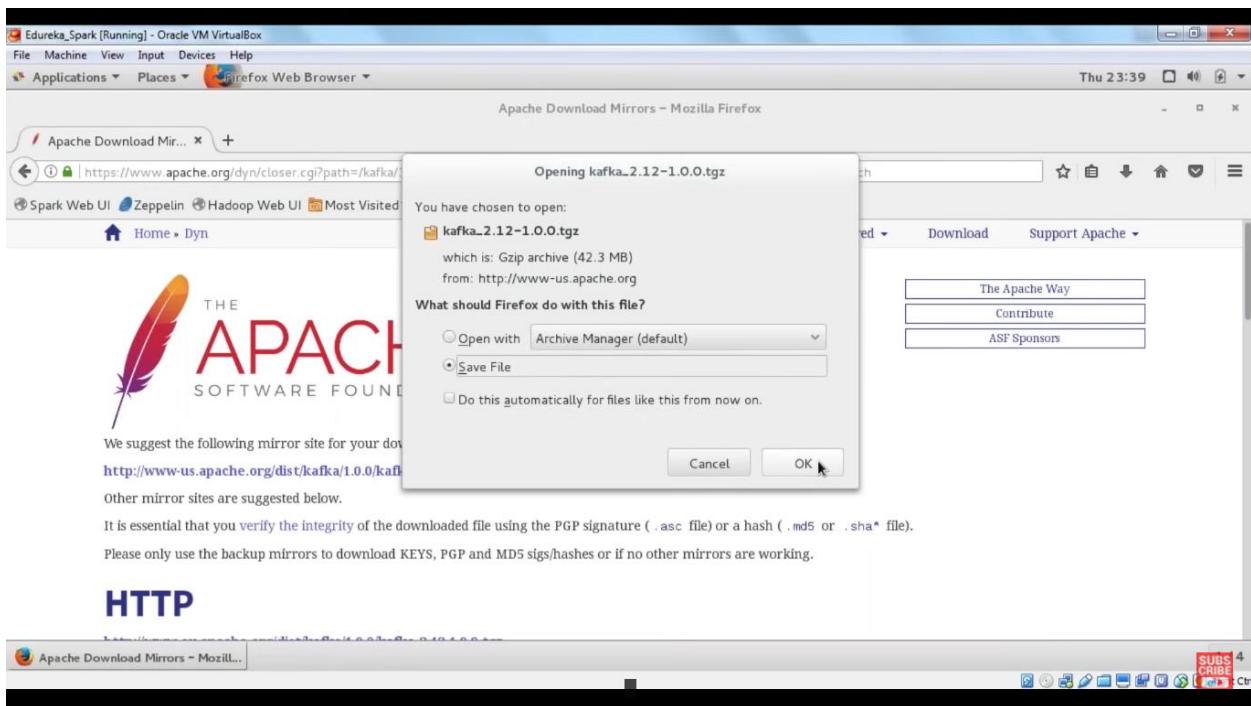
edureka!

Copyright © 2017, edureka and/or its affiliates. All rights reserved.

SUBSCRIBE

The screenshot shows a Firefox browser window with the title bar "Edureka_Spark [Running] - Oracle VM VirtualBox" and the status bar "Thu 23:37". The address bar shows the URL "https://www.google.co.in/search?q=download+apache+Kafka&ie=utf-8&oe=utf-8&client=firefox-b-ab&gws_rc". The search query "download apache Kafka" is entered in the search bar. Below the search bar, there are navigation buttons for "Connecting...", a refresh button, and a plus sign. The main content area displays search results for "download apache Kafka" on Google. The first result is a link to the Apache Software Foundation's Kafka download page: "Download - Apache Kafka - The Apache Software Foundation!". The second result is a link to the Kafka quickstart guide: "Quickstart - Apache Kafka - The Apache Software Foundation!". At the bottom of the browser window, there is a toolbar with various icons and a "SUBSCRIBE" button.





HTTP

Note : uncomment the port line

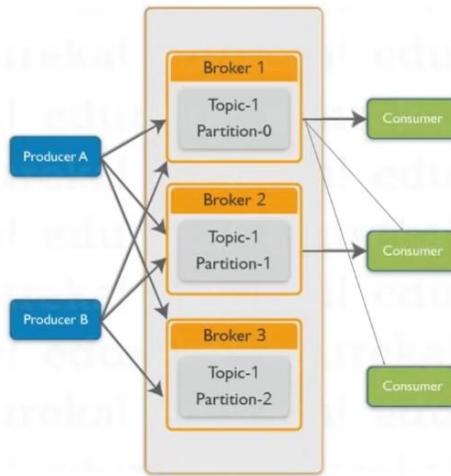
```

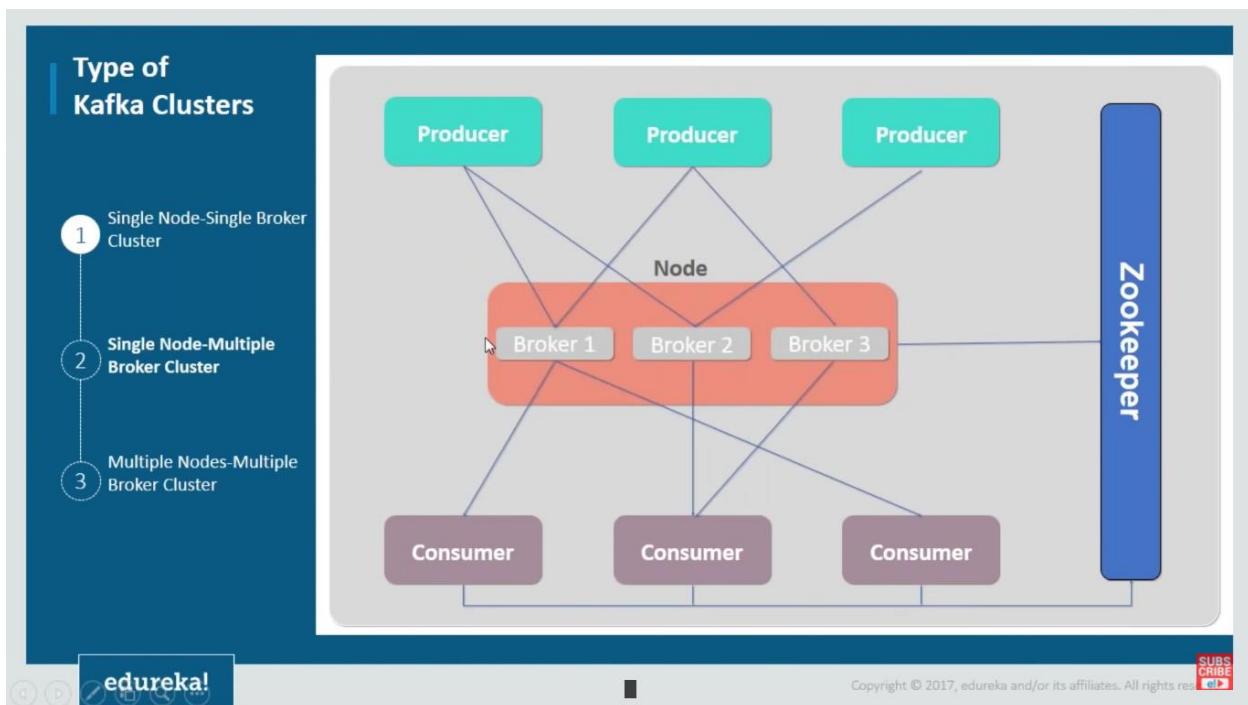
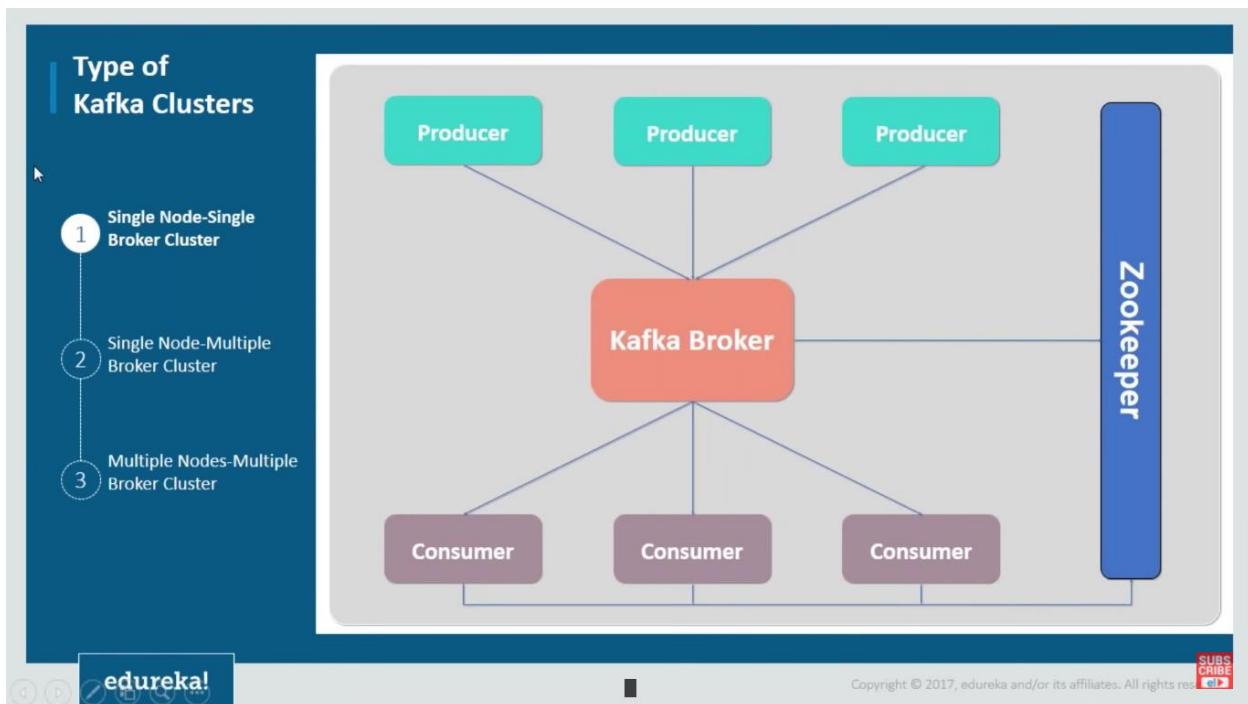
[edureka@localhost kafka_2.12-0.11.0.0]$ ./bin/kafka-topics.sh --describe --zookeeper localhost:2181 --topic test-edureka
Topic:test-edureka PartitionCount:1 ReplicationFactor:1 Configs:
Topic: test-edureka Partition: 0 Leader: 0 Replicas: 0 Isr: 0
[edureka@localhost kafka_2.12-0.11.0.0]$

