**Kafka and zookeeper installation**

<https://www.agiratech.com/kafka-zookeeper-multi-node-cluster-setup/>

**Kafka commands**

*Create topic:*

bin/kafka-topics.sh --zookeeper $ZK\_HOSTS --create --topic $TOPIC\_NAME --replication-factor 3 --partitions 1

bin/kafka-topics.sh --zookeeper 192.168.26.133:3701 --create --topic DropcopyEx --replication-factor 3 --partitions 1

*Describe topic:*

kafka-topics.sh --describe --zookeeper <zkhost>:<zkPort> --topic <topic name>

bin/kafka-topics.sh --describe --zookeeper 192.168.26.133:3701 --topic Transformed\_Transaction\_partition-0

This will print the topic details like, number of partitions, replication factor, leader, replicas and in-sync replicas

*How to check the overridden properties of topic like retention period :*

bin/kafka-topics.sh --describe --zookeeper 192.168.26.133:3701 --topic topic7 --topics-with-overrides

*Change topic retention time: (Server level Default is 7 days)*

*(All topics used in Sentinel have 48hrs retention)*

kafka-topics.sh --zookeeper <zkhost>:<zkPort> --alter --topic <topic name> --config retention.ms= <desired time in milliseconds>

*Purge topic:*

Temporarily update the retention time on the topic to say 12hrs

kafka-topics.sh --zookeeper <zkhost>:<zkPort> --alter --topic <topic name> --config retention.ms= 43200000

Then wait for the purge to take effect. Once purged reset the retention.ms property to the old value of 48hrs

kafka-topics.sh --zookeeper <zkhost>:<zkPort> --alter --topic <topic name> --config retention.ms=172800000

kafka-topics.sh --zookeeper <zkhost>:<zkPort> --alter --topic <topic name> --config retention.ms=86400000 (7 days)

*Get current offset of any topic:*

kafka-run-class.sh kafka.tools.GetOffsetShell --broker-list <kafka-ip>:<kafka-port> --topic <topic name> --time -1

bin/kafka-run-class.sh kafka.tools.GetOffsetShell --broker-list 192.168.24.104:7701 --topic testTopic1 --time -1

*Check lag of all topics in a group (lag means the number of message which are published but have not been consumed by*

kafka-consumer-groups.sh --bootstrap-server <kafka-ip>:<kafka-port> --describe --group <group name>

*List all topics inside a broker :*

kafka-topics.sh --list –zookeeper <zkhost>:<zkPort>

bin/kafka-topics.sh --list --zookeeper 192.168.26.133:3701

(Here 3701 is zookeeper port which is defined in zookeeper.connect property in server.properties)

*Produce a message:*

bin/kafka-console-producer.sh --broker-list broker\_1\_ip:port broker\_2\_ip:port broker\_n\_ip:port --topic topic\_name

bin/kafka-console-producer.sh --broker-list 192.168.26.133:7701 --topic AKRM

(Here port is 7701 for broker which is mentioned in Listener property in server.properties)

*Consume a message: (from any broker)*

bin/kafka-console-consumer.sh --zookeeper zookeeper\_ip:port --from-beginning --topic topic\_name

bin/kafka-console-consumer.sh --bootstrap-server 192.168.26.133:7701 --from-beginning --topic AKRM

*Alter the Replication factor of an already created Topic:*

1. Increase the partition to match the replication factor

bin/kafka-topics.sh --zookeeper 192.168.26.133:3701 --alter --topic AK\_TEST --partitions 3

* Then do a describe to check that the number of partitions have now increased from 1 to 3
* bin/kafka-topics.sh --describe --zookeeper 192.168.26.133:3701 --topic AK\_TEST

1. Now create a json file with custom configuration details for the new replication factor details

using below format :

{"version":1,

"partitions":[

{"topic":"AK\_TEST","partition":0, "replicas":[0,1,2]},

{"topic":"AK\_TEST","partition":1, "replicas":[0,1,2]},

{"topic":"AK\_TEST","partition":2, "replicas":[0,1,2]}

]}

1. Now use the above json to replicate the topics across all brokers

bin/kafka-reassign-partitions.sh --zookeeper 192.168.26.133:3701 --reassignment-json-file <<FILENAME.json> --execute

bin/kafka-reassign-partitions.sh --zookeeper 192.168.26.133:3701 --reassignment-json-file bin/increase-replication-factor.json --execute

