# 一 . 介绍

## Zookeeper介绍

### 1.概念

Zookeeper:分布式的一致性协调服务；

集群的管理者，监视着集群中各个节点的状态根据节点提交的反馈进行下一步合理操作；

多层级的节点命名空间，树状目录，这些节点都可以设置关联的数据，不能用于存放大量的数据，每个节点存放数据上限为1M；

**四种类型的节点（znode）：**

* 持久化目录节点：客户端与zookeeper断开连接后，该节点依旧存在
* 持久化顺序编号目录节点：客户端与zookeeper断开连接后，该节点依旧存在，只是Zookeeper给该节点名称进行顺序编号
* 临时目录节点：客户端与zookeeper断开连接后，该节点被删除
* 临时顺序编号目录节点：客户端与zookeeper断开连接后，该节点被删除，只是Zookeeper给该节点名称进行顺序编号

### 2. Zookeeper的通知机制

client端会对某个znode建立一个**watcher事件**，当该znode发生变化时，这些client会收到zk的通知，然后client可以根据znode变化来做出业务上的改变等。

### 3. zookeeper的功能

1、命名服务（文件系统）

命名服务是指通过指定的名字来**获取资源**或者**服务的地址**，利用zk创建一个全局的路径，即是**唯一**的路径，这个路径就可以作为一个名字，指向集群中的集群，提供的服务的地址，或者一个远程的对象等等。  
2、配置管理（文件系统、通知机制）

程序分布式的部署在不同的机器上，将程序的配置信息放在zk的**znode**下，当有配置发生改变时，也就是znode发生变化时，可以通过改变zk中某个目录节点的内容，利用**watcher**通知给各个客户端，从而更改配置。  
3、集群管理（文件系统、通知机制）

所谓集群管理无在乎两点：**是否有机器退出和加入、选举master**。

对于第一点，所有机器约定在父目录下**创建临时目录节点**，然后监听父目录节点的子节点变化消息。一旦有机器挂掉，该机器与 zookeeper的连接断开，其所创建的临时目录节点被删除，**所有其他机器都收到通知：某个兄弟目录被删除**。  
新机器加入也是类似，**所有机器收到通知：新兄弟目录加入**，对于第二点，我们稍微改变一下，**所有机器创建临时顺序编号目录节点，每次选取编号最小的机器作为master就好**。  
4、分布式锁（文件系统、通知机制）

5、队列管理（文件系统、通知机制）

### 4.Zookeeper数据复制

Zookeeper作为一个集群提供一致的数据服务，自然，它要在**所有机器间**做数据复制。数据复制的好处：  
1、容错：一个节点出错，不致于让整个系统停止工作，别的节点可以接管它的工作；  
2、提高系统的扩展能力 ：把负载分布到多个节点上，或者增加节点来提高系统的负载能力；  
3、提高性能：让**客户端本地访问就近的节点，提高用户访问速度**。

从客户端读写访问的透明度来看，数据复制集群系统分下面两种：  
1、*写主*(WriteMaster) ：对数据的*修改提交给指定的节点*。读无此限制，可以读取任何一个节点。这种情况下客户端需要对读与写进行区别，俗称*读写分离*；  
2、*写任意*(Write Any)：对数据的*修改可提交给任意的节点*，跟读一样。这种情况下，客户端对集群节点的角色与变化透明。

**对zookeeper来说，它采用的方式是*写任意*。**通过增加机器，它的读吞吐能力和响应能力扩展性非常好，而写，随着机器的增多吞吐能力肯定下降（这也是它建立observer的原因），而响应能力则取决于具体实现方式，是*延迟复制保持最终一致性*，还是*立即复制快速响应*。

## Kafka介绍

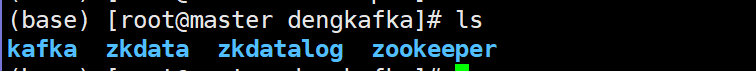
Kafka：作为消息中间件，一般部署在流式组件前一个，主要为了避免高峰期计算来的压力；

