Introduction to Kafka

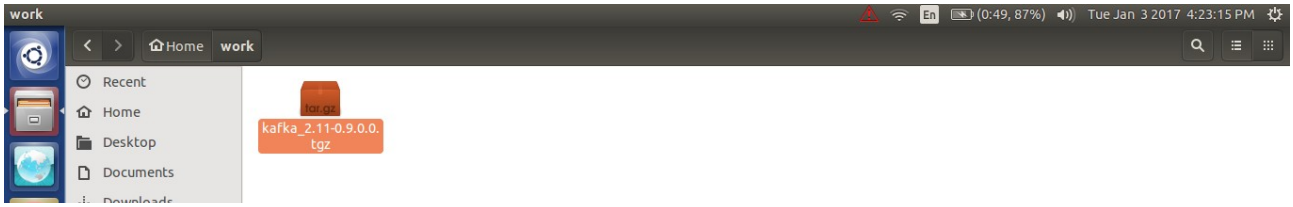
- Kafka Installation
- Kafka Practice Commands
- Kafka Cluster Practice Commands

Kafka Installation

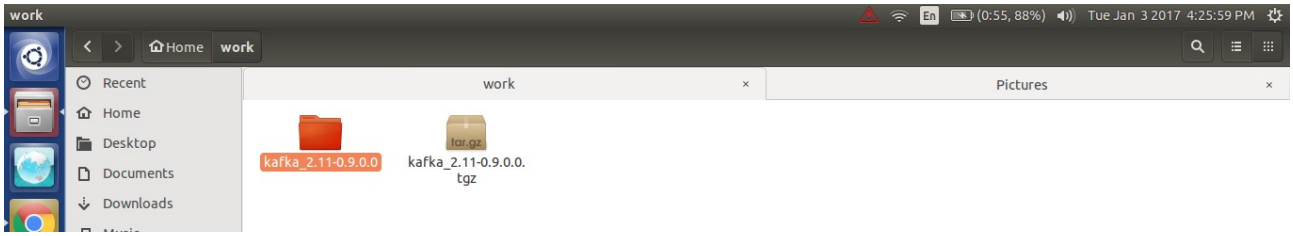
1. Create `**/home/orienit/work**` folder using below command
   - *mkdir /home/orienit/work*

2. Download `**kafka_2.11-0.9.0.0.tgz**` file from this link.
   (https://archive.apache.org/dist/kafka/0.9.0.0/kafka_2.11-0.9.0.0.tgz)

3. Copy `**kafka_2.11-0.9.0.0.tgz**` file into `**/home/orienit/work**` folder

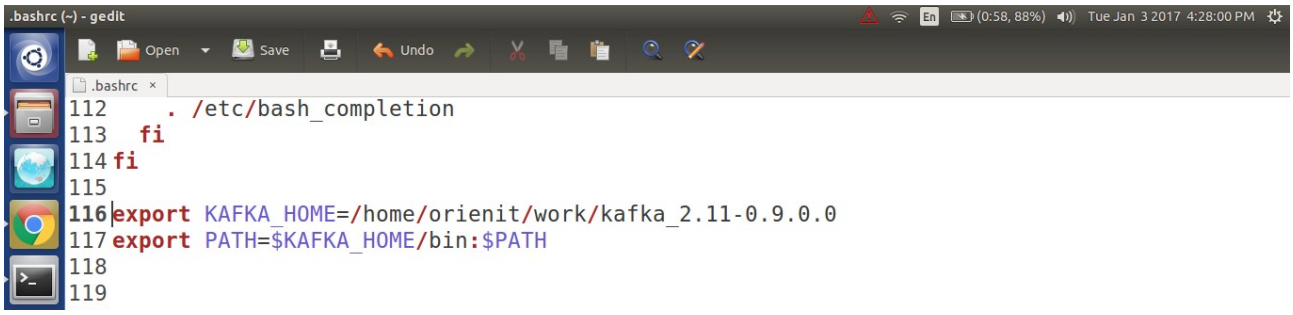


4. Extract `**kafka_2.11-0.9.0.0.tgz**` file into `**/home/orienit/work**` folder



5. Execute below command to open `~/**.bashrc**` file
   - **Command:** gedit ~/.bashrc

6. Update `~/**.bashrc**` file with below changes
   - export KAFKA_HOME=/home/orienit/work/kafka_2.11-0.9.0.0
   - export PATH=$KAFKA_HOME/bin:$PATH