```

Kafka Cluster

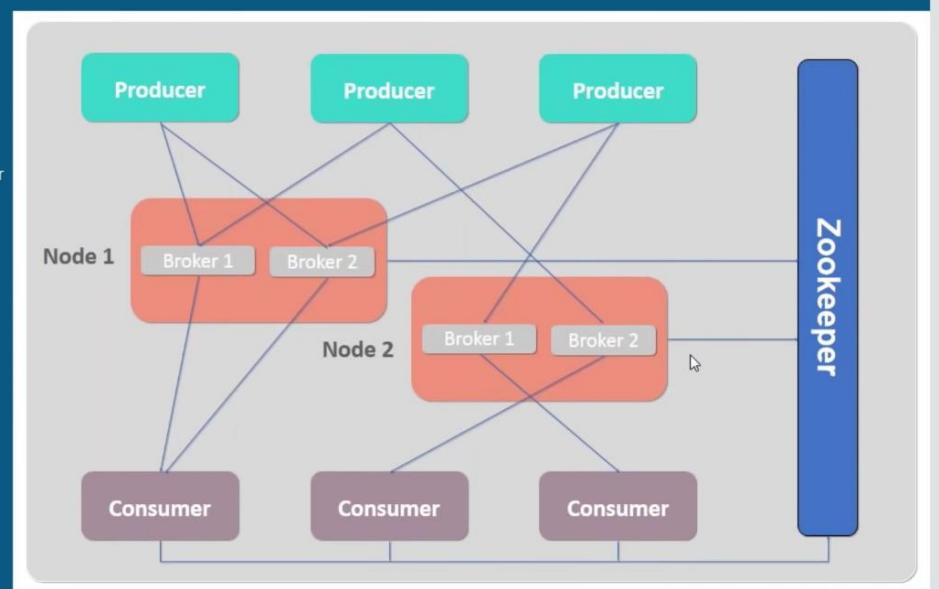
- Kafka brokers are designed to operate as part of a cluster
- One broker will also function as the cluster controller
- Controller is responsible for administrative operations, like
 - Assigning partitions to brokers
 - Monitoring for broker failures in a cluster
- A particular partition is owned by a broker, and that broker is called the leader of the partition
- All consumers and producers operating on that partition must connect to the leader





Type of Kafka Clusters

- 1 Single Node-Single Broker Cluster
- 2 Single Node-Multiple Broker Cluster
- 3 Multiple Nodes-Multiple Broker Cluster





Let's configure a Single Node-Single Broker Cluster

Single Node Single Broker Cluster

Step 1 : Start the Zookeeper Server :

Command: `bin/zkServer.sh start`

```
[edureka@localhost zookeeper-3.4.10]$ bin/zkServer.sh start
ZooKeeper JMX enabled by default
Using config: /lib/zookeeper-3.4.10/bin/../conf/zoo.cfg
Starting zookeeper ... STARTED
```

Step 2 : Start the Kafka Server

Command: `bin/kafka-server-start.sh config/server.properties`

```
[edureka@localhost kafka_2.11-0.11.0.1]$ bin/kafka-server-start.sh config/server.properties
```

Single Node Single Broker Cluster

Step 3 : Create a Topic with the name “topic-edu”

```
Command: bin/kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1 --topic topic-edu
```

```
[edureka@localhost kafka_2.11-0.11.0.1]$ bin/kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1 --topic topic-edu
Created topic "topic-edu".
```

Step 4 : Start a Producer

```
Command: bin/kafka-console-producer.sh --broker-list localhost:9092 --topic topic-edu
```

```
[edureka@localhost kafka_2.11-0.11.0.1]$ bin/kafka-console-producer.sh --broker-list localhost:9092 --topic topic-edu
```



Copyright © 2017, edureka and/or its affiliates. All rights reserved.



Single Node Single Broker Cluster

Step 5 : Start a Consumer

```
Command: bin/kafka-console-consumer.sh --zookeeper localhost:2181 --topic topic-name --from-beginning
```

```
[edureka@localhost kafka_2.11-0.11.0.1]$ bin/kafka-console-consumer.sh --zookeeper localhost:2181 --topic topic-edu --from-beginning
```

Step 6 : Producer can publish a message to a topic which will be received by the consumer which has subscribed to that topic

Producer

```
[edureka@localhost kafka_2.11-0.11.0.1]$ bin/kafka-console-producer.sh --broker-list localhost:9092 --topic topic-edu  
>My name is parth  
>Hello parth  
>how are you  
>I m gud
```

Consumer

```
[edureka@localhost kafka_2.11-0.11.0.1]$ bin/kafka-console-consumer.sh --zookeeper localhost:2181 --topic topic-edu --from-beginning  
Using the ConsoleConsumer with old consumer is deprecated and will be removed in a future major release. Consider using the new consumer by passing  
--new-consumer instead of [zookeeper].  
My name is parth  
Hello parth  
how are you  
I m gud
```

edureka!

Copyright © 2017, edureka and/or its affiliates. All rights reserved.



Kafka Course Outline



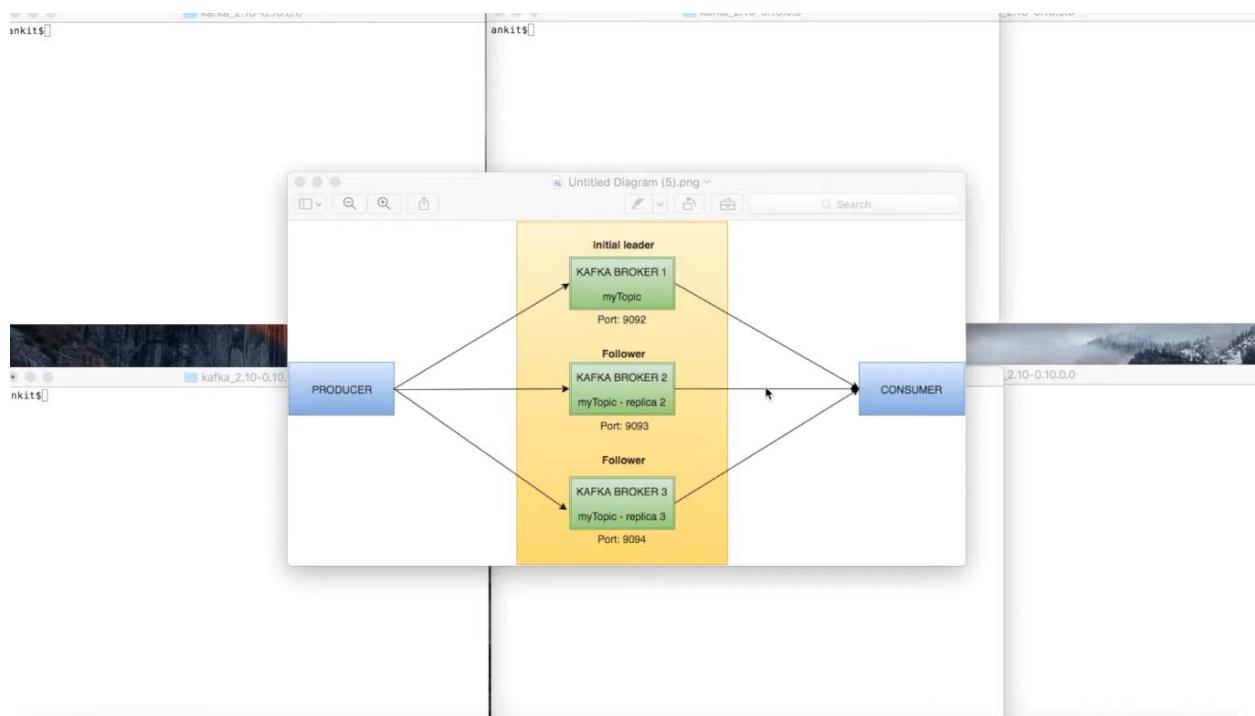
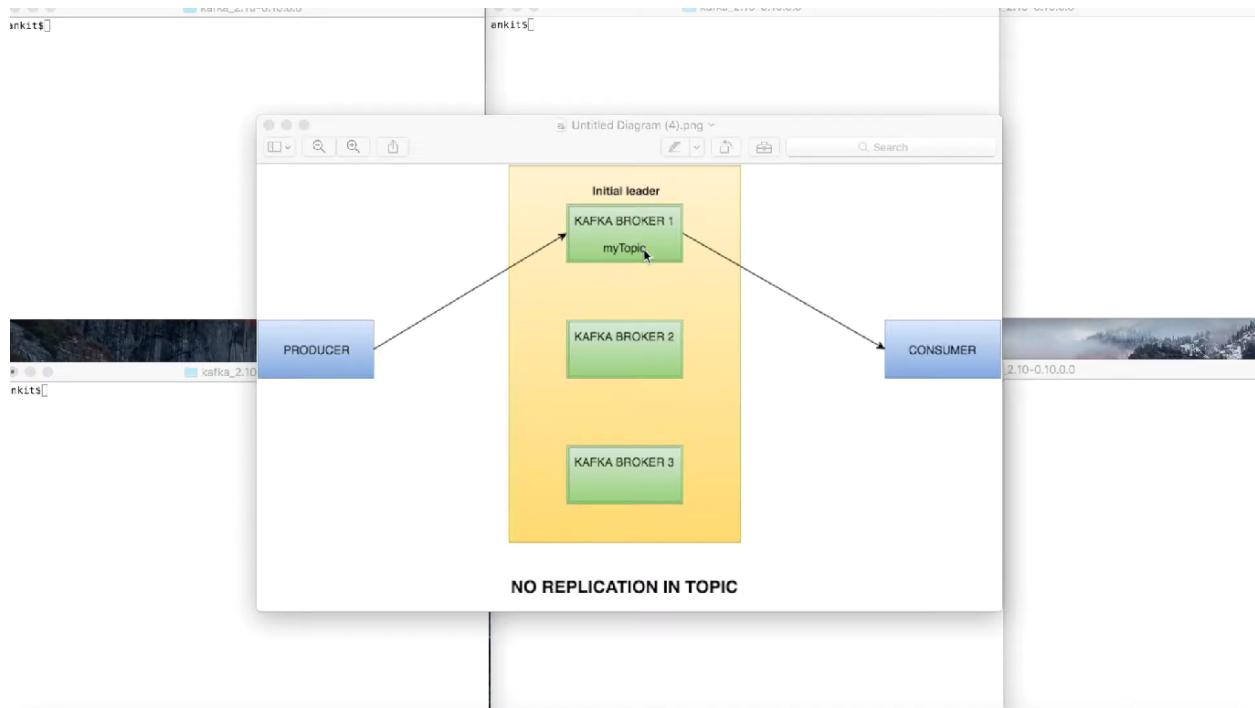
edureka!

