1. 安装依赖的java环境

安装包版本：jdk-8u25-linux-x64.tar.gz

安装步骤：

|  |
| --- |
| #tar -zxvf jdk-8u25-linux-x64.tar.gz  # cd jdk1.8.0\_25/  # ll  total 28812  drwxr-xr-x 2 10 143 4096 Sep 18 2014 bin  -r--r--r-- 1 10 143 3244 Sep 18 2014 COPYRIGHT  drwxr-xr-x 4 10 143 115 Sep 18 2014 db  drwxr-xr-x 3 10 143 125 Sep 18 2014 include  -rw-r--r-- 1 10 143 5025522 Sep 17 2014 javafx-src.zip  drwxr-xr-x 5 10 143 4096 Sep 18 2014 jre  drwxr-xr-x 5 10 143 4096 Sep 18 2014 lib  -r--r--r-- 1 10 143 40 Sep 18 2014 LICENSE  drwxr-xr-x 4 10 143 44 Sep 18 2014 man  -r--r--r-- 1 10 143 159 Sep 18 2014 README.html  -rw-r--r-- 1 10 143 525 Sep 18 2014 release  -rw-r--r-- 1 10 143 21057306 Sep 18 2014 src.zip  -rw-r--r-- 1 10 143 110114 Sep 17 2014 THIRDPARTYLICENSEREADME-JAVAFX.txt  -r--r--r-- 1 10 143 178400 Sep 18 2014 THIRDPARTYLICENSEREADME.txt  #mkdir –p /app/jdk  #cp -r ../jdk1.8.0\_25 /app/jdk  #vim /etc/profile  在最后插入如下几行：  export JAVA\_HOME=/app/jdk/jdk1.8.0\_25  export PATH=$JAVA\_HOME/bin:$JAVA\_HOME/jre/bin:$PATH:$HOME/bin  export CLASSPATH=.:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/jre/lib/tools.jar  # source /etc/profile  # java -version  java version "1.8.0\_25"  Java(TM) SE Runtime Environment (build 1.8.0\_25-b17)  Java HotSpot(TM) 64-Bit Server VM (build 25.25-b02, mixed mode)  #javac  Usage: javac <options> <source files>  where possible options include:  -g Generate all debugging info  -g:none Generate no debugging info  -g:{lines,vars,source} Generate only some debugging info  -nowarn Generate no warnings  -verbose Output messages about what the compiler is doing  -deprecation Output source locations where deprecated APIs are used  -classpath <path> Specify where to find user class files and annotation processors  -cp <path> Specify where to find user class files and annotation processors  -sourcepath <path> Specify where to find input source files  -bootclasspath <path> Override location of bootstrap class files  -extdirs <dirs> Override location of installed extensions  -endorseddirs <dirs> Override location of endorsed standards path  -proc:{none,only} Control whether annotation processing and/or compilation is done.  -processor <class1>[,<class2>,<class3>...] Names of the annotation processors to run; bypasses default discovery process  -processorpath <path> Specify where to find annotation processors  -parameters Generate metadata for reflection on method parameters  -d <directory> Specify where to place generated class files  -s <directory> Specify where to place generated source files  -h <directory> Specify where to place generated native header files  -implicit:{none,class} Specify whether or not to generate class files for implicitly referenced files  -encoding <encoding> Specify character encoding used by source files  -source <release> Provide source compatibility with specified release  -target <release> Generate class files for specific VM version  -profile <profile> Check that API used is available in the specified profile  -version Version information  -help Print a synopsis of standard options  -Akey[=value] Options to pass to annotation processors  -X Print a synopsis of nonstandard options  -J<flag> Pass <flag> directly to the runtime system  -Werror Terminate compilation if warnings occur  @<filename> Read options and filenames from file |