7. Re-open the terminal
8. Verify with below command
   - **Command:** echo $KAFKA_HOME

9. Let's play with kafka

Kafka Practice Commands

1. Follow above procedure to install the kafka
2. Start the `**zookeeper**` using below command (New Terminal)

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

```
orienit@kalyan:~$ $KAFKA_HOME/bin/zookeeper-server-start.sh $KAFKA_HOME/config/zookeeper.properties

[2017-01-03 16:41:44,986] INFO Reading configuration from: /home/orienit/work/kafka_2.11-0.9.0.0/co
nfig/zookeeper.properties (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2017-01-03 16:41:45,024] INFO autopurge.snapRetainCount set to 3 (org.apache.zookeeper.server.Data
dirCleanupManager)
[2017-01-03 16:41:45,024] INFO autopurge.purgeInterval set to 0 (org.apache.zookeeper.server.Datadi
rCleanupManager)
[2017-01-03 16:41:45,024] INFO Purge task is not scheduled. (org.apache.zookeeper.server.DatadirCle
anupManager)
[2017-01-03 16:41:45,025] WARN Either no config or no quorum defined in config, running  in standal
one mode (org.apache.zookeeper.server.quorum.QuorumPeerMain)
[2017-01-03 16:41:45,086] INFO Reading configuration from: /home/orienit/work/kafka_2.11-0.9.0.0/co
nfig/zookeeper.properties (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2017-01-03 16:41:45,086] INFO Starting server (org.apache.zookeeper.server.ZooKeeperServerMain)
[2017-01-03 16:41:45,117] INFO Server environment:zookeeper.version=3.4.6-1569965, built on 02/20/2
014 09:09 GMT (org.apache.zookeeper.server.ZooKeeperServer)
[2017-01-03 16:41:45,117] INFO Server environment:host.name=localhost (org.apache.zookeeper.server.
ZooKeeperServer)
[2017-01-03 16:41:45,117] INFO Server environment:java.version=1.8.0_66-internal (org.apache.zookee
per.server.ZooKeeperServer)
[2017-01-03 16:41:45,117] INFO Server environment:java.vendor=Oracle Corporation (org.apache.zookee
```

3. Start the `**kafka server**` using below command (New Terminal)

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

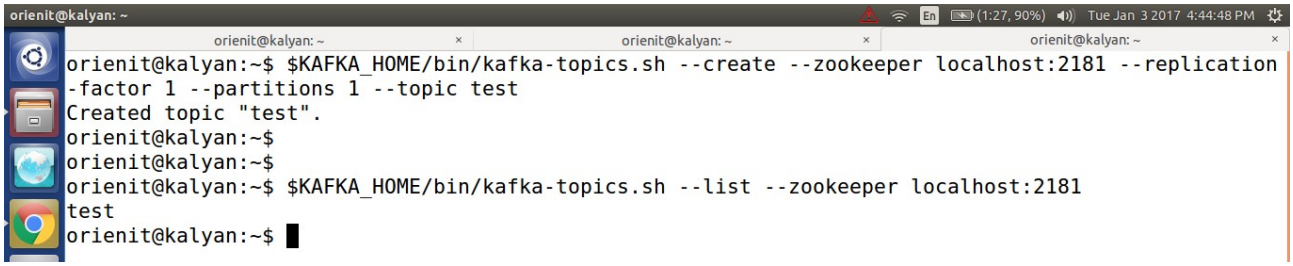
```
orienit@kalyan:~$ $KAFKA_HOME/bin/kafka-server-start.sh $KAFKA_HOME/config/server.properties
[2017-01-03 16:43:14,430] INFO KafkaConfig values:
        advertised.host.name = null
        metric.reporters = []
        quota.producer.default = 9223372036854775807
        offsets.topic.num.partitions = 50
        log.flush.interval.messages = 9223372036854775807
        auto.create.topics.enable = true
        controller.socket.timeout.ms = 30000
        log.flush.interval.ms = null
        principal.builder.class = class org.apache.kafka.common.security.auth.DefaultPrincipalBuild
er
        replica.socket.receive.buffer.bytes = 65536
        min.insync.replicas = 1
        replica.fetch.wait.max.ms = 500
        num.recovery.threads.per.data.dir = 1
        ssl.keystore.type = JKS
        default.replication.factor = 1
        ssl.truststore.password = null
        log.preallocate = false
```