Copyright © 2017, edureka and/or its affiliates. All rights reserved.



Follow file for demo
(Includes all commands)

Second Demo :

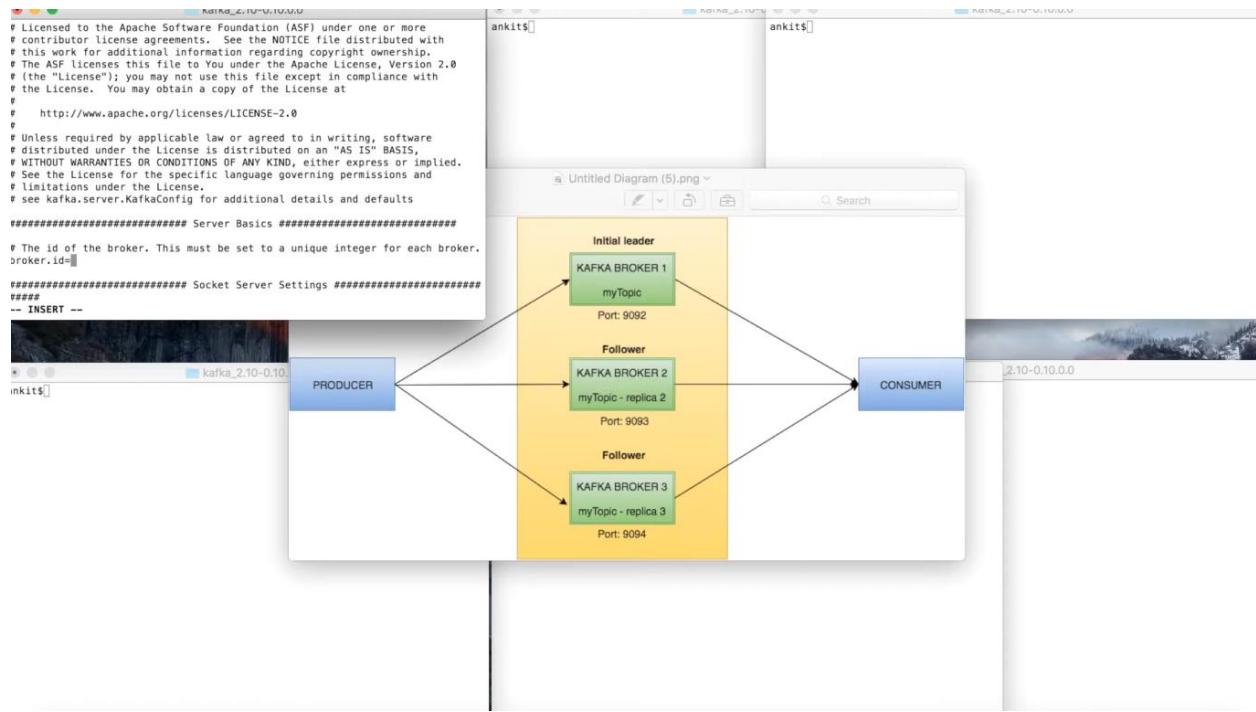


To make it as fault tolerant we need to replicate it to multiple broker (meaning that multiple replica of same topic)