配置成功。

1. zookeeper安装配置

集群部署要求：至少三个节点以上且必须为奇数，每个节点必须部署在不同的机器；

集群环境安装：在三台server上搭建zookeeper集群

当前稳定版本：zookeeper-3.4.6.tar.gz

安装步骤：

|  |
| --- |
| # tar -zxvf zookeeper-3.4.6.tar.gz  # cd zookeeper-3.4.6/  # ll  total 1532  drwxr-xr-x 2 1000 1000 141 Feb 20 2014 bin  -rw-rw-r-- 1 1000 1000 82446 Feb 20 2014 build.xml  -rw-rw-r-- 1 1000 1000 80776 Feb 20 2014 CHANGES.txt  drwxr-xr-x 2 1000 1000 74 Feb 20 2014 conf  drwxr-xr-x 10 1000 1000 122 Feb 20 2014 contrib  drwxr-xr-x 2 1000 1000 4096 Feb 20 2014 dist-maven  drwxr-xr-x 6 1000 1000 4096 Feb 20 2014 docs  -rw-rw-r-- 1 1000 1000 1953 Feb 20 2014 ivysettings.xml  -rw-rw-r-- 1 1000 1000 3375 Feb 20 2014 ivy.xml  drwxr-xr-x 4 1000 1000 4096 Feb 20 2014 lib  -rw-rw-r-- 1 1000 1000 11358 Feb 20 2014 LICENSE.txt  -rw-rw-r-- 1 1000 1000 170 Feb 20 2014 NOTICE.txt  -rw-rw-r-- 1 1000 1000 1770 Feb 20 2014 README\_packaging.txt  -rw-rw-r-- 1 1000 1000 1585 Feb 20 2014 README.txt  drwxr-xr-x 5 1000 1000 44 Feb 20 2014 recipes  drwxr-xr-x 8 1000 1000 145 Feb 20 2014 src  -rw-rw-r-- 1 1000 1000 1340305 Feb 20 2014 zookeeper-3.4.6.jar  -rw-rw-r-- 1 1000 1000 836 Feb 20 2014 zookeeper-3.4.6.jar.asc  -rw-rw-r-- 1 1000 1000 33 Feb 20 2014 zookeeper-3.4.6.jar.md5  -rw-rw-r-- 1 1000 1000 41 Feb 20 2014 zookeeper-3.4.6.jar.sha1  #mkdir -p /app/zookeeper/  #cp -r \* /app/zookeeper/  #mkdir -p /app/zookeeper/data  #mkdir -p /app/zookeeper/logs  #cd /app/zookeeper/conf/  #cp zoo\_sample.cfg zoo.cfg  #vim zoo.cfg  修改配置参数，没有就添加：  tickTime=2000  dataDir=/app/zookeeper/data  logDir=/app/zookeeper/logs  maxClientCnxns=60  clientPort=2181  maxClientCnxns=60  syncLimit=5  initLimit=10  autopurge.snapRetainCount=3  autopurge.purgeInterval=1  server.1=192.168.233.83:2888:3888  server.2=192.168.233.84:2888:3888  server.3=192.168.233.85:2888:3888 |

参数说明：

* clientPort zookeeper 监听端口
* dataDir 数据目录
* server.X=A:B:C 其中X是代表第几号Server,A为该Server 所在IP地址，B为Server间数据交换端口，C为leader 选举端口
* autopurge.snapRetainCount 保存快照或日志文件数
* autopurge.purgeInterval 清除数据时间间隔 单位小时
* maxClientCnxns单个客户端与单台服务器之间的连接数的限制，是ip级别的，默认是60，如果设置为0，
* initLimit Follower在启动过程中，会从Leader同步所有最新数据，然后确定自己能够对外服务的起始状态。Leader允许F在 initLimit时间内完成这个工作。通常情况下，我们不用太在意这个参数的设置。如果ZK集群的数据量确实很大了，F在启动的时候，从Leader上同步数据的时间也会相应变长，因此在这种情况下，有必要适当调大这个参数了。
* syncLimit在运行过程中，Leader负责与ZK集群中所有机器进行通信，例如通过一些心跳检测机制，来检测机器的存活状态。如果L发出心跳包在syncLimit之后，还没有从F那里收到响应，那么就认为这个F已经不在线了。注意：不要把这个参数设置得过大，否则可能会掩盖一些问题。

在该Server 的data 目录新增myid文件，文件内容为相应ID,该ID 就是在zoo.cfg 中配置的X值,如按当前zoo.cfg 配置,设置myid 分别在zoo1、zoo2和zoo3执行

|  |
| --- |
| echo "1" > /app/zookeeper/data/myid  echo "2" >/app/zookeeper/data/myid  echo "3" > /app/zookeeper/data/myid |

启动停止Server

不使用monit 监控启动bin/zkServer.sh start

不使用monit 监控停止bin/zkServer.sh stop

见monit 监控说明,建议通过monit 监控启动与停止

目录说明：

|  |  |
| --- | --- |
| **目录** | **功能** |
| /app/zookeeper | zookeeper程序的安装与运行目录 |
| /app/tmp | 临时目录，用于存放上传到服务器的安装程序等 |
| /app/ zookeeper/conf | zookeeper配置文件存放目录 |
| /app/ zookeeper/lib | zookeeper依赖的lib包存放目录 |
| /app/ zookeeper/contrib | zookeeper |
| /app/ zookeeper/bin | zookeeper命令安装目录 |
| /app/zookeeper/data | Zookeeper存储快照文件snapshot的目录 |
| /app/zookeeper/logs | Zookeeper运行日志存放目录 |