4. Create a `**test**` topic using below command (New Terminal)

   $KAFKA_HOME/bin/kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1 --topic test

5. List out all the topics

   $KAFKA_HOME/bin/kafka-topics.sh --list --zookeeper localhost:2181



6. Start the `**kafka producer**` using below command (New Terminal)

   $KAFKA_HOME/bin/kafka-console-producer.sh --broker-list localhost:9092 --topic test

7. Start the `**kafka consumer**` using below command (New Terminal)

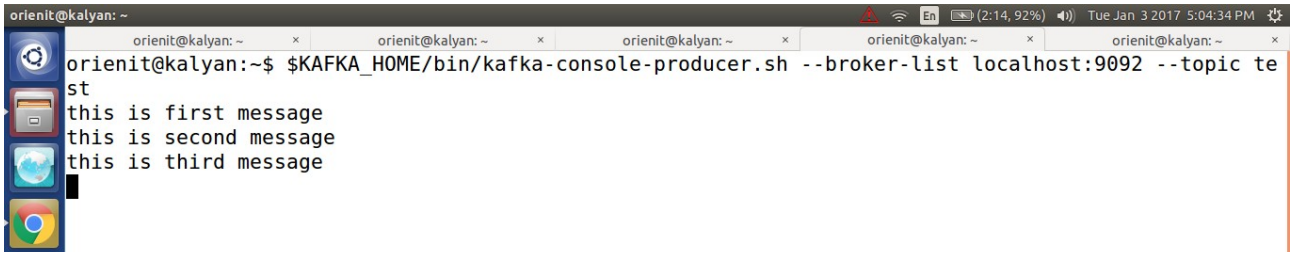
   - **Command in Kafka-0.9.x**

   $KAFKA_HOME/bin/kafka-console-consumer.sh --zookeeper localhost:2181 --topic test --from-beginning

   - **Command in Kafka-0.10.x**

   $KAFKA_HOME/bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic test --from-beginning

8. After starting the `**kafka producer**` then Send some messages to `**kafka producer**` like below



9. After starting the `**kafka consumer**` then Recieve some messages from `**kafka producer**` like below

```
orienit@kalyan:~$ $KAFKA_HOME/bin/kafka-console-consumer.sh --zookeeper localhost:2181 --topic test
 --from-beginning
this is first message
this is second message
this is third message
```

Kafka Cluster Practice Commands

10. Create new configuration files in `**$KAFKA_HOME/config**` folder for kafka brokers

    cp $KAFKA_HOME/config/server.properties $KAFKA_HOME/config/server-1.properties

    cp $KAFKA_HOME/config/server.properties $KAFKA_HOME/config/server-2.properties

```
orienit@kalyan:~$ cp $KAFKA_HOME/config/server.properties $KAFKA_HOME/config/server-1.properties
orienit@kalyan:~$ cp $KAFKA_HOME/config/server.properties $KAFKA_HOME/config/server-2.properties
orienit@kalyan:~$
```

11. Now edit these new files and set the following properties:

    $KAFKA_HOME/config/server-1.properties:

    broker.id=1

    listeners=PLAINTEXT://:9093

    log.dir=/tmp/kafka-logs-1

    $KAFKA_HOME/config/server-2.properties:

    broker.id=2

    listeners=PLAINTEXT://:9094

    log.dir=/tmp/kafka-logs-2

12. Start the `**kafka server**` using below command (New Terminal)

    $KAFKA_HOME/bin/kafka-server-start.sh $KAFKA_HOME/config/server-1.properties

13. Start the `**kafka server**` using below command (New Terminal)

    $KAFKA_HOME/bin/kafka-server-start.sh $KAFKA_HOME/config/server-2.properties

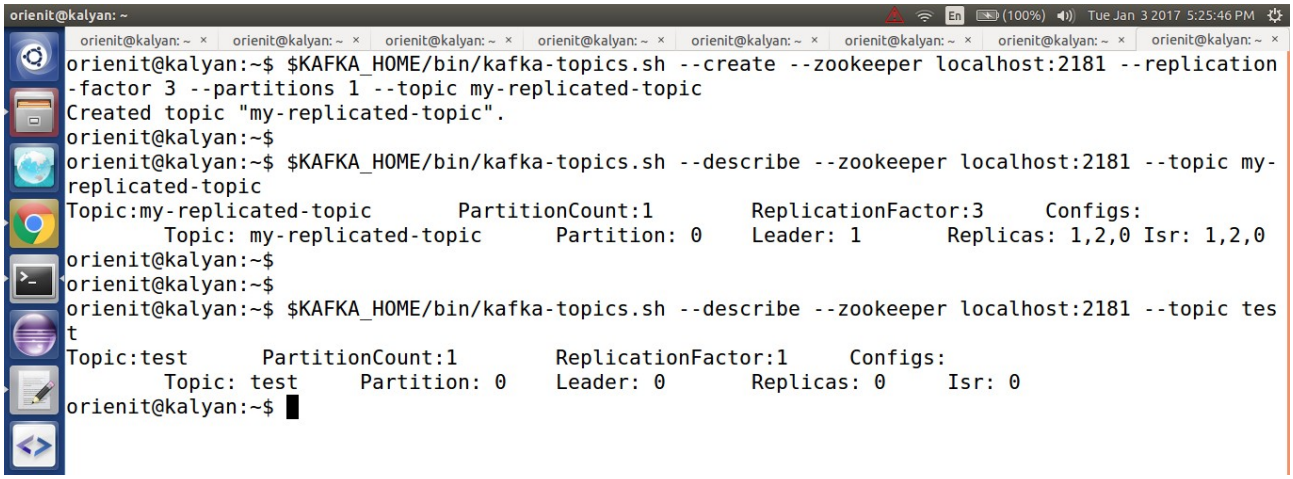
14. Now create a new topic with a replication factor of three:

    $KAFKA_HOME/bin/kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 3 --partitions 1 --topic my-replicated-topic

15. Descibe the topics run below command

    $KAFKA_HOME/bin/kafka-topics.sh --describe --zookeeper localhost:2181 --topic my-replicated-topic

    $KAFKA_HOME/bin/kafka-topics.sh --describe --zookeeper localhost:2181 --topic test



16. Start the `**kafka producer**` using below command (New Terminal)

    $KAFKA_HOME/bin/kafka-console-producer.sh --broker-list localhost:9092 --topic my-replicated-topic

17. Start the `**kafka consumer**` using below command (New Terminal)

    - **Command in Kafka-0.9.x**

    $KAFKA_HOME/bin/kafka-console-consumer.sh --zookeeper localhost:2181 --topic my-replicated-topic --from-beginning

    - **Command in Kafka-0.10.x**

    $KAFKA_HOME/bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic my-replicated-topic --from-beginning

18. After starting the `**kafka producer**` then Send some messages to `**kafka producer**` like below

```
orienit@kalyan:~$ $KAFKA_HOME/bin/kafka-console-producer.sh --broker-list localhost:9092 --topic my
-replicated-topic
replica message 1
replica message 2
replica message 3
```

19. After starting the `**kafka consumer**` then Recieve some messages from `**kafka producer**` like below

```
orienit@kalyan:~$ $KAFKA_HOME/bin/kafka-console-consumer.sh --zookeeper localhost:2181 --topic my-r
eplicated-topic --from-beginning
replica message 1
replica message 2
replica message 3
```

20. Descibe the topics run below command

    $KAFKA_HOME/bin/kafka-topics.sh --describe --zookeeper localhost:2181 --topic my-replicated-topic

```
orienit@kalyan:~$ $KAFKA_HOME/bin/kafka-topics.sh --describe --zookeeper localhost:2181 --topic my-
replicated-topic
Topic:my-replicated-topic       PartitionCount:1        ReplicationFactor:3     Configs:
        Topic: my-replicated-topic      Partition: 0    Leader: 1       Replicas: 1,2,0 Isr: 1,2,0
orienit@kalyan:~$
```