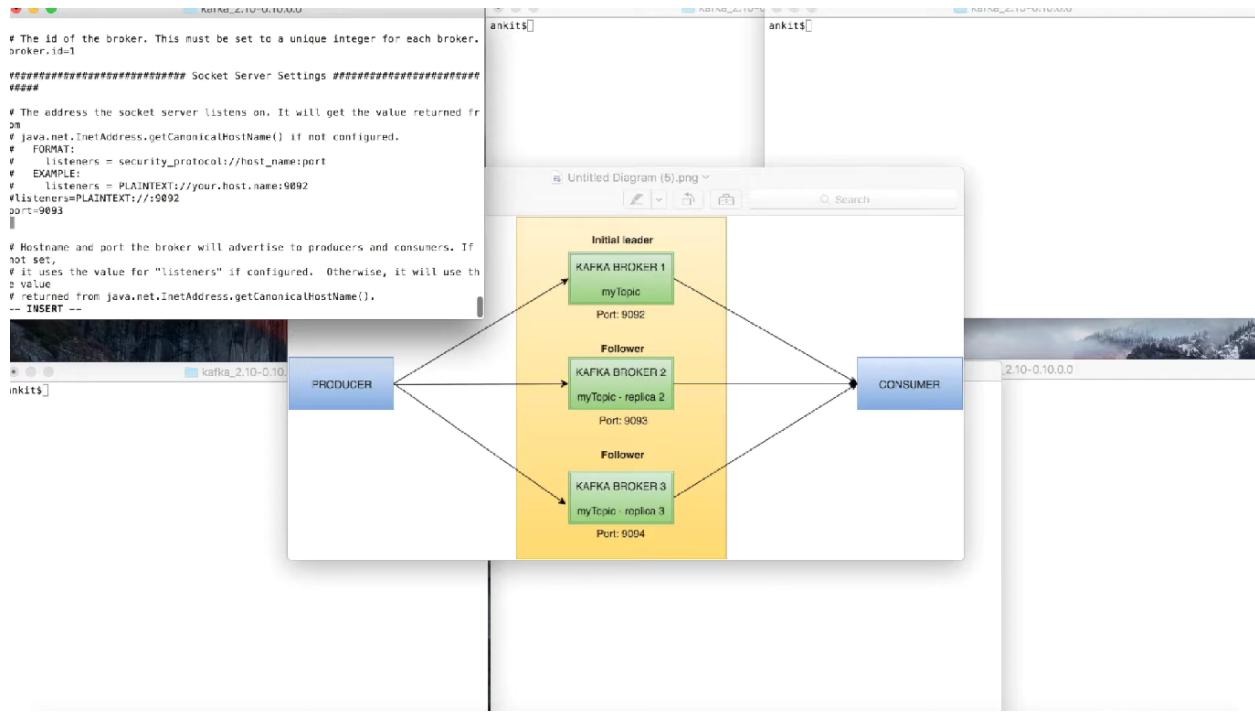


Create 2 more copied of servers.property files

Edit the port to make it diff as we running this demo in same machine

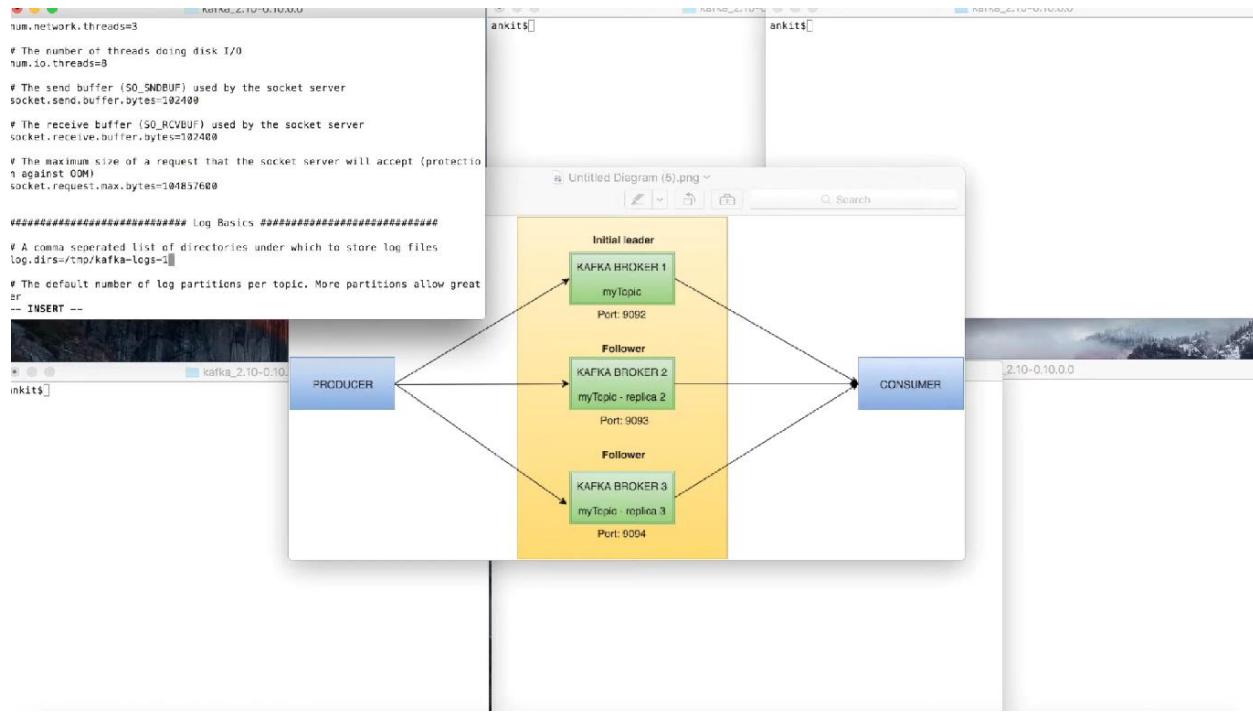


Change the broker id first = 1 and 2 simultaneously



Change the port = 9093

And comment that line which we
uncommented in single node architecture



Change the log directory as well to logs-1 for this broker

Do this for third one also all three steps

Then up zookeeper and all three broker with diff server.properties

And then create topic with replication factor as 3

The diagram illustrates a Kafka cluster architecture. It features three yellow rectangular boxes representing Kafka BROKERS, each containing a green box labeled 'myTopic'. Broker 1 is labeled 'Initial leader'. Broker 2 is labeled 'Follower' and 'myTopic - replica 2'. Broker 3 is also labeled 'Follower' and 'myTopic - replica 3'. A blue box labeled 'PRODUCER' has arrows pointing to both Broker 2 and Broker 3. A blue box labeled 'CONSUMER' has an arrow pointing from Broker 3. The background shows a landscape scene with mountains and water.

After all steps start producer now this time give all three broker while defining broker-list

And start consumer everything working fine

1

Now to check fault taurernece kill one of the
broker and see

The screenshot shows a terminal window with several tabs open, displaying log output from Kafka components. The tabs include:

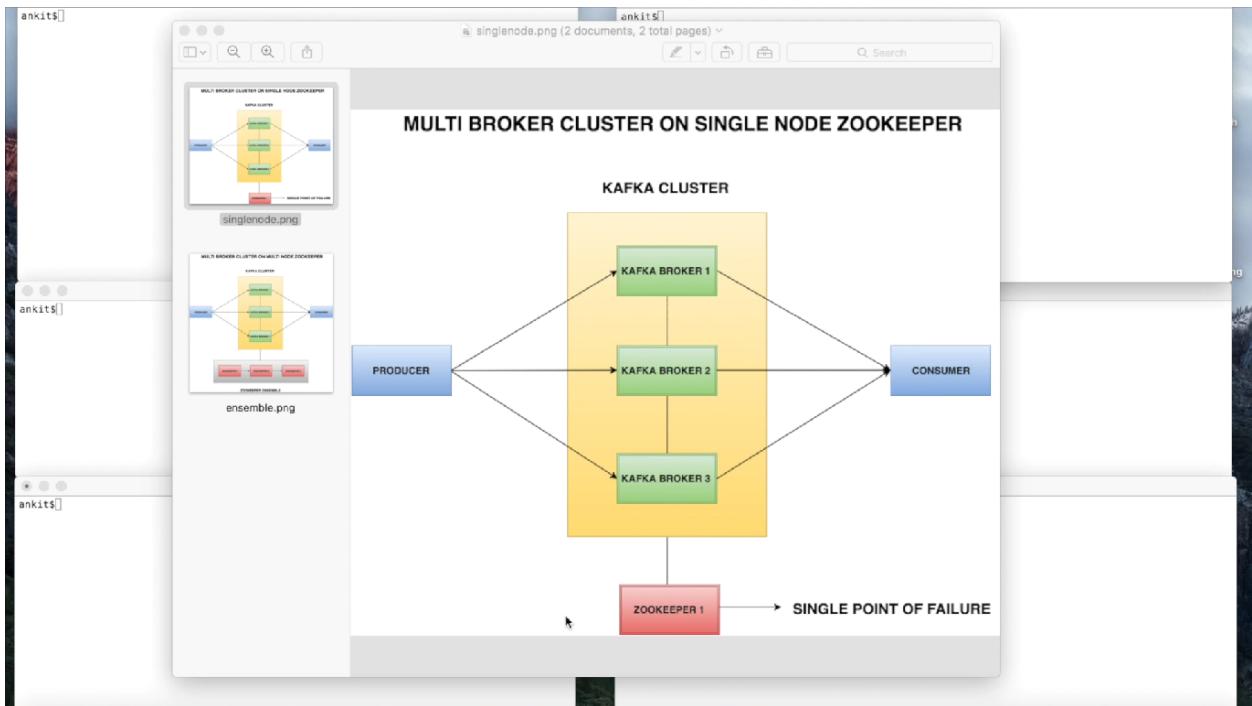
- `kafka_2.10.0.10.0.0`: Shows logs for a Kafka server instance.
- `zookeeper`: Shows logs for ZooKeeper servers.
- `follower`: Shows logs for a Kafka follower instance.
- `leader`: Shows logs for a Kafka leader instance.
- `replicaFetcher`: Shows logs for a ReplicaFetcher thread.
- `replicaFetcherManager`: Shows logs for a ReplicaFetcherManager.
- `replicaFetcherManager`: Shows logs for another ReplicaFetcherManager instance.
- `replicaFetcherManager`: Shows logs for a third ReplicaFetcherManager instance.

The logs contain detailed information about socket connections, session establishment, partition metadata, and consumer errors. Key log entries include:

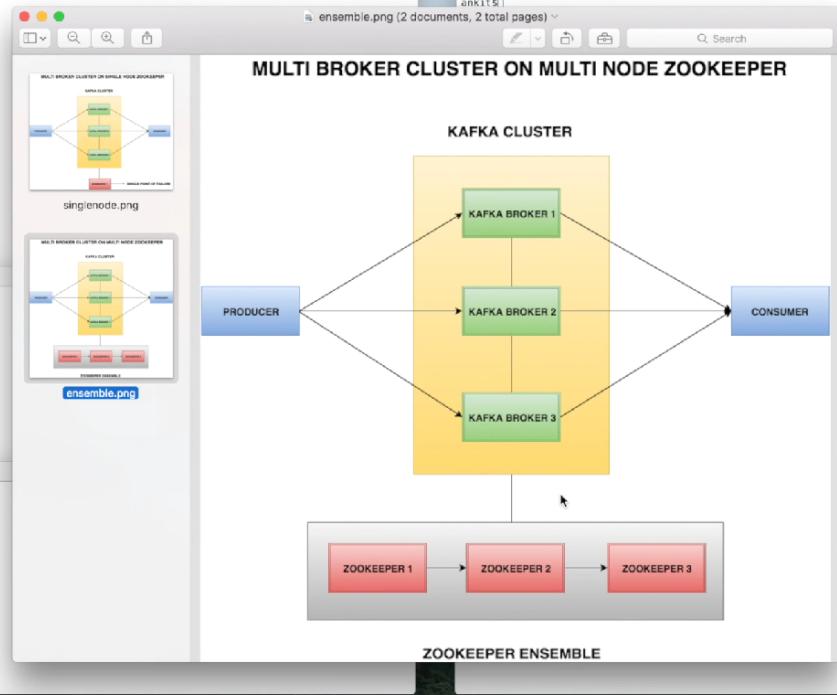
- ZooKeeper logs: Accepting socket connections, establishing sessions, and handling requests for replication topics.
- Kafka logs: Partition assignment, leader election, and consumer errors related to the `replicationtopic`.
- ReplicaFetcher logs: Fetching offsets and committing them to the log.