ZooKeeper服务命令

•1. 启动ZK服务:       sh bin/zkServer.sh start  
•2. 查看ZK服务状态:   sh bin/zkServer.sh status  
•3. 停止ZK服务:       sh bin/zkServer.sh stop  
•4. 重启ZK服务:       sh bin/zkServer.sh restart

zk客户端命令

•1. 显示根目录下、文件： ls / 使用 ls 命令来查看当前 ZooKeeper 中所包含的内容

•2. 显示根目录下、文件： ls2 / 查看当前节点数据并能看到更新次数等数据  
•3. 创建文件，并设置初始内容： create /zk "test" 创建一个新的 znode节点“ zk ”以及与它关联的字符串  
•4. 获取文件内容： get /zk 确认 znode 是否包含我们所创建的字符串  
•5. 修改文件内容： set /zk "zkbak" 对 zk 所关联的字符串进行设置  
•6. 删除文件： delete /zk 将刚才创建的 znode 删除  
•7. 退出客户端： quit  
•8. 帮助命令： help

1. kafka集群安装配置

集群部署要求:3个节点，每个节点必须部署在不同机器.

下载当前kafka 最新稳定版本kafka\_2.9.2-0.8.1.1.tgz(目前升级到kafka\_2.12-2.0.0.tgz)

|  |
| --- |
| #tar -zxvf kafka\_2.12-2.0.0.tgz  # cd kafka\_2.12-2.0.0/  # mkdir -p /app/kafka  # cp -r \* /app/kafka/  # cd /app/kafka/config/  #mkdir –p /app/kafka/logs  # vim server.properties  修改配置参数：  broker.id=0 (每个主机不一致)  listeners=PLAINTEXT://10.202.233.83:9092  prot=9092  advertised.listeners=PLAINTEXT://10.202.233.83:9092  host.name=10.202.233.83  num.network.threads=2  num.io.threads=8  socket.send.buffer.bytes=1048576  socket.receive.buffer.bytes=1048576  socket.request.max.bytes=104857600  log.dirs=/app/kafka/logs  num.partitions=2  log.retention.hours=168  log.retention.bytes=1073741824  log.segment.bytes=536870912  log.retention.check.interval.ms=60000  log.cleaner.enable=false  auto.leader.rebalance.enable=true  auto.create.topics.enable=false  zookeeper.connect=192.168.233.83:2181, 192.168.233.84:2181, 192.168.233.85:2181  zookeeper.connection.timeout.ms=1000000 |

参数说明：

* broker.id 唯一ID 为非负数字，用于区分broker节点
* port kafka Server监听端口
* host.name 服务器IP
* log.dirs kafka 数据文件存储路径
* zookeeper.connect zookeeper 地址，尾部/kafka/ta用于多kafka 集群共享zookeeper，用该路径区分不同kafka 集群
* auto.leader.rebalance.enable true自动平衡主节点
* auto.create.topics.enable false 不允许在未创建topic 之前写入消息

启动停止server

1. 不使用monit 监控启动：

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

#sh /app/kafka/bin/kafka-server-start.sh -daemon /app/kafka/config/server.properties

不使用monit 监控启动：./bin/kafka-server-stop.sh

**monit 监控说明,建议通过monit 监控启动与停止kafka**

目录结构说明

|  |  |
| --- | --- |
| **目录** | **功能** |
| /app/kafka | kafka程序的安装与运行目录 |
| /app/tmp | 临时目录，用于存放上传到服务器的安装程序等 |
| /app/kafka/conf | Kafka配置文件存放目录 |
| /app/kafka/libs | Kafka 依赖的lib包存放目录 |
| /app/kafka/logs | kafka日志文件存放目录 |
| /app/kafka/bin | Kafka命令安装目录 |
| /app/kafka/data\_logs | Kafka数据文件存放目录 |

创建和查看主题命令

bin/kafka-topics.sh --create --zookeeper zoo1:2181 --replication-factor 3 --partitions 1 --topic mytest

>bin/kafka-topics.sh --list --zookeeper zoo1:2181

>bin/kafka-topics.sh --describe --zookeeper zoo1:2181 --topic page\_ mytest

> /app/kafka/bin/kafka-topics.sh --zookeeper 172.30.226.204:2181 --alter --topic sfo-statsd-info --partitions 12 拆分partition