21. Now let's test out fault-tolerance. Broker 1 was acting as the leader (as per above screenshot)

22. Get the process id of Broker 1 then kill it using below command

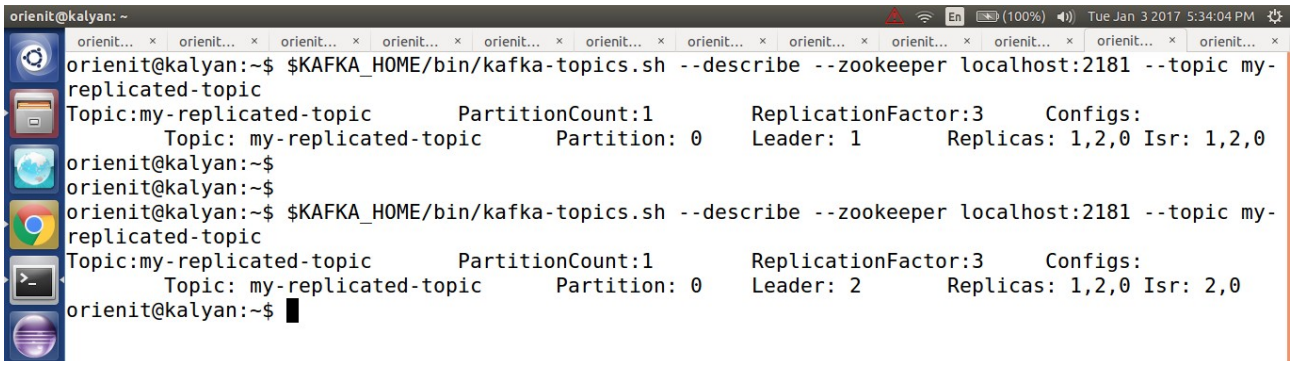
    ps aux | grep server-1.properties

    kill -9 26860

```
orienit@kalyan:~$ ps aux | grep server-1.properties
orienit   26860  2.5  1.2 4629164 208580 pts/20 Sl+  17:21   0:16 /usr/lib/jvm/java-1.8.0-openjdk/bi
n/java -Xmx1G -Xms1G -server -XX:+UseG1GC -XX:MaxGCPauseMillis=20 -XX:InitiatingHeapOccupancyPercen
t=35 -XX:+DisableExplicitGC -Djava.awt.headless=true -Xloggc:/home/orienit/work/kafka_2.11-0.9.0.0/
bin/../logs/kafkaServer-gc.log -verbose:gc -XX:+PrintGCDetails -XX:+PrintGCDateStamps -XX:+PrintGCT
imeStamps -Dcom.sun.management.jmxremote -Dcom.sun.management.jmxremote.authenticate=false -Dcom.su
n.management.jmxremote.ssl=false -Dkafka.logs.dir=/home/orienit/work/kafka_2.11-0.9.0.0/bin/../logs
 -Dlog4j.configuration=file:/home/orienit/work/kafka_2.11-0.9.0.0/bin/../config/log4j.properties -c
p :/home/orienit/work/kafka_2.11-0.9.0.0/bin/../libs/* kafka.Kafka /home/orienit/work/kafka_2.11-0.
9.0.0/config/server-1.properties
orienit   28584  0.0  0.0  13700  2128 pts/24   S+   17:32   0:00 grep --color=auto server-1.propert
ies
orienit@kalyan:~$
orienit@kalyan:~$ kill -9 26860
```

23. Descibe the topics run below command

    $KAFKA_HOME/bin/kafka-topics.sh --describe --zookeeper localhost:2181 --topic my-replicated-topic



24. Broker 2 was acting as the leader (as per above screenshot)

25. But the messages are still available for consumption even though the leader that took the writes originally is down:

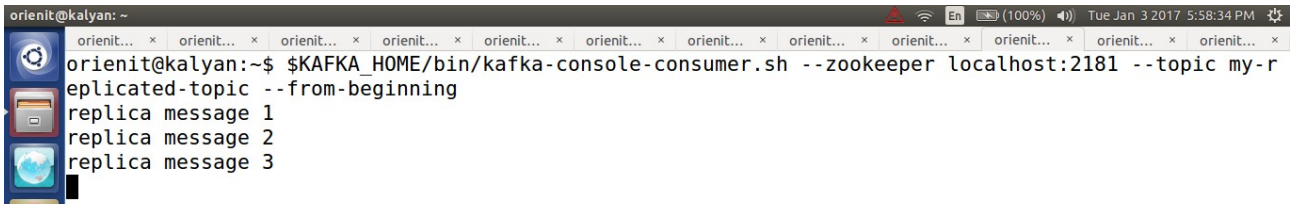
26. Start the `**kafka consumer**` using below command (New Terminal)

    - **Command in Kafka-0.9.x**

    $KAFKA_HOME/bin/kafka-console-consumer.sh --zookeeper localhost:2181 --topic my-replicated-topic --from-beginning

    - **Command in Kafka-0.10.x**

    $KAFKA_HOME/bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic my-replicated-topic –from-beginning



27. As per above information we can understand `**kafka provides fault-tolerent**`