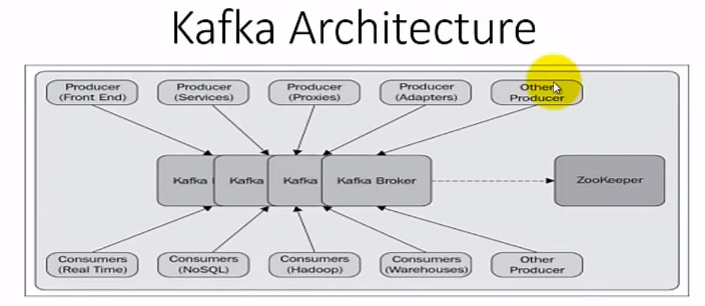
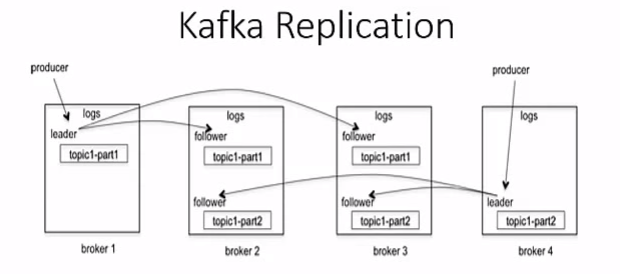
**Quick Ref: Kafka**