# 二．Zookeeper集群安装

规划三台虚拟机：

* 192.168.101.118
* 192.168.101.119
* 192.168.101.120

三台主机都装zookeeper，kafka；



建立了四个目录，放置kafka，zookeeper数据目录，zookeeper日志目录，zookeeper安装目录；

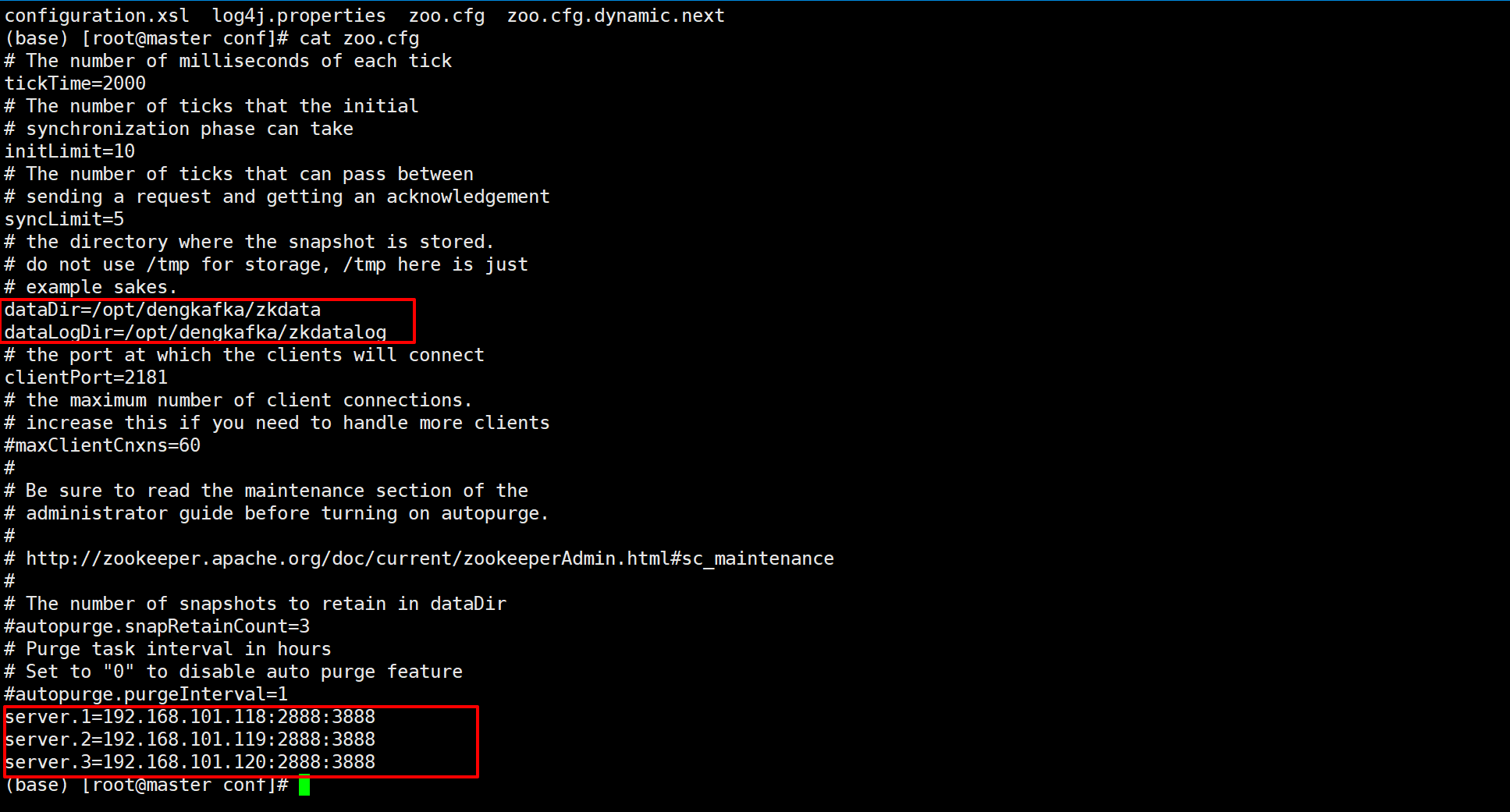
解压zookeeper的tar包如下：（注意：下载3.5.5以上的版本需要下载带bin的，不然其中没有lib目录，zookeeper无法启动）