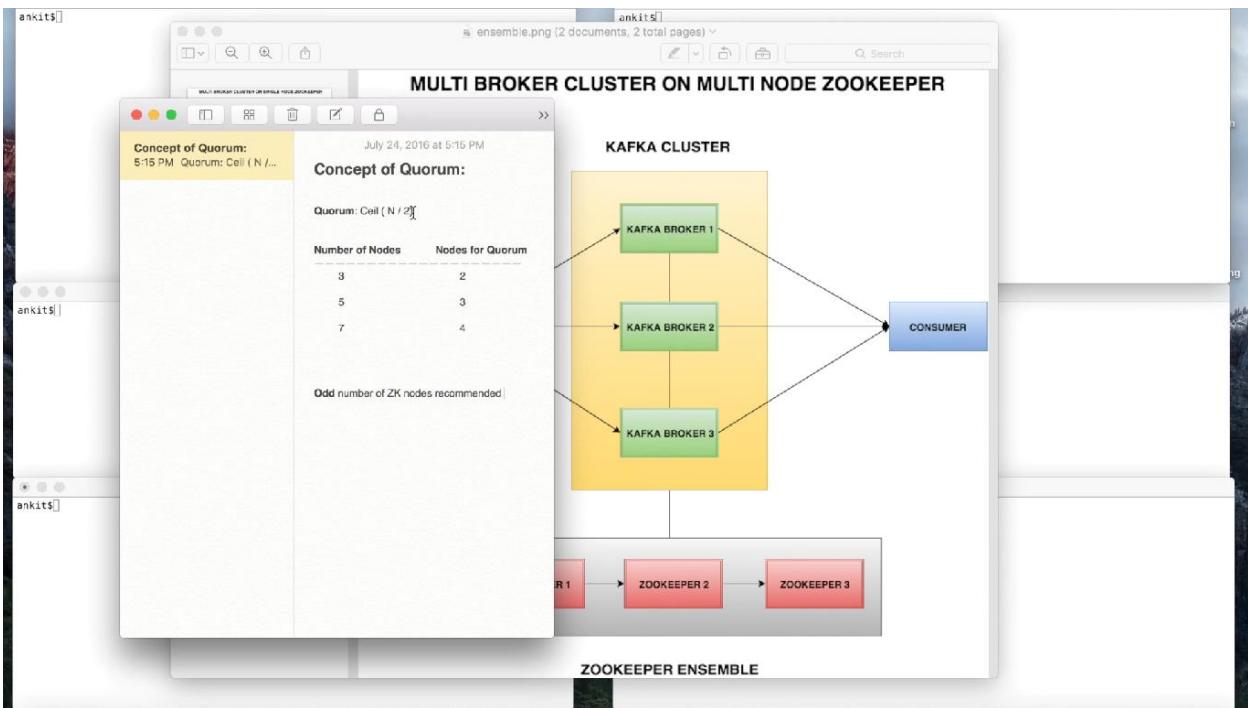
Still u able to work .

DEMO -3



Multiple broker with multiple node
zookeeper server





Note : Ideally in zoopkeeper documentation , we should keep odd no of brokers , this will help zookeeper to decide the leader better way instead of even no of broker (initially) or we can say odd one is more fault taulernet

Similarly odd no of zookeeper instances

The screenshot shows a Windows desktop environment. In the center, there are two Notepad++ windows side-by-side. Both windows have the title bar "C:\kafka_2.12-2.8.1\config\zookeeper.properties - Notepad++". The left window contains configuration settings for Kafka servers, while the right window contains the Apache Zookeeper license file. Below the windows, the Windows taskbar is visible, featuring the Start button, a search bar with the placeholder "Type here to search", and several pinned icons for applications like File Explorer, Edge, and File History.

```

# See the License for the specific language governing permissions and
# limitations under the License.
# the directory where the snapshot is stored.
dataDir=/tmp/kafkatutorial/zookeeper
# the port at which the clients will connect
clientPort=2181
# disable the per-ip limit on the number of connections since this is a non-production config
# tickTime=2000
initLimit=5
syncLimit=2

server.1=localhost:2666:3666
server.2=localhost:2667:3667
server.3=localhost:2668:3668

maxClientCnxs=0
-- INSERT --

```

```

1 # Licensed to the Apache Software Foundation (ASF) under one or more
2 # contributor license agreements. See the NOTICE file distributed with
3 # this work for additional information regarding copyright ownership.
4 # The ASF licenses this file to You under the Apache License, Version 2.0
5 # (the "License"); you may not use this file except in compliance with
6 # the License. You may obtain a copy of the License at
7 #
8 #     http://www.apache.org/licenses/LICENSE-2.0
9 #
10 # Unless required by applicable law or agreed to in writing, software
11 # distributed under the License is distributed on an "AS IS" BASIS,
12 # WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
13 # See the License for the specific language governing permissions and
14 # limitations under the License.
15 # the directory where the snapshot is stored.
16 dataDir=/tmp/zookeeper
17 # the port at which the clients will connect
18 clientPort=2181
19 # disable the per-ip limit on the number of connections since this is a
20 # non-production config
21 tickTime=2000
22 initLimit=5
23 syncLimit=2
24
25 server.1=localhost:2666:3666
26 server.2=localhost:2667:3667
27 server.3=localhost:2668:3668
28 maxClientCnxs=0
29 # Disable the adminserver by default to avoid port conflicts.
30 # Set the port to something non-conflicting if choosing to enable this
31 admin.enableServer=false
32 # admin.serverPort=8080
33

```

Edit zookeeper.properties : (Add below)

tickTime=2000

initLimit=5

syncLimit=2

server.1=localhost:2666:3666

server.2=localhost:2667:3667

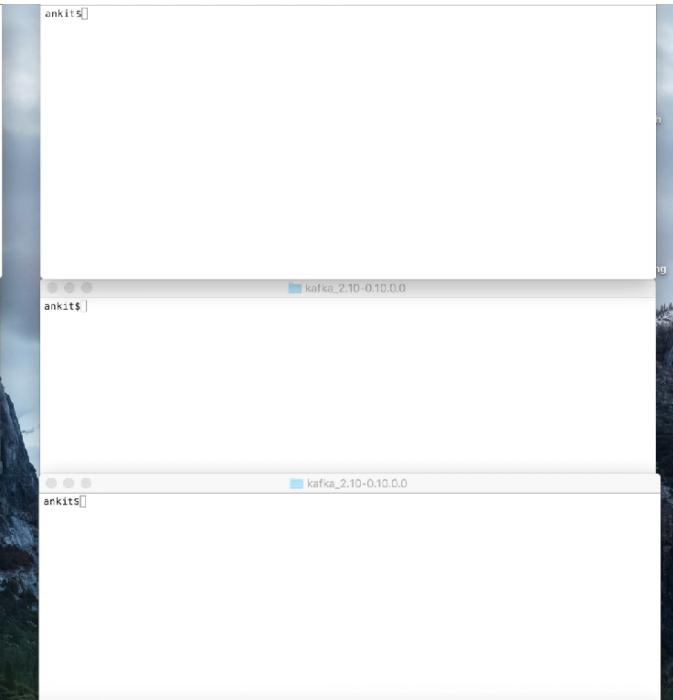
server.3=localhost:2668:3668

```
ankit$ vi config/zookeeper.properties
ankit$ vi config/zookeeper.properties
ankit$ cp config/zookeeper.properties config/zookeeper1.properties
ankit$ cp config/zookeeper.properties config/zookeeper2.properties
ankit$ vi config/
```

Make 2 more copied of zookeeper.prop file

```
# contributor license agreements. See the NOTICE file distributed with
# this work for additional information regarding copyright ownership.
# The ASF licenses this file to You under the Apache License, Version 2.0
# (the "License"); you may not use this file except in compliance with
# the License. You may obtain a copy of the License at
#
#     http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
# the directory where the snapshot is stored.
dataDir=/tmp/kafkaturorial/zookeeper
# the port at which the clients will connect
clientPort=2081
# disable the per-ip limit on the number of connections since this is a non-production config
# tickTime=2000
-- INSERT --
```

```
ankits$
```