* Then do a describe to check that the number of partitions have now increased from 1 to 3
* bin/kafka-topics.sh --describe --zookeeper 192.168.24.105:3701 --topic DropcopyEx

*Delete a topic:*

1. Stop Kafka server
2. Make sure that delete.topic.enable = true is set in properties file
3. Use below command :

bin/kafka-topics.sh --zookeeper 192.168.26.134:3701 --delete --topic 4101

1. Check under kafka-logs that the corresponding topic is now deleted Or check under /brokers/topics folder (based on your configurations) if topic is deleted or not
2. Try recreating topic with same name and config to make sure it is deleted.

*To get the latest offset number of a message on a given topic:*

/home/mcxuat/kafka\_2.12-0.11.0.0/bin/kafka-run-class.sh kafka.tools.GetOffsetShell --broker-list 192.168.26.133:7701 --topic DropcopyEx --time -1

*To start kafka:*

nohup ./bin/kafka-server-start.sh config/server.properties & -> 104

nohup ./bin/kafka-server-start.sh config/server-2.properties & -> 105

*To start zookeeper:*

cd zookeeper-3.4.9/

./bin/zkServer.sh start

*Kill Kafka and Zookeeper:*

Ps-ef | grep kafka -> kill kafka pid

Ps –ef | grep zookeeper -> kill zookeeper pid

**First bring up zookeeper then restart kafka**

*Default properties while creating kafka topic from java code :*

offsets.topic.replication.factor=1

transaction.state.log.replication.factor=1

transaction.state.log.min.isr=1

num.partitions=1

auto.create.topics.enable = true

*Meta Properties path (defined in server.properties file as logs.dir)* : /home/app-admin/kafka\_messages/meta.properties

*To identify underlying brokers:*

*Inside kafka folder:*

./bin/zookeeper-shell.sh 192.168.24.105:3701 ls /brokers/ids

OR

echo dump | nc 192.168.26.135 3701 | grep brokers

You would see something like below: WATCHER::  WatchedEvent state:SyncConnected type:None path:null [0, 1, 2]

IF no broker is attached to the zookeeper the below error will occur during topic creation:

**Error while executing topic command : replication factor: 1 larger than available brokers: 0**

And ls /brokers/ids will show empty array : type:None path:null []

**Solution:**

1. Created fresh server-1.properties with new broker.id. Updated the same broker.id in **meta.properties** file located **at /home/app-admin/aashka** folder and used this folder as ***logs.dir***value in **server-1.properties** file.
2. Navigate to **zookeeper-3.4.9/conf/zoo.cfg** file and update the value server.<number> with new broker.id value i.e **server.<new myid>:**hostname (This setting is needed to avoid error “**java.lang.RuntimeException : My id 4 is not in peer list**) update <new myid> with new broker.id value

Now you can start kafka using server-1.properties as parameter :

**nohup ./bin/kafka-server-start.sh config/server-1.properties &**

*Steps to hard delete a topic :*

- cd zookeeper-3.4.9/

- bin/zkCli.sh -server 192.168.26.134:3701

- ls /brokers/topics -> rmr /brokers/topics/TOPICNAME

- ls /config/topics -> rmr /config/topics/TOPICNAME

- ls /admin/delete\_topics -> rmr /admin/delete\_topics

* Delete the topicName from kafka\_messages folder too (once the log file is empty).

To check if zookeeper and kafka are up:

* netstat -anp | grep 3701 //zookeeper
* netstat -anp | grep 7701 //kafka

**For file deletion issue during Kafka restart:**

Either delete files manually or do : **sudo –i**  -> then enter password..

