

1. Проверьте, что вы находитесь в каталоге с Kafka, а если нет - перейдите туда:

```
cd kafka_2.13-2.7.0
```

2. Запускаем Zookeeper.

```
rm -rf /tmp/zookeeper
```

```
./bin/zookeeper-server-start.sh ./config/zookeeper.properties
```

3. Запускаем брокеры Кафки

```
cat << 'EOF' > kafka-distributed-start.sh
```

```
#!/usr/bin/env bash
```

```
for i in {0..2}
```

```
do
```

```
  cp ./config/server.properties "./config/server$i.properties"
```

```
  sed -i "s/broker.id=0/broker.id=$i/g" ./config/server$i.properties
```

```
  echo "listeners=PLAINTEXT://:909$i" >> ./config/server$i.properties
```

```
  echo "broker.rack=my-rack-$i" >> ./config/server$i.properties
```

```
  mkdir /tmp/kafka-logs$i
```

```
  sed -i "s/log.dirs=\/tmp\/kafka-logs/log.dirs=\/tmp\/kafka-logs$i/g"
```

```
./config/server$i.properties
```

```
(./bin/kafka-server-start.sh config/server$i.properties &)
```

```
done
```

```
EOF
```

```
chmod +x kafka-distributed-start.sh
```

```
./kafka-distributed-start.sh
```

4. Создаем топик

```
./bin/kafka-topics.sh --create --topic registrations --partitions 3
```

```
--bootstrap-server localhost:9092 --replication-factor 3
```

5. Проверяем, что топик создан

```
./bin/kafka-topics.sh --describe --topic registrations --bootstrap-server
```

```
localhost:9092
```

6. Остановим один брокер

```
kill -s ${SIGNAL:-TERM} $(ps ax | grep -i 'kafka\.Kafka' | grep
```

```
'server0.properties' | grep java | grep -v grep | awk '{print $1}' | head -n1)
```

7. Пробуем создать еще один топик, и получим ошибку, связанную со слишком

высоким фактором репликации

```
./bin/kafka-topics.sh --create --topic stats --partitions 3 --bootstrap-server localhost:9092 --replication-factor 3
```

8. Пробуем создать топик снова с фактором репликации 2

```
./bin/kafka-topics.sh --create --topic stats --partitions 3 --bootstrap-server localhost:9092 --replication-factor 2
```

9. Убираем broker.rack на первом брокере

```
sed -i "/broker.rack=my-rack-0/d" ./config/server0.properties
```

10. Включаем первый брокер

```
./bin/kafka-server-start.sh ./config/server0.properties &
```

11. Пробуем создать еще один топик, и не сможем этого сделать (в логах брокеров можно будет найти ошибку kafka.admin.AdminOperationException: Not all brokers have rack information for replica rack aware assignment)

```
./bin/kafka-topics.sh --create --topic metrics --partitions 3 --bootstrap-server localhost:9092 --replication-factor 3
```

12. Пробуем вернуть конфиг broker.rack динамически (без рестарта брокера), и получим ошибку Cannot update these configs dynamically: Set(broker.rack)

```
./bin/kafka-configs.sh --bootstrap-server localhost:9092 --entity-type brokers --entity-name 0 --alter --add-config broker.rack=my-rack-0
```

13. Возвращаем конфигурацию broker.rack на первый брокер

```
echo broker.rack=my-rack-0 >> ./config/server0.properties
```

14. Перезагружаем первый брокер чтобы конфигурация обновилась

```
kill -s ${SIGNAL:-TERM} $(ps ax | grep -i 'kafka\.Kafka' | grep 'server0.properties' | grep java | grep -v grep | awk '{print $1}' | head -n1) && ./bin/kafka-server-start.sh ./config/server0.properties &
```

15. Пересоздаем топик (успех :)

```
./bin/kafka-topics.sh --create --topic metrics --partitions 3 --bootstrap-server localhost:9092 --replication-factor 3
```

16. Удаляем топик. После останавливаем брокеры, и потом Zookeeper.

```
./bin/kafka-topics.sh --delete --topic metrics --bootstrap-server localhost:9092
```

```
kill -s ${SIGNAL:-TERM} $(ps ax | grep -i 'kafka\.Kafka' | grep  
'server[0-2].properties' | grep java | grep -v grep | awk '{print $1}' | head  
-n1)
```

```
./bin/zookeeper-server-stop.sh config/zookeeper.properties
```