****

****

|  |  |  |
| --- | --- | --- |
| **S.No** | **Topic** | **Desc** |
|  | **Web Ref** |  |
|  | Intro | <https://www.youtube.com/watch?v=ArUHr3Czx-8> |
|  | Sample program | <https://www.youtube.com/watch?v=1Og9n9FJteM> |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  | **General Info** |  |
| 1 | Pre-request | # List of pre-request   1. Zookeeper for coordination 2. Java JDK 3. Kafka Binaries |
| 2 | About Kafka | **Kafka is a distributed messaging system**  It is next generation distributed messaging system |
| 2.1 | **Usage of Kafka** | **# Data is the main ingredient of Internet applications and typically includes the following**   1. Web Page visits and clicks 2. User activities 3. Events corresponding to logins 4. Social networking activities such as likes, shares and comments 5. Application specific metrics (ex: logs, page load time, performance, etc) |
| 2.2 | User activity | **# What is user activity means? Need an example**  Every action that user performs are user activity history. Few ex:   1. **Social media:** every login, duration of stay, in which page they are entering, where is exiting, links clicked, etc. Ex: There are 1 million users are using your website and you have 100 web pages then think how many traffic you have every moment 2. **In telecom:**  Every message, phone calls that you makes, duration, voice mail , etc 3. **Setup box:** every channel you watch, how long you are watching, how frequently you are changing |
| 2.3 | Messaging system | **# Use of messaging system**  Problems in the present big-data era:   1. The very first challenge is to collect the data as it is huge 2. The second challenge is analyze the huge data   Solution:   1. Use messaging system to solve this problem , because it provides seamless integration among distributed applications with the help of messages, that are shared between them |
| 3 | Important Components of Kafka | # Important components   1. Broker 2. Replication 3. Partition 4. Leader 5. Zookeeper |
| 3.1 | Broker | 1. Broker is a key component in Kafka, it is a server/cluster which managing topic |
| 3.2 | Replication | 1. Kafka is very powerful in terms of availability 2. Ex: If you have 3 broker, even one goes down, we can get topic from other brokers replicas 3. No of Replica = No of Active Broker 4. So whenever you create a new topic, specify the replication factor |
| 3.3 | Partition | 1. Partition is used to improve the performance of the Kafka 2. Each partition will be a leader |
| 3.4 | Leader | 1. Leader will retrieve the information from the producer and pushing into current partition as well as make replica to other borkers 2. If one leader is goes down then Kafka will automatically use another available leader from the replica |
| 3.5 | Zookeeper | 1. It is a coordinator 2. It maintains metadata information 3. Notify the consumer in case of any change, so that consumer will pull the new information from the topic |
| 3.6 | Topic? |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  | **commands** |  |
| 1 | Kafka folder structure | # This folder has two sub directories  >cd kafka/kafka\_2.9.2-6.8.15   1. bin 2. config |
| 2 | >cd config | # Config has 4 properties. First two are properties for server and remaining are properties for Kafka utilities   1. zookeeper.properties 2. server.properties *-- Contains kafka server details* 3. producer.properties 4. consumer.properties |
| 2.1 | Server.properties | # Contains kafka server details   1. broker\_id = 0 *– Unique integer for each broker* 2. Port=9092 3. Log.dirs=/tmp/kafka-logs 4. Num.network.threads=3 *– default value* 5. Num.partition=1 *– default value* 6. Zookeeper.connect=ip\_address:2181 *-- 2181 is zk port* |
| 2.2 | Zookeeper.properties | # Mainly two to three properties   1. dataDir=/tmp/zookeeper 2. clientPort=2181 3. maxClientCnxns=0 |
| 3 | **>cd bin** | # bin has following daemons. Need to start the below daemons  # The first two are server daemons and remaining two are utilities  Major daemons:   1. zookeeper-server-start.sh 2. kafka-server-start.sh 3. kafka-console-producer.sh *-- Kafka Utilities* 4. kafka-console-consumer.sh *– Kafka Utilities*   Additional daemons:   1. kafka-topic.sh |
| 3.1 | Zookeeper-server-start.bat | **# Step 1**: start zookeeper. Pass zookeeper.properties as parameter   1. >cd kafka\kafka\_2.9.2-6.8.15\ 2. >bin\windows-or-linux\ Zookeeper-server-start.bat config\zookeeper.properties |
| 3.2 | kafka-server-start.sh | **# Step2:** kafka-server-start.bat. Pass server.properties as parameter   1. >cd kafka\kafka\_2.9.2-6.8.15\ 2. >bin\windows-or-linux\kafka-server-start.bat config\server.properties |
| 3.3 | Kafka-topics.sh | **# Step3:** Create topic  # Replication factor passed as 3 in the below example but before you decide this, check Server.properties, if property defined replication factor as less than 3, it will through an error   1. >bin/kafka-topics.batch --create –zookeeper ip\_addr:2181 --replication-factor 3 --partitions 1 --topic <TOPIC\_NAME>   Ex:   1. >>bin/kafka-topics.batch --create –zookeeper ip\_addr:2181 --replication-factor 3 --partitions 1 --topic test\_Topic |
| 3.4 | List of Topics | # Step 3.1: List topics   1. >bin/windows/kafka-topics.batch --list –zookeeper ip\_addr:2181 |
| 3.5 | Kafka-console-producer.sh | **# Step 4: Create producer (pass kafka server ip & port as parameter in broker\_list)**  # Use Kafka utilities (producer) to validate Kafka is properly configured  **# start producer and do NOT close producer window. Use separate window for consumer**   1. >bin/windows/kafka-console-producer.batch --broker-list <*BROKER\_LIST*> --topic <*TOPIC\_NAME*> 2. BROKER\_LIST: ip\_addr:port of broker kafka server (ex: 127.0.0.1:**9092**) 3. You can pass more than one comma separated broker\_list 4. TOPIC\_NAME: test\_Topic   **Note: To validate:** Once producer started and consumer started… whatever you enter from producer window will be displayed in consumer window. |
| 3.6 | Kafka-console-consumer.sh | **# Step 5:** **Create consumer (pass zookeeper ip\_addr & port as parameter)**  # Open a new linux window for consumer, do NOT close or use producer window for validation. If you are using single node  # Start consumer from a different node *– TB verified*  # Use console consumer   1. >bin/kafka-console-consumer.sh --zookeeper ip\_addr:2181 --topic <TOPIC\_NAME> |
| 3.7 | Server with replication factor | **# Step 6**: Server with replication factor   1. >bin/kafka-topics.sh --create --zookeeper ip\_addr:2181 --replication-facotr <INT\_REPLICATION\_FACTOR> --partitions <INT\_PARTITION> |
| 3.8 | Descritption of Kafka setup | **# Step 7:** Description of Kafka setup   1. >bin/kafka-topics.sh --describe --zookeeper ip\_addr:2181 --topic <TOPIC\_NAME> |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |