

# kafka学习笔记-learning1

## 环境安装（Linux上的安装使用）

### 环境及软件版本

- LINUX : CentSO-8.2.2004-x86\_64

- JDK : 11.0.8

- Zookeeper : 3.6.2

下载地址: <https://mirror.bit.edu.cn/apache/zookeeper/zookeeper-3.6.2/apache-zookeeper-3.6.2-bin.tar.gz>

- kafka : 2.13-2.6.0

下载地址: [https://mirror.bit.edu.cn/apache/kafka/2.6.0/kafka\\_2.12-2.5.0.tgz](https://mirror.bit.edu.cn/apache/kafka/2.6.0/kafka_2.12-2.5.0.tgz)

### 软件安装

- Java 安装

```
yum install java-11-openjdk-devel.x86_64
```

#### 检查安装情况

```
[solomon@localhost kafka_2.13-2.6.0]$ java -version
openjdk version "11.0.8" 2020-07-14 LTS
OpenJDK Runtime Environment 18.9 (build 11.0.8+10-LTS)
OpenJDK 64-Bit Server VM 18.9 (build 11.0.8+10-LTS, mixed mode, sharing)
[solomon@localhost kafka_2.13-2.6.0]$
```

- 安装zookeeper

到安装包目录下执行命令

```
tar -zxvf apache-zookeeper-3.6.2-bin.tar.gz
```

#### zookeeper 配置

```
cp zoo_sample.cfg zoo.cfg
vi zoo.cfg
```

修改dataDir, dataLogDir

```
1 192.168.159.130 x +
# The number of milliseconds of each tick
tickTime=2000
# The number of ticks that the initial
# synchronization phase can take
initLimit=10
# The number of ticks that can pass between
# sending a request and getting an acknowledgement
syncLimit=5
# the directory where the snapshot is stored.
# do not use /tmp for storage, /tmp here is just
# example sakes.
dataDir=/home/solomon/Documents/apache-zookeeper-3.6.2-bin/data
dataLogDir=/home/solomon/Documents/apache-zookeeper-3.6.2-bin/log
# the port at which the clients will connect
clientPort=2181
# the maximum number of client connections.
# increase this if you need to handle more clients
#maxClientCnxns=60
#
# Be sure to read the maintenance section of the
# administrator guide before turning on autopurge.
#
# http://zookeeper.apache.org/doc/current/zookeeperAdmin.html#sc_maintenance
#
# The number of snapshots to retain in dataDir
#autopurge.snapRetainCount=3
# Purge task interval in hours
# Set to "0" to disable auto purge feature
#autopurge.purgeInterval=1

## Metrics Providers
#
# https://prometheus.io Metrics Exporter
```

1,1 Top

## 启动命令

```
./zkServer.sh start
```

## 检查安装命令

```
./zkServer.sh status
```

## 运行结果如下

```
solomon@localhost apache-zookeeper-3.6.2-bin]$ cd bin
solomon@localhost bin]$ ./zkServer.sh status
/usr/bin/java
ZooKeeper JMX enabled by default
Using config: /home/solomon/Documents/apache-zookeeper-3.6.2-bin/bin/../conf/zoo.cfg
Client port found: 2181. Client address: localhost. Client SSL: false.
Mode: standalone
solomon@localhost bin]$
```