进入conf目录：

mv zoo\_sample.cfg zoo.cfg

修改zoo.cfg文件，如下红色标注所示：



|  |
| --- |
| #第一台主机  echo "1" >/opt/dengkafka/zkdata/myid *##生成ID，这里需要注意， myid对应的zoo.cfg的server.ID，比如第二台zookeeper主机对应的myid应该是2*  #第二台主机  echo "2" >/opt/dengkafka/zkdata/myid  #第三胎主机  echo "3" >/opt/dengkafka/zkdata/myid |

**启动zookeeper：**

cd /opt/dengkafka/zookeeper/apache-zookeeper-3.5.6-bin/bin

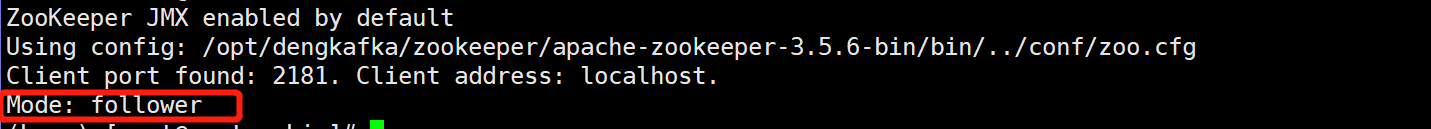
./zkServer.sh start

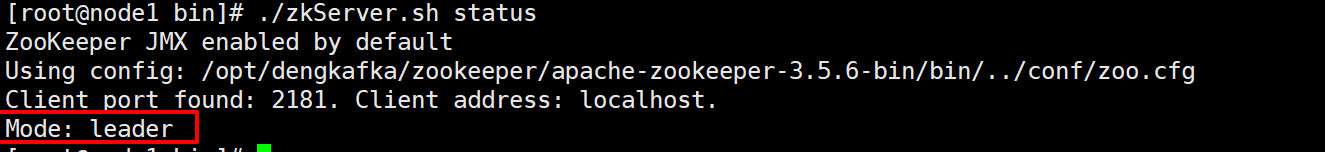
**关闭zookeeper：**

./zkServer.sh stop

查看zookeeper状态：

./zkServer.sh status

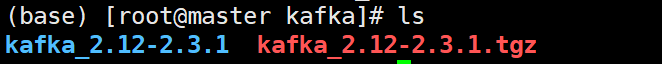




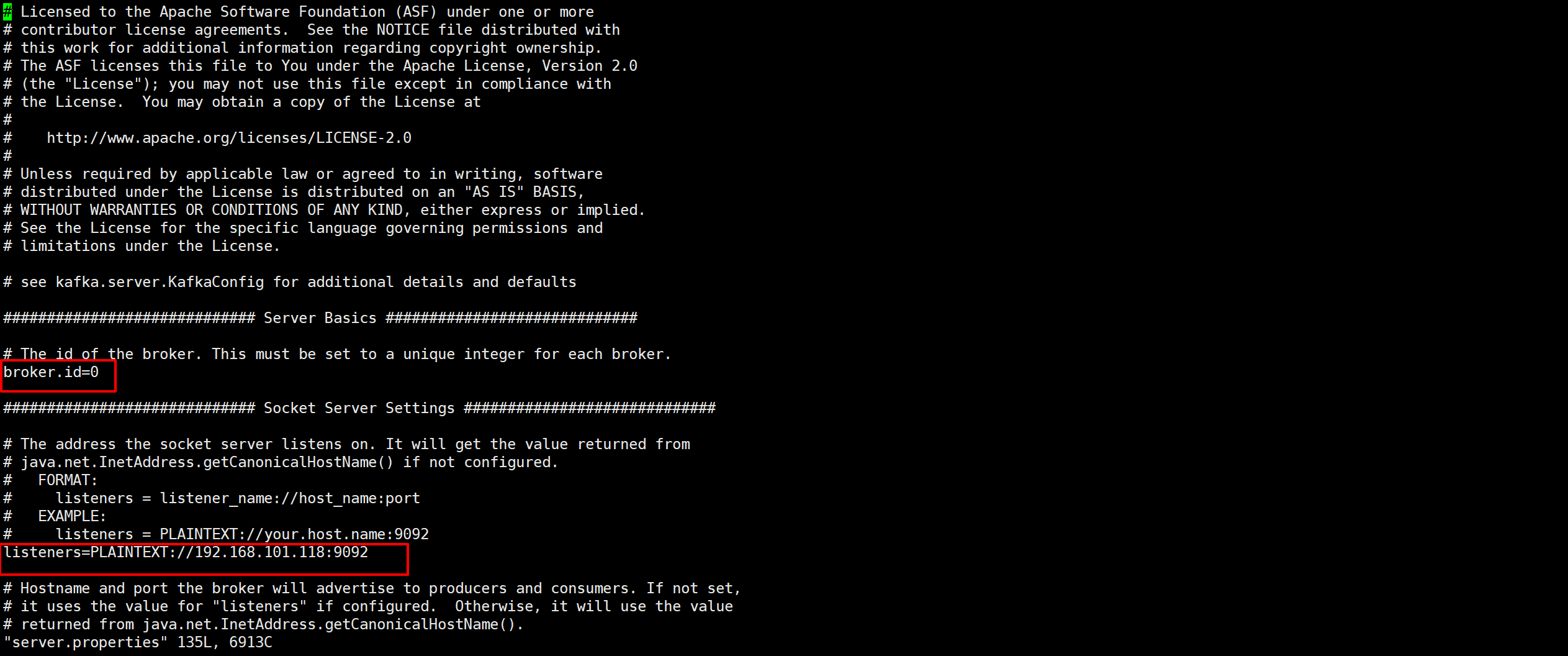
# 三．Kafka集群安装

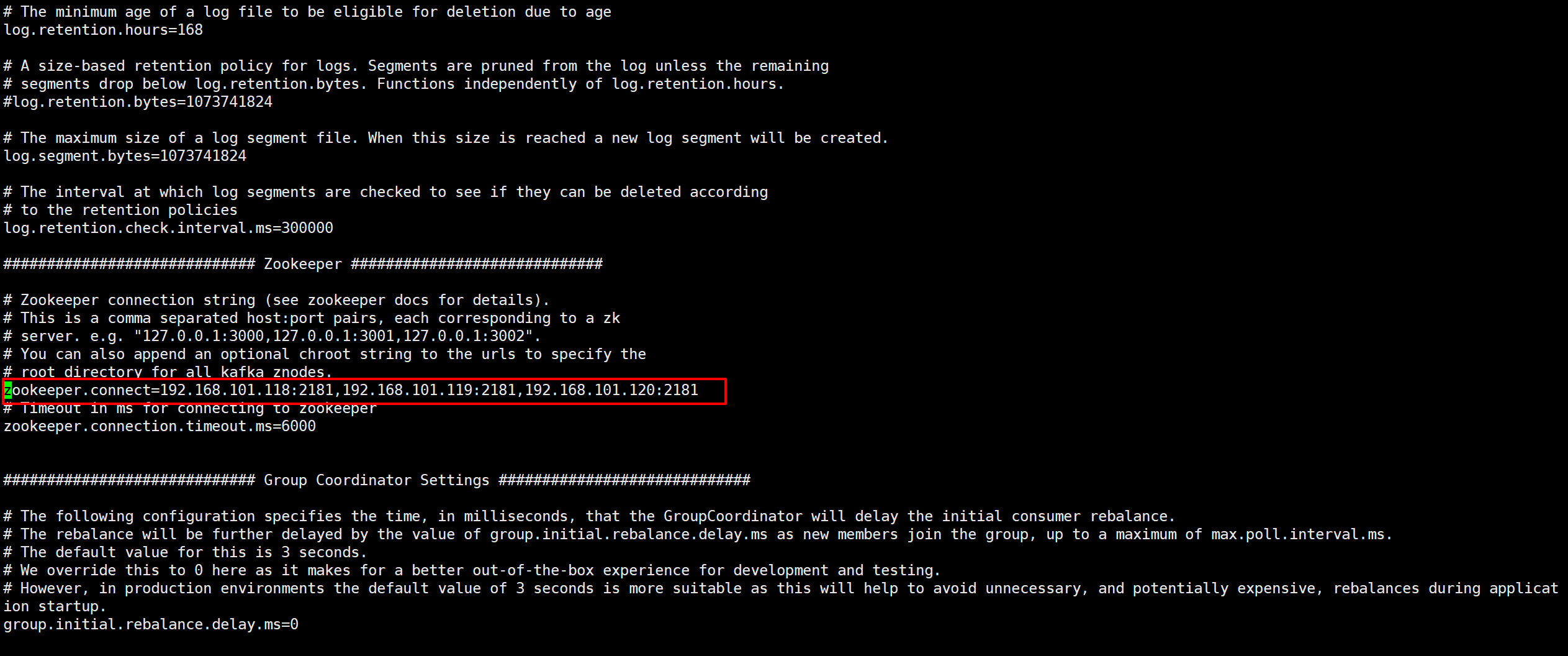
## 安装kafka

解压kafka的压缩包即可；