Change the port from 2081 to 2082

Also change the data log directory

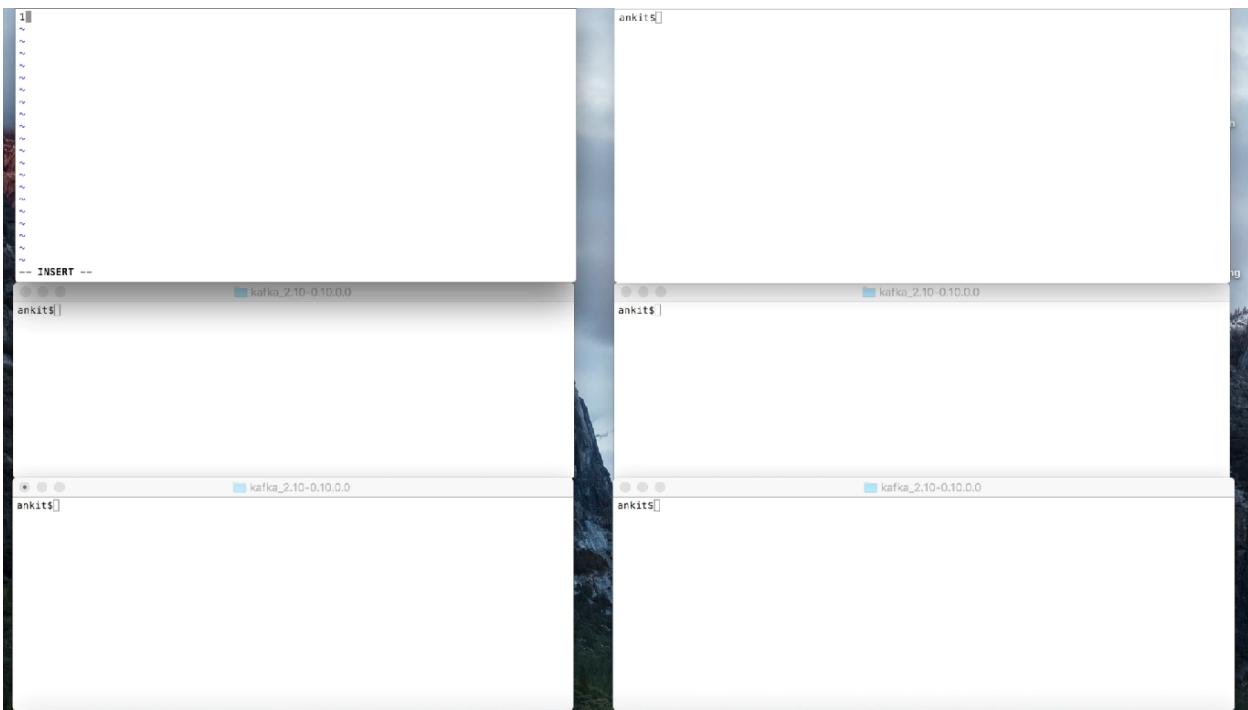
from

dataDir=/tmp/zookeeper

to

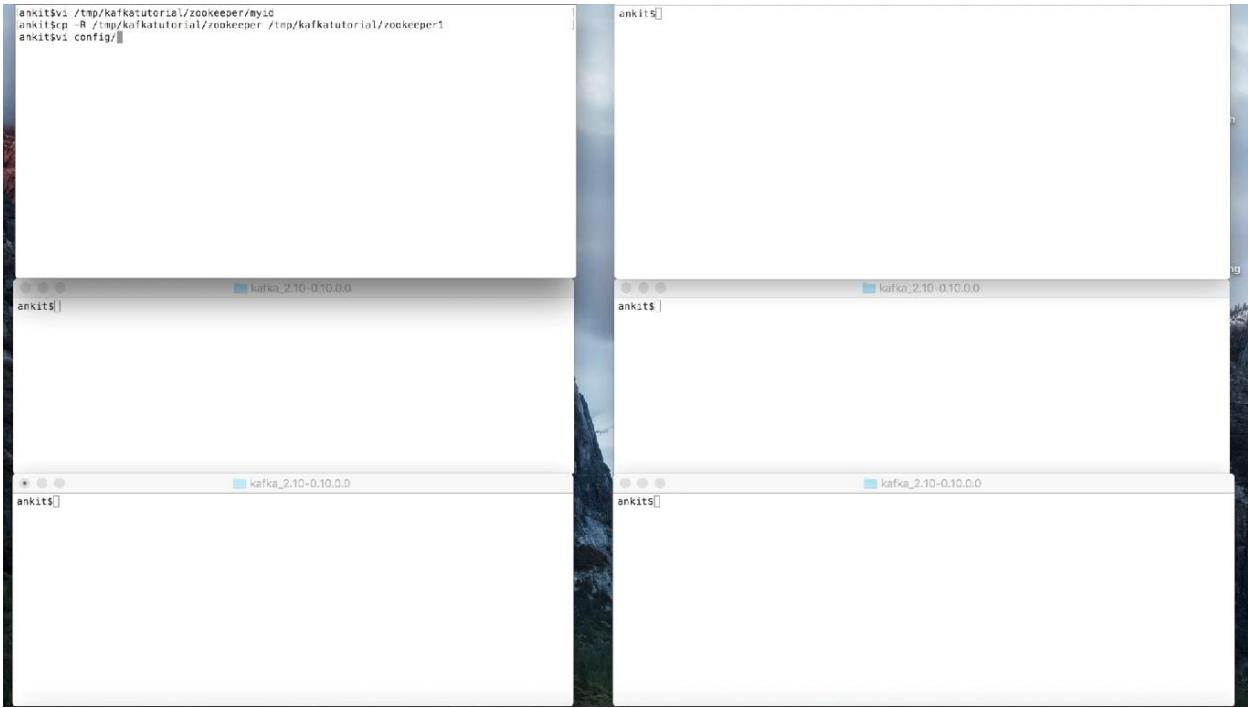
dataDir=/tmp/zookeeper1

Similarly do changes for 3 file 2083 ,
zookeper2



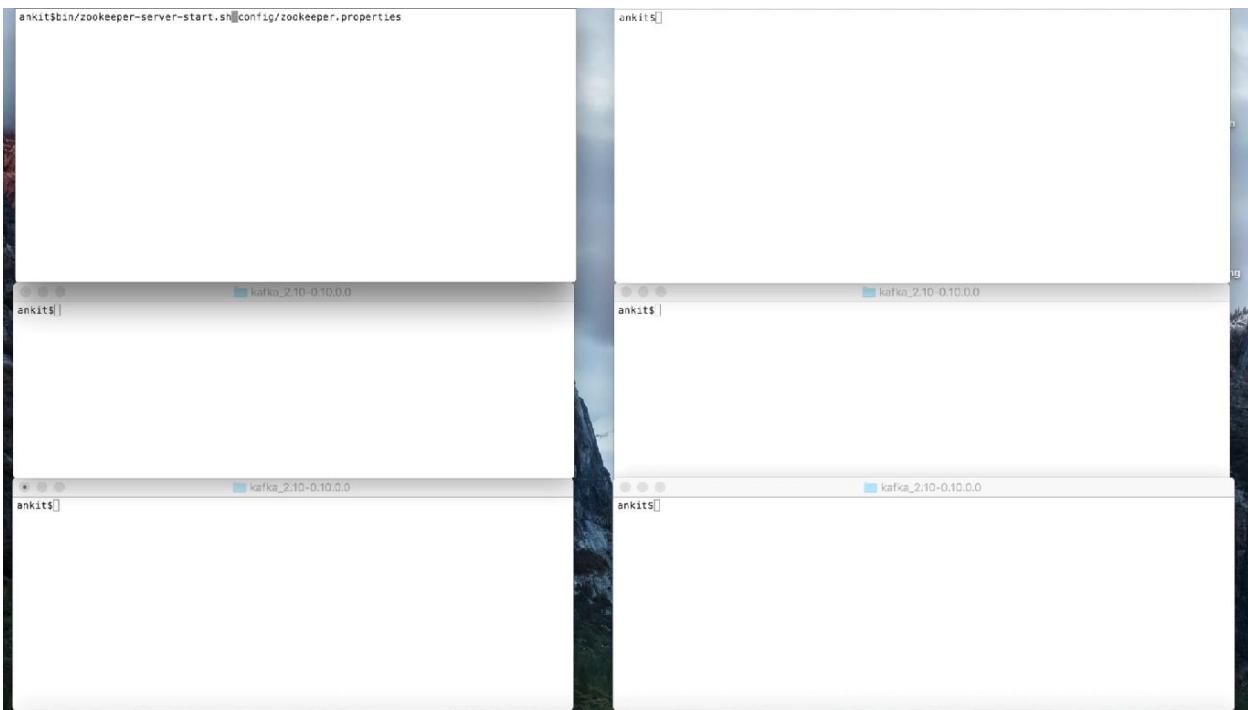
Now create file with name myid and enter 1
in it in existing zookeeper logs directory

Then copy the directory and paste it with
diff name (what name u mentioned in
zookeeper1.properties)



After copy edit the id changes to 2 in myid file .