## • 安装Kafka

### 到安装包目录下执行命令

```
tar -xvf kafka_2.13-2.6.0.tgz
```

## kafka配置文件

```
cd config
vi server.properties
```

## 修改如下

```
broker.id=1 #三台机器的id分别为1, 2, 3
listeners=PLAINTEXT://192.168.56.130:9092 #设置为每台机器各自的IP
log.dirs=/tmp/kafka-logs
offsets.topic.replication.factor=3
transaction.state.log.replication.factor=3
transaction.state.log.min.isr=1
zookeeper.connect=localhost:2181
```

```
1 192.168.159.130 x +
# 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 = listener_name://host_name:port
# EXAMPLE:
# listeners = PLAINTEXT://your.host.name:9092
listeners=PLAINTEXT://192.168.159.130:9092

# Hostname and port the broker will advertise to producers and consumers. If not set,
# it uses the value for "listeners" if configured. Otherwise, it will use the value
# returned from java.net.InetAddress.getCanonicalHostName().
#advertised.listeners=PLAINTEXT://your.host.name:9092

# Maps listener names to security protocols, the default is for them to be the same. See the config documentation for more details
#listener.security.protocol.map=PLAINTEXT:PLAINTEXT,SSL:SSL,SASL_PLAINTEXT:SASL_PLAINTEXT,SASL_SSL:SASL_SSL

# The number of threads that the server uses for receiving requests from the network and sending responses to the network
num.network.threads=3

# The number of threads that the server uses for processing requests, which may include disk I/O
num.io.threads=8
```

```
1 192.168.159.130 x +
# be set to delete segments after a period of time, or after a given size has accumulated.
# A segment will be deleted whenever *either* of these criteria are met. Deletion always happens
# from the end of the log.

# The minimum age of a log file to be eligible for deletion due to age
log.retention.hours=168

# A size-based retention policy for logs. Segments are pruned from the log unless the remaining
# segments drop below log.retention.bytes. Functions independently of log.retention.hours.
#log.retention.bytes=1073741824

# The maximum size of a log segment file. When this size is reached a new log segment will be created.
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 #####

# Zookeeper connection string (see zookeeper docs for details).
# This is a comma separated host:port pairs, each corresponding to a zk
# server. e.g. "127.0.0.1:3000,127.0.0.1:3001,127.0.0.1:3002".
# You can also append an optional chroot string to the urls to specify the
# root directory for all kafka znodes.
zookeeper.connect=localhost:2181

# Timeout in ms for connecting to zookeeper
zookeeper.connection.timeout.ms=18000

##### Group Coordinator Settings #####
```

125,1

94%

启动  
进入安装目录

```
./bin/kafka-server-start.sh -daemon config/server.properties
```

## 查看状态

```
[3]+ Stopped vim server.properties
[solomon@localhost config]$ jps
13447 Jps
3337 QuorumPeerMain
3497 Kafka
[solomon@localhost config]$
```

### • Kafka 常用命令工具

#### ◦ 创建Topic

创建一个Topic，使用一个partiton

```
bin/kafka-topics.sh --bootstrap-server 192.168.159.130:9092 --create --topic myTopic
```

```
[solomon@localhost kafka_2.13-2.6.0]$ bin/kafka-topics.sh --bootstrap-server 192.168.159.130:9092 --create --topic myTopic
Created topic myTopic.
[solomon@localhost kafka_2.13-2.6.0]$
```

#### ◦ 查看创建的Topic

```
bin/kafka-topics.sh --bootstrap-server 192.168.159.130:9092 --describe --topic myTopic
```

```
[solomon@localhost kafka_2.13-2.6.0]$ bin/kafka-topics.sh --bootstrap-server 192.168.159.130:9092 --describe --topic myTopic
Topic: myTopic PartitionCount: 1 ReplicationFactor: 1 Configs: segment.bytes=1073741824
Topic: myTopic Partition: 0 Leader: 1 Replicas: 1 Isr: 1
[solomon@localhost kafka_2.13-2.6.0]$
```

#### ◦ 生产者

发送消息进入消息模式

可以向队列输入消息

```
bin/kafka-console-producer.sh --bootstrap-server 192.168.159.130:9092 --topic myTopic
```

```
Topic: myTopic Partition: 0 Leader: 1 Replicas: 1 Isr: 1
[solomon@localhost kafka_2.13-2.6.0]$ bin/kafka-console-producer.sh --bootstrap-server 192.168.159.130:9092 --topic myTopic
>hello pip
>
```

#### ◦ 消费者

接收队列里的消息

```
bin/kafka-console-consumer.sh --bootstrap-server 192.168.159.130:9092 --topic myTopic
```

```
bin config tools LICENSE logs nonprod NOTICE ZLIC 6069
[solomon@localhost kafka_2.13-2.6.0]$ bin/kafka-console-consumer.sh --bootstrap-server 192.168.159.130:9092 --topic myTopic
hello pip
```