## 修改配置文件





## 启动kafka

|  |
| --- |
| ./kafka-server-start.sh -daemon /opt/dengkafka/kafka/kafka\_2.12-2.3.1/config/server.properties & |

## 创建topic

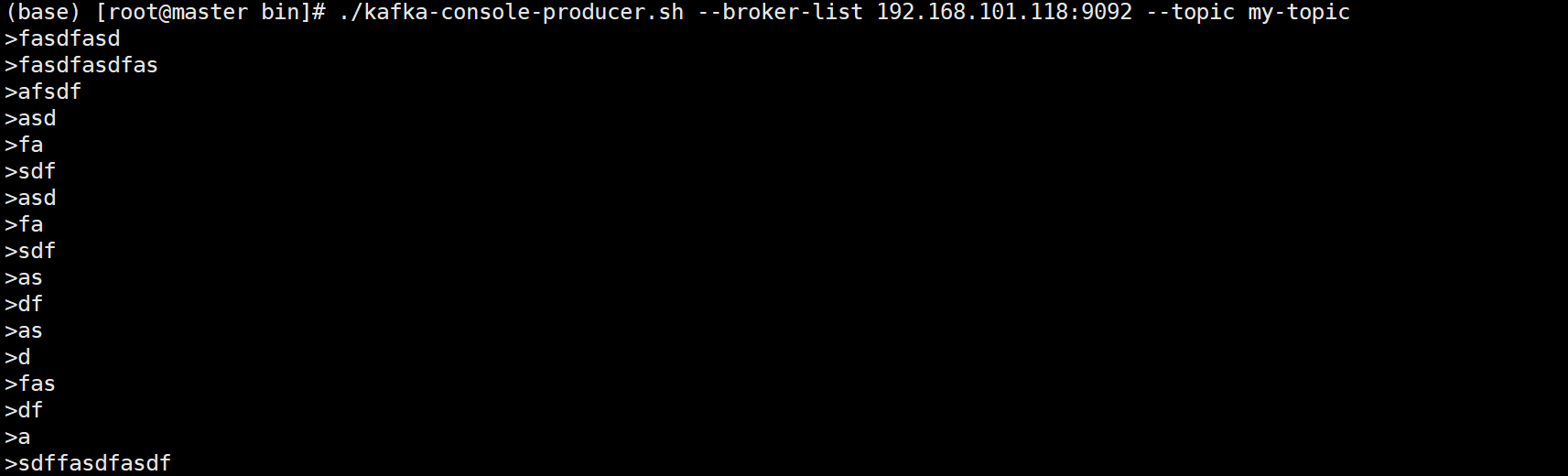
|  |
| --- |
| ./kafka-topics.sh --create --zookeeper 192.168.101.118:2181,192.168.101.119:2181,192.168.101.120:2181 --replication-factor 2 --partitions 1 --topic my-topic |