Do for third directory as well same steps



Start all 3 zookeeper one by one

When u first start , in first one u might see some error because it tries to find its other mates , but as soon as iu start others those errors u cant see again

```
[2015-07-24 17:59:22,672] INFO Server environment:user.home=/Users/anrajput (org.apache.zookeeper.server.ZooKeeperServer)
[2015-07-24 17:59:22,673] INFO Server environment:user.dir=/Users/anrajput/Downloads/kafkatutorial/kafka_2.10-0.18.0.4 (org.apache.zookeeper.ZooKeeperServer)
[2015-07-24 17:59:22,674] INFO Created server with tickTime 2000 minSessionTimeout 4000 maxSessionTimeout 40000 datadir /tmp/kafkaTutorial/zookeeper/version-2 snapdir /tmp/kafkaTutorial/zookeeper/version-2 (org.apache.zookeeper.server.ZooKeeperServer)
[2015-07-24 17:59:22,675] INFO FOLLOWING - LEADER ELECTION TOOK - 34197 (org.apache.zookeeper.server.ZooKeeperServer)
[2015-07-24 17:59:22,692] INFO Getting a diff from the leader 0x400000001 (org.apache.zookeeper.server.QueuedLearner)
[2015-07-24 17:59:22,696] INFO Snappshutting: 0x400000001 to /tmp/kafkaTutorial/zookeeper/versien-2/snapshot.400000001 (org.apache.zookeeper.server.persistence.FileTxnSnapshotLog)
[2015-07-24 17:59:23,001] INFO [0x400000001]: Accepted socket connection from 72.17.6.1:16238 (org.apache.zookeeper.server.TCPServer)
[2015-07-24 17:59:23,004] INFO Client attempting to renew session 0x1561f8ccf430000 at /127.0.0.1:62585 (org.apache.zookeeper.server.ZooKeeperServer)
[2015-07-24 17:59:23,004] INFO Revalidating client: 0x1561f8ccf430000 (org.apache.zookeeper.server.QueuedLearner)
[2015-07-24 17:59:23,009] INFO Established session 0x1561f8ccf430000 with negotiated timeout 30000 for client /127.0.0.1:62585 (org.apache.zookeeper.server.ZooKeeperServer)
[2015-07-24 17:59:22,673] INFO LEADING - LEADER ELECTION TOOK - 226 (org.apache.zookeeper.server.Quorum.Leader)
[2015-07-24 17:59:22,691] INFO Follower sid: 1 : info : org.apache.zookeeper.server.quorum.QuorumPeer@jorim@im9999@10.3.3.543 (org.apache.zookeeper.server.quorum.LearnerHandler)
[2015-07-24 17:59:22,692] INFO Synchronizing With Follower sid: 1 maxCommittedLog=0x4000000001 lastCommittedLog=0x2000000001 peerLastZxid=0x4000000000 (org.apache.zookeeper.server.quorum.LearnerHandler)
[2015-07-24 17:59:22,692] INFO Sending DIFF (org.apache.zookeeper.server.quorum.LearnerHandler)
[2015-07-24 17:59:22,698] INFO Received NEWLEADER-ACK message from 1 (org.apache.zookeeper.server.quorum.LearnerHandler)
[2015-07-24 17:59:22,699] INFO Have quorum of supporters, sids: [ 1,2 ]; starting up and setting last processed zxid: 0x5000000000 (org.apache.zookeeper.server.quorum.Leader)

[2015-07-24 17:59:22,673] INFO Server environment:user.home=/Users/anrajput (org.apache.zookeeper.server.ZooKeeperServer)
[2015-07-24 17:59:22,673] INFO Server environment:user.dir=/Users/anrajput/Downloads/kafkatutorial/kafka_2.10-0.18.0.4 (org.apache.zookeeper.ZooKeeperServer)
[2015-07-24 17:59:22,674] INFO Created server with tickTime 2000 minSessionTimeout 4000 maxSessionTimeout 40000 datadir /tmp/kafkaTutorial/zookeeper/version-2 snapdir /tmp/kafkaTutorial/zookeeper/version-2 (org.apache.zookeeper.server.ZooKeeperServer)
[2015-07-24 17:59:22,675] INFO FOLLOWING - LEADER ELECTION TOOK - 34197 (org.apache.zookeeper.server.ZooKeeperServer)
[2015-07-24 17:59:22,692] INFO Getting a diff from the leader 0x400000001 (org.apache.zookeeper.server.QueuedLearner)
[2015-07-24 17:59:22,696] INFO Snappshutting: 0x400000001 to /tmp/kafkaTutorial/zookeeper/versien-2/snapshot.400000001 (org.apache.zookeeper.server.persistence.FileTxnSnapshotLog)
[2015-07-24 17:59:23,001] INFO [0x400000001]: Accepted socket connection from 72.17.6.1:16238 (org.apache.zookeeper.server.TCPServer)
[2015-07-24 17:59:23,004] INFO Client attempting to renew session 0x1561f8ccf430000 at /127.0.0.1:62585 (org.apache.zookeeper.server.ZooKeeperServer)
[2015-07-24 17:59:23,004] INFO Revalidating client: 0x1561f8ccf430000 (org.apache.zookeeper.server.QueuedLearner)
[2015-07-24 17:59:23,009] INFO Established session 0x1561f8ccf430000 with negotiated timeout 30000 for client /127.0.0.1:62585 (org.apache.zookeeper.server.ZooKeeperServer)
[2015-07-24 17:59:22,673] INFO LEADING - LEADER ELECTION TOOK - 226 (org.apache.zookeeper.server.Quorum.Leader)
[2015-07-24 17:59:22,691] INFO Follower sid: 1 : info : org.apache.zookeeper.server.quorum.QuorumPeer@jorim@im9999@10.3.3.543 (org.apache.zookeeper.server.quorum.LearnerHandler)
[2015-07-24 17:59:22,692] INFO Synchronizing With Follower sid: 1 maxCommittedLog=0x4000000001 lastCommittedLog=0x2000000001 peerLastZxid=0x4000000000 (org.apache.zookeeper.server.quorum.LearnerHandler)
[2015-07-24 17:59:22,692] INFO Sending DIFF (org.apache.zookeeper.server.quorum.LearnerHandler)
[2015-07-24 17:59:22,698] INFO Received NEWLEADER-ACK message from 1 (org.apache.zookeeper.server.quorum.LearnerHandler)
[2015-07-24 17:59:22,699] INFO Have quorum of supporters, sids: [ 1,2 ]; starting up and setting last processed zxid: 0x5000000000 (org.apache.zookeeper.server.quorum.Leader)
```

Now u can see errors gone , because now
quorum maintained .

Now go for broker settings :

```

(my state) {org.apache.zookeeper.server.quorum.FastLeaderElection}
[2016-07-24 18:02:31,212] INFO Accepted socket connection from /127.0.0.1:62597 (org.apache.zookeeper.server.NIOServerCnxnFactory)
[2016-07-24 18:02:31,219] INFO Client attempting to establish new session at /127.0.0.1:62597 (org.apache.zookeeper.server.ZooKeeperServer)
[2016-07-24 18:02:31,220] WARN Got zxid 0x5000000001 expected 0x1 (org.apache.zookeeper.server.quorum.Leader)
[2016-07-24 18:02:31,220] INFO Creating new log file: log.5000000001 (org.apache.zookeeper.server.PreProcessor)
[2016-07-24 18:02:31,224] INFO Established session 0x1561f9040c70000 with negotiated timeout 6000 for client /127.0.0.1:62597 (org.apache.zookeeper.server.ZooKeeperServer)
[2016-07-24 18:02:31,224] INFO Closed socket connection for client /127.0.0.1:62597 which had sessionid 0x1561f9040c70000 (org.apache.zookeeper.server.NIOServerCnxn)
[2016-07-24 18:12:49,420] INFO Accepted socket connection from /127.0.0.1:62643 (org.apache.zookeeper.server.NIOServerCnxn)
[2016-07-24 18:12:49,567] INFO Client attempting to establish new session at /127.0.0.1:62643 (org.apache.zookeeper.server.ZooKeeperServer)
[2016-07-24 18:12:49,567] INFO Established session 0x1561f9040c70001 with negotiated timeout 6000 for client /127.0.0.1:62643 (org.apache.zookeeper.server.ZooKeeperServer)
[2016-07-24 18:12:49,567] INFO Closed socket connection for client /127.0.0.1:62643 which had sessionid 0x1561f9040c70001 (org.apache.zookeeper.server.NIOServerCnxn)

log.segment.bytes=1073741824
# The interval at which log segments are checked to see if they can be deleted according
# to the retention policies
log.retention.check.interval.ms=300000
#####
# Zookeeper connection string (see zookeeper docs for details).
# This is a comma separated host:port pairs, each corresponding to a zk
# server. e.g. "127.0.0.1:3000,127.0.0.1:3002".
# You can also append an optional chroot string to the urls to specify the
# root directory for all kafka znodes.
zookeeper.connect=127.0.0.1:2181,127.0.0.1:2182,127.0.0.1:2183
# Timeout in ms for connecting to zookeeper
zookeeper.connection.timeout.ms=6000

-- INSERT --
ankits$ [kafka_2.10-0.10.0]

```

In server.properties file

Add all zookeeper server details under

Zookeerer.connect = ;;;

Then copy this server.properties to
server1.properties and server2.properties

(my state) (org.apache.zookeeper.server.quorum.FastLeaderElection)

```
[2015-07-24 18:02:31,212] INFO Accepted socket connection from /127.0.0.1:62597 (org.apache.zookeeper.server.NIOServerCnxnFactory)
[2015-07-24 18:02:31,219] INFO Client attempting to establish new session at /127.0.0.1:62597 (org.apache.zookeeper.server.ZooKeeperServer)
[2015-07-24 18:02:31,228] WARN Got zxid 0x5000000001 expected 0x1 (org.apache.zookeeper.server.quorum.Leader)
[2015-07-24 18:02:31,228] INFO Creating new log file: log.5000000001 (org.apache.zookeeper.server.ZooKeeperServer)
[2015-07-24 18:02:31,224] INFO Established session 0x1561f9040c70000 with negotiated timeout 60000 for client /127.0.0.1:62597 (org.apache.zookeeper.server.ZooKeeperServer)
[2015-07-24 18:02:44,004] INFO Closed socket connection for client /127.0.0.1:62597 which had sessionid 0x1561f9040c70000 (org.apache.zookeeper.server.NIOServerCnxn)
[2015-07-24 18:12:49,120] INFO Accepted socket connection from /127.0.0.1:62643 (org.apache.zookeeper.server.NIOServerCnxnFactory)
[2015-07-24 18:12:49,126] INFO Client attempting to establish new session at /127.0.0.1:62643 (org.apache.zookeeper.server.ZooKeeperServer)
[2015-07-24 18:12:49,567] INFO Closed socket connection for client /127.0.0.1:62643 which had sessionid 0x1561f9040c70001 (org.apache.zookeeper.server.ZooKeeperServer)
[2015-07-24 18:13:38,089] INFO Closed socket connection for client /127.0.0.1:62643 which had sessionid 0x1561f9040c70001 (org.apache.zookeeper.server.NIOServerCnxn)
[]
```

kafka_2.10-0.10.0.0

```
[561f9040c70001 type:create cxid:0x27 zxid:0x5000000000 txntype:-1 repath:/n/a Error Path:/brokers Error:KeeperErrorCode = NodeExists for /brokers (org.apache.zookeeper.server.PreRequ estProcessor)
[2015-07-24 18:12:49,961] INFO Got user-level KeeperException when processing sessionid:0x1 561f9040c70001 type:create cxid:0x28 zxid:0x5000000010 txntype:-1 repath:/n/a Error Path:/bro kers/ids Error:KeeperErrorCode = NodeExists for /brokers/ids (org.apache.zookeeper.server.PreRequ estProcessor)
[2015-07-24 18:12:49,961] INFO Processed session termination for sessionid: 0x3561f900fb00 0 (org.apache.zookeeper.server.PreRequestProcessor)
[2015-07-24 18:12:49,739] INFO Client attempting to establish new session at /127.0.0.1:626 01 (org.apache.zookeeper.server.ZooKeeperServer)
[2015-07-24 18:12:49,745] INFO Established session 0x3561f900fb0009 with negotiated timeout 60000 for client /127.0.0.1:62601 (org.apache.zookeeper.server.ZooKeeperServer)
[2015-07-24 18:12:54,420] INFO Closed socket connection for client /127.0.0.1:62601 which had sessionid 0x3561f900fb0009 (org.apache.zookeeper.server.NIOServerCnxn)
[2015-07-24 18:12:59,721] INFO Accepted socket connection from /127.0.0.1:62616 (org.apache.zookeeper.server.NIOServerCnxnFactory)
[2015-07-24 18:12:59,265] INFO Client attempting to establish new session at /127.0.0.1:626 46 (org.apache.zookeeper.server.ZooKeeperServer)
[2015-07-24 18:12:59,265] INFO Established session 0x3561f900fb0091 with negotiated timeout 60000 for client /127.0.0.1:62646 (org.apache.zookeeper.server.ZooKeeperServer)
[2015-07-24 18:12:59,381] INFO Closed socket connection for client /127.0.0.1:62646 which had sessionid 0x3561f900fb0091 (org.apache.zookeeper.server.NIOServerCnxn)
[]
```