Now restart Kafka. This will delete all the files

**Setting replication factor at Broker Level :**

* For issue where if one broker is down, the other 2 brokers also won’t receive any messages:
* Check **offsets.topic.replication.factor** parameter inside **kafka/config/server.properties** file and change its default “1” value to number of Brokers. In our case = 3.
* Kafka needs replication at topic level as well as broker level.
* Refer below links for the same :

<https://stackoverflow.com/questions/49390677/kafka-consumer-not-able-to-consume-messages-using-bootstrap-server-name/51540528#51540528>

<https://stackoverflow.com/questions/34844209/consumer-not-receiving-messages-kafka-console-new-consumer-api-kafka-0-9>

<https://stackoverflow.com/questions/58566578/multiple-offsets-topic-replication-factor-in-kafka-cluster>

<https://strimzi.io/blog/2021/06/08/broker-tuning/>

*Issue :*

* When trying to start Kafka, it would start on given port but none of the consuming applications were able to Join the consumer group of Kafka.
* Expected log message in any consumer application like Engine, Iris etc :

**INFO AppInfoParser: Kafka version : 0.10.0.1**

**INFO AppInfoParser: Kafka commitId : a7a17cdec9eaa6c5**

**INFO AbstractCoordinator: Discovered coordinator 192.168.26.134:7701 (id: 2147483646 rack: null) for group kafkaSpark1000.**

**INFO ConsumerCoordinator: Revoking previously assigned partitions [] for group kafkaSpark1000**

**INFO AbstractCoordinator: (Re-)joining group kafkaSpark1000**

**INFO AbstractCoordinator: Successfully joined group kafkaSpark1000 with generation 1**

**INFO ConsumerCoordinator: Setting newly assigned partitions [topic1-0] for group kafkaSpark1000**

* But the application will stop at “Started Application” or just below message

**INFO AppInfoParser: Kafka version : 0.10.0.1**

**INFO AppInfoParser: Kafka commitId : a7a17cdec9eaa6c5**

* The issue only existed in Meets env and not SIT env and Re-Join messages were received in SIT.

Cause:

* Found below error on debugging a simple Kafka Consumer code to read a topic:

**org.apache.kafka.common.errors.GroupCoordinatorNotAvailableException: The group coordinator is not available.**

* To debug further fired below command in kafka broker as per the below listed document :

**bin/kafka-topics.sh --describe --zookeeper 192.168.24.105:3701**

* This will give list of all topics including the \_\_consumer\_offset which had a corrupted “Leader” value as per below document. The expected Leader value should have been 1 or 3 (Broker ids) but instead it was -1 hence kafka wasn’t able to resolve the consumer offset to any topic or consumer group.

*Document:*

<https://community.microstrategy.com/s/article/Kafka-cluster-health-check-fails-with-the-error-Group-coordinator-lookup-failed-The-coordinator-is-not-available?language=en_US>

*Solution:*

* Stop Kafka and Zookeeper.
* As per above document, navigate to **dataDir=/tmp/zookeeper**  mentioned in your zoo.cfg file

and move/delete all folders except myid. In other words remover version-2 folder.

* Now navigate to KafkaMessages folder and remove everything inside it.
* Restart Kafka and Zookeeper
* Now bring up one of your consumer applications and it should be successfully able to Re-Join Kafka
* Alternatively you can fire below command again to see the Leader value against \_\_consumer\_offset topic which should now be one of the broker ids.