## 生产消息

|  |
| --- |
| ./kafka-console-producer.sh --broker-list 192.168.101.118:9092 --topic my-topic |

会进入阻塞模式，可以进行输入数据

如下：



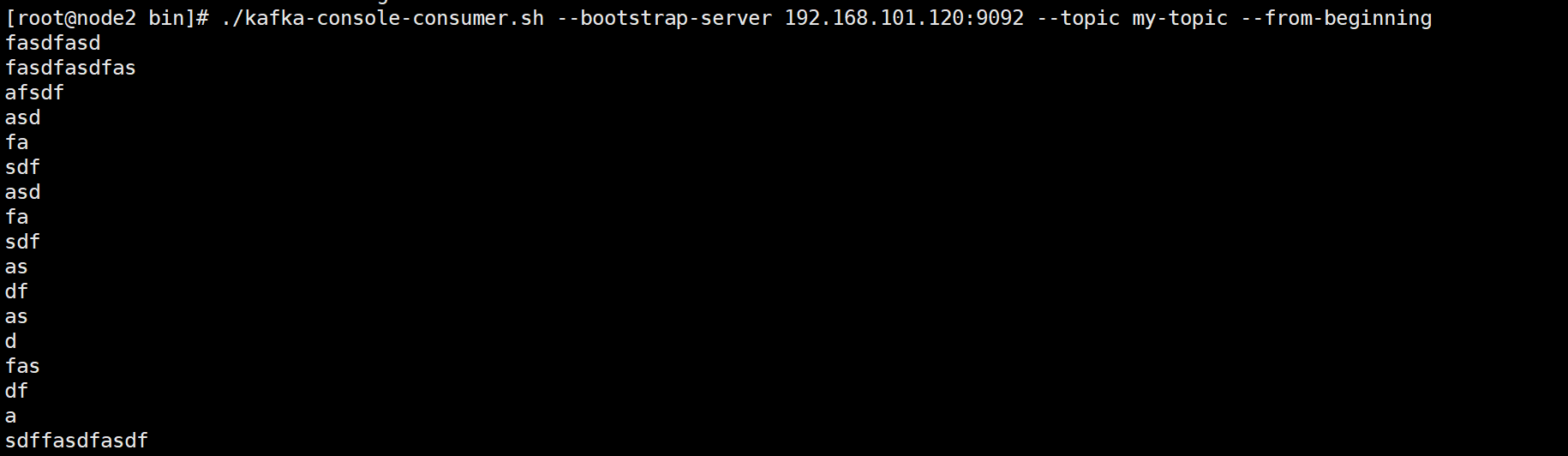
## 消费消息

|  |
| --- |
| ./kafka-console-consumer.sh --bootstrap-server 192.168.101.118:9092 --topic my-topic --from-beginning |

生产消息后，这里会收到消息；--bootstrap-server 192.168.101.118:9092这个可以配置对应的监听主机ip；

如下：



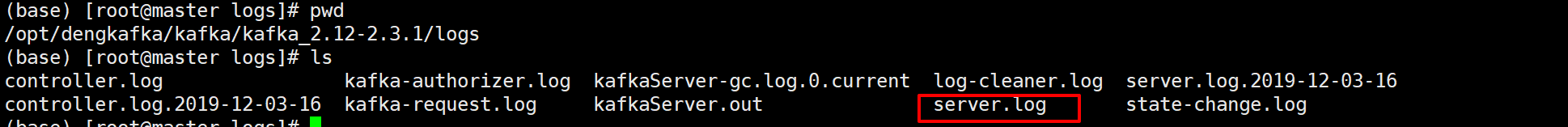


## 查看topic

|  |
| --- |
| ./kafka-topics.sh --list --zookeeper 192.168.101.118:2181,192.168.101.119:2181,192.168.101.120:2181 |



## 查看日志



# 四．Kafka测试

批量杀进程：