kafka_2.10-0.10.0.0

```
[2015-07-24 18:02:49,734] INFO Accepted socket connection from /127.0.0.1:62601 (org.apache.zookeeper.server.ZooKeeperServer)
[2015-07-24 18:02:49,739] INFO Client attempting to establish new session at /127.0.0.1:626 01 (org.apache.zookeeper.server.ZooKeeperServer)
[2015-07-24 18:02:49,745] INFO Established session 0x3561f900fb0009 with negotiated timeout 60000 for client /127.0.0.1:62601 (org.apache.zookeeper.server.ZooKeeperServer)
[2015-07-24 18:02:49,745] INFO Closed socket connection for client /127.0.0.1:62601 which had sessionid 0x3561f900fb0009 (org.apache.zookeeper.server.NIOServerCnxn)
[2015-07-24 18:12:49,120] INFO Accepted socket connection from /127.0.0.1:62643 (org.apache.zookeeper.server.NIOServerCnxnFactory)
[2015-07-24 18:12:49,126] INFO Client attempting to establish new session at /127.0.0.1:62643 (org.apache.zookeeper.server.ZooKeeperServer)
[2015-07-24 18:12:49,567] INFO Closed socket connection for client /127.0.0.1:62643 which had sessionid 0x1561f9040c70001 (org.apache.zookeeper.server.ZooKeeperServer)
[2015-07-24 18:13:38,089] INFO Closed socket connection for client /127.0.0.1:62643 which had sessionid 0x1561f9040c70001 (org.apache.zookeeper.server.NIOServerCnxn)
[]
```

kafka_2.10-0.10.0.0

```
[561f9040c70001 type:create cxid:0x27 zxid:0x5000000000 txntype:-1 repath:/n/a Error Path:/brokers Error:KeeperErrorCode = NodeExists for /brokers (org.apache.zookeeper.server.PreRequ estProcessor)
[2015-07-24 18:12:49,961] INFO Got user-level KeeperException when processing sessionid:0x1 561f9040c70001 type:create cxid:0x28 zxid:0x5000000010 txntype:-1 repath:/n/a Error Path:/bro kers/ids Error:KeeperErrorCode = NodeExists for /brokers/ids (org.apache.zookeeper.server.PreRequ estProcessor)
[2015-07-24 18:12:49,961] INFO Processed session termination for sessionid: 0x3561f900fb00 0 (org.apache.zookeeper.server.PreRequestProcessor)
[2015-07-24 18:12:49,961] INFO Accepted socket connection from /127.0.0.1:62601 (org.apache.zookeeper.server.NIOServerCnxnFactory)
[2015-07-24 18:12:49,961] INFO Client attempting to establish new session at /127.0.0.1:626 01 (org.apache.zookeeper.server.ZooKeeperServer)
[2015-07-24 18:12:49,961] INFO Established session 0x3561f900fb0009 with negotiated timeout 60000 for client /127.0.0.1:62601 (org.apache.zookeeper.server.ZooKeeperServer)
[2015-07-24 18:12:49,961] INFO Closed socket connection for client /127.0.0.1:62601 which had sessionid 0x3561f900fb0009 (org.apache.zookeeper.server.NIOServerCnxn)
[2015-07-24 18:12:59,721] INFO Accepted socket connection from /127.0.0.1:62616 (org.apache.zookeeper.server.NIOServerCnxnFactory)
[2015-07-24 18:12:59,265] INFO Client attempting to establish new session at /127.0.0.1:626 46 (org.apache.zookeeper.server.ZooKeeperServer)
[2015-07-24 18:12:59,265] INFO Established session 0x3561f900fb0091 with negotiated timeout 60000 for client /127.0.0.1:62646 (org.apache.zookeeper.server.ZooKeeperServer)
[2015-07-24 18:12:59,381] INFO Closed socket connection for client /127.0.0.1:62646 which had sessionid 0x3561f900fb0091 (org.apache.zookeeper.server.NIOServerCnxn)
[]
```

kafka_2.10-0.10.0.0

```
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
# see kafka.server.KafkaConfig for additional details and defaults
#####
##### Server Basics #####
# The id of the broker. This must be set to a unique integer for each broker.
broker.id=1
#####
##### Socket Server Settings #####
# The address the socket server listens on. It will get the value returned from
# java.net.InetAddress.getCanonicalHostName() if not configured.
# FORMAT:
# listeners = security_protocol://host_name:port
# EXAMPLE:
# listeners = PLAINTEXT://your.host.name:9092
#listeners=PLAINTEXT://:9092
-- INSERT --
```

ankits\$

Change port to 9093

Change broker id to 1

Change log directory

- Same do for server3.properties

Change port to 9094

Change broker id to 2

Change log directory

The image shows three terminal windows side-by-side, each displaying log output from a Kafka or Zookeeper server. The top window is titled 'ankitsbin/kafka-server-start.sh config/server.properties' and contains Zookeeper logs. The middle window is titled 'kafka_2.10-0.10.0.0' and contains Zookeeper logs. The bottom window is also titled 'kafka_2.10-0.10.0.0' and contains Kafka logs. All logs show various INFO and WARN messages related to socket connections, session establishment, and session termination.

```

[my_state] (org.apache.zookeeper.server.quorum.FastLeaderElection)
[2015-07-24 18:02:31,212] INFO Accepted socket connection from /127.0.0.1:62597 (org.apache.zookeeper.server.NIOServerCnxnFactory)
[2015-07-24 18:02:31,219] INFO Client attempting to establish new session at /127.0.0.1:625
97 (org.apache.zookeeper.server.ZooKeeperServer)
[2015-07-24 18:02:31,228] WARN Got zxid 0x500000001 expected 0x (org.apache.zookeeper.serv
er.quorum.leader)
[2015-07-24 18:02:31,228] INFO Creating new log file: log.500000001 (org.apache.zookeeper.s
erver.log)
[2015-07-24 18:02:31,224] INFO Established session 0x1561f9840c70000 with negotiated timeout
6400 for client /127.0.0.1:62597 (org.apache.zookeeper.server.ZooKeeperServer)
[2015-07-24 18:02:31,044] INFO Closed socket connection for client /127.0.0.1:62597 which h
ad sessionid 0x1561f9040c70000 (org.apache.zookeeper.server.NIOServerCnxn)
[2015-07-24 18:12:49,120] INFO Accepted socket connection from /127.0.0.1:62543 (org.apache.z
ookeeper.server.NIOServerCnxnFactory)
[2015-07-24 18:12:49,126] INFO Client attempting to establish new session at /127.0.0.1:626
43 (org.apache.zookeeper.server.ZooKeeperServer)
[2015-07-24 18:12:49,567] INFO Established session 0x1561f9840c70001 with negotiated timeout
6400 for client /127.0.0.1:62643 (org.apache.zookeeper.server.ZooKeeperServer)
[2015-07-24 18:13:38,001] INFO Closed socket connection for client /127.0.0.1:62643 which h
ad sessionid 0x1561f9040c70001 (org.apache.zookeeper.server.NIOServerCnxn)
[2015-07-24 18:12:49,730] INFO Accepted socket connection from /127.0.0.1:62601 (org.apache.z
ookeeper.server.NIOServerCnxnFactory)
[2015-07-24 18:12:49,739] INFO Client attempting to establish new session at /127.0.0.1:62
601 (org.apache.zookeeper.server.ZooKeeperServer)
[2015-07-24 18:12:49,745] INFO Established session 0x1561f9840c70000 with negotiated timeout
6400 for client /127.0.0.1:62601 (org.apache.zookeeper.server.ZooKeeperServer)
[2015-07-24 18:12:54,461] INFO Closed socket connection for client /127.0.0.1:62601 which h
ad sessionid 0x3561f9040c70000 (org.apache.zookeeper.server.NIOServerCnxn)
[2015-07-24 18:12:59,721] INFO Accepted socket connection from /127.0.0.1:62616 (org.apache.z
ookeeper.server.NIOServerCnxnFactory)
[2015-07-24 18:12:59,265] INFO Client attempting to establish new session at /127.0.0.1:62
646 (org.apache.zookeeper.server.ZooKeeperServer)
[2015-07-24 18:12:59,265] INFO Established session 0x3561f9840c70001 with negotiated timeout
6400 for client /127.0.0.1:62646 (org.apache.zookeeper.server.ZooKeeperServer)
[2015-07-24 18:12:59,481] INFO Closed socket connection for client /127.0.0.1:62645 which h
ad sessionid 0x3561f9040c70001 (org.apache.zookeeper.server.NIOServerCnxn)

```

Now start all 3 server (broker server)

All 3 broker and 3 zookeeper server up and running now

Now create the topic

With all zookeeper server and RF =3

After creating the topic start the producer

Now start the consumer

Now u can kill one zookeeper and then see u
can still send the messages and deliver to the
consumer

(Because at this time other two zookeeper jumped in and performed the quorum and support it 3 broker server)

=====**END**=====