**bin/kafka-topics.sh --describe --zookeeper 192.168.24.105:3701**

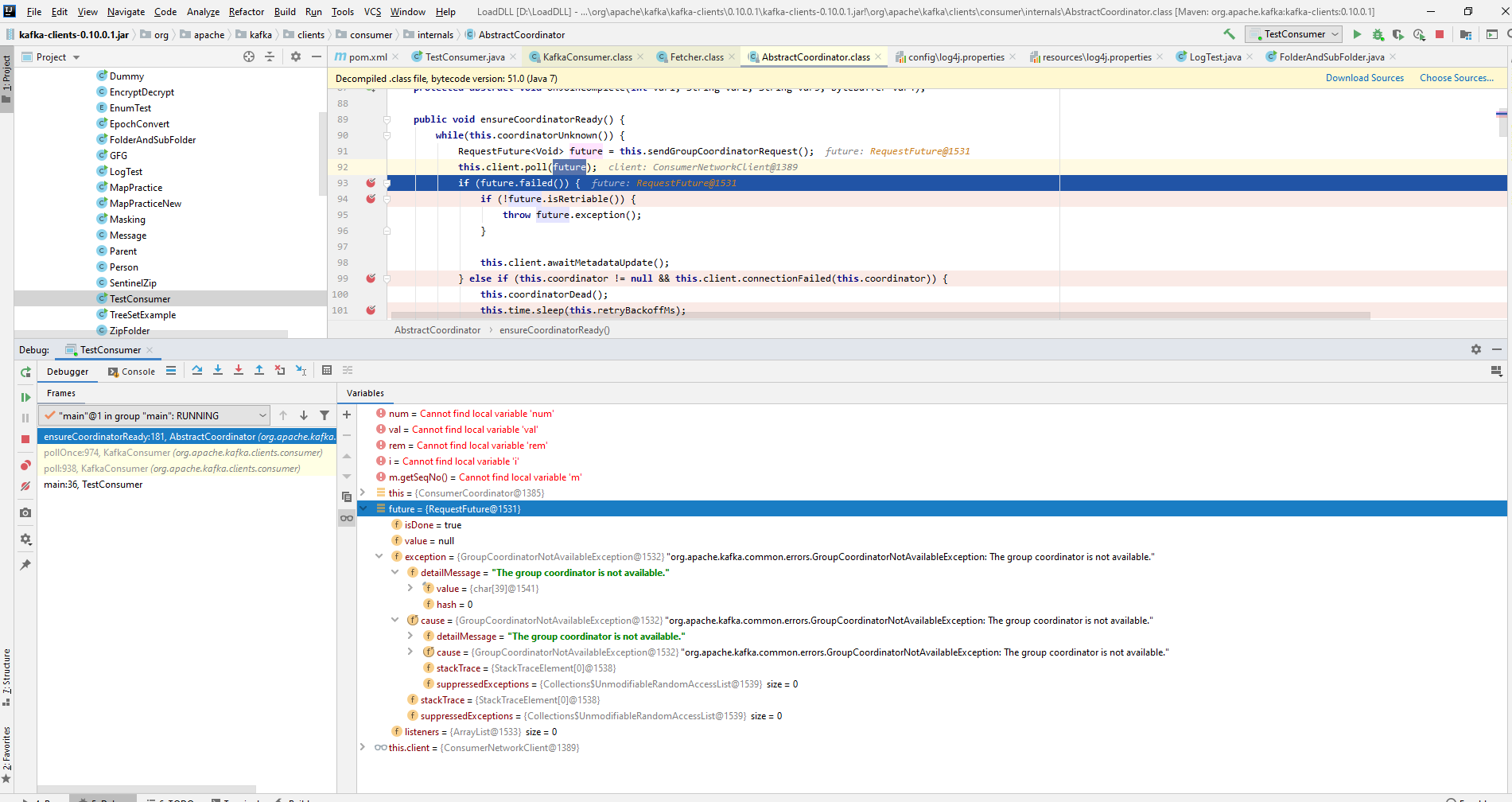
*How to Debug:*

* Wrote a simple java code to read a topic (Refer : TestConsumer.java in LoadDLL)

Place a debug point at : consumer.poll() -> inside poll() : this.pollOnce(remaining) -> inside pollOnce() : ensureCoordinatorReady () -> inside ensureCoordinatorReady : if (future.failed())

Now check the Future value in inspect element and it should give the desired error message :

**org.apache.kafka.common.errors.GroupCoordinatorNotAvailableException: The group coordinator is not available.**



Issue:

* While starting Kafka for the first time we might get an error : **“NoClassDefFoundError: Could not initialize class kafka.network.RequestChannel”** Or error : **“Consumer Marking the Coordinator 564611656 dead”**

Cause:

* Missing entry in /etc/hosts

Solution:

* Make an entry of IP and hostname in /etc/hosts (Check name of host via “hostname” cmd)
* Due to above error, offset of any topic won’t be generated for any broker (getoffsetshell cmd for any topic will fail) and consecutive all consumers will fail to connect to kafka and topics

Check memory of any directory : df –h du –sh \*

Commands to find and kill kafka process easily : pgrep –f kafka\_2 pkill –f -9 kafka\_2\*