Zookeeper是一个分布式服务框架，用于分布式应用程序的高性能服务协调，在很多分布式服务场景中都会用到，例如：配置管理、命名、同步等等。Zookeeper做为分布式应用一致性协调服务的中间件，让我们一起来使用下它吧。

Zookeeper下边简称zk

一、zk的文件系统、会话和监听机制

1、文件系统

zk使用文件系统目录树作为数据模型，每一个子目录项都是一个znode，znode都可以具有关联的数据（存储上限为1M）和子节点（子节点和数据可以并存），结构如下图：



znode可以分为四种：持久节点、持久顺序节点、临时节点、临时顺序节点

持久节点（PERSISTENT），zk与客户端断开连接后依旧存在,只有主动删除才会消失

持久顺序节点（PERSISTENT\_SEQUENTIAL）,与持久节点一样，并且节点名称后会自动增加一个10位数字序列号

临时节点（EPHEMERAL）zk与客户端断开连接时，节点会自动删除（临时节点没有子节点）

临时顺序节点（EPHEMERAL\_SEQUENTIAL）,与临时节点一样，并且节点名称后会加序列号。

2、会话(session)

zk的服务器端和客户端通过一次TCP长连接建议一个会话（session），服务器端在建立连接以后会为sessionId创建一个密码，以后的每次重连的时候需要同时传递sessionId+密码

客户端的状态有：

CONNECTING，连接中，在客户端尝试连接到zk服务

CONNECTED ，连接成功后，状态转换为CONNECTED

CLOSED,关闭，发生在客户端关闭连接、session过期、认证失败的时候

3、监听机制

watch事件是一次性的触发器，当watch监听的节点发生改变的时候，服务器将这个改变发送给设置了Watch的客户端。

监听机制是解决分布式一致性的重要基础，例如：应用配置、分布式锁的实现，都是以监听机制为基础的。

注意：监听器只能执行一次，如果要一直监听，需要实现循环监听（具体实现放在文章后边）

监听机制可以通过下图理解：



二、zk的安装和使用

1、安装zk

下载zk的压缩包

wget [https://mirrors.tuna.tsinghua.edu.cn/apache/zookeeper/zookeeper-3.5.8/](https://mirrors.tuna.tsinghua.edu.cn/apache/zookeeper/zookeeper-3.5.8/apache-zookeeper-3.5.8.tar.gz)[[apache-zookeeper-3.5.8-bin.tar.gz](https://mirrors.tuna.tsinghua.edu.cn/apache/zookeeper/zookeeper-3.5.8/apache-zookeeper-3.5.8.tar.gz)](https://mirrors.tuna.tsinghua.edu.cn/apache/zookeeper/zookeeper-3.5.8/apache-zookeeper-3.5.8-bin.tar.gz)

解压

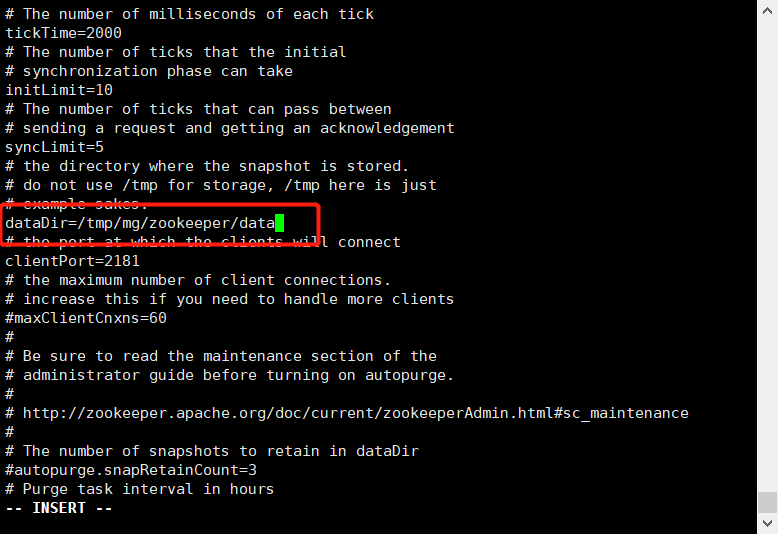
tar -zxvf zookeeper-3.4.12.tar.gz

mv zookeeper-3.4.12 zookeeper

打开zookeeper/conf，复制一份zoo\_sample.cfg为zoo.cfg

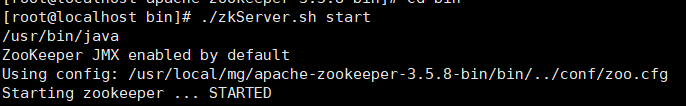
cp zoo\_sample.cfg zoo.cfg

打开zoo.cfg，修改dataDir为已存在路径，如下图



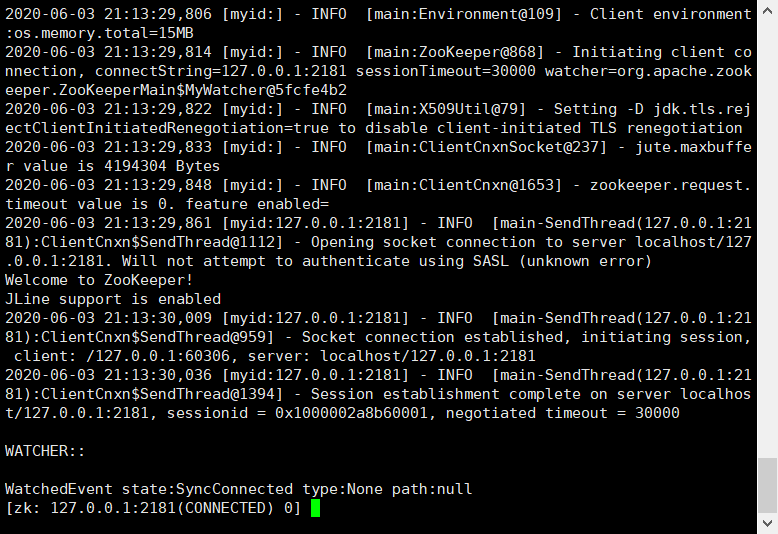
来到zookeeper/bin目录下启动服务

./zkServer.sh start



启动后使用zkCli.sh脚本启动客户端连接

./zkCli.sh -server 127.0.0.1:2181



现在安装成功了，那么看下操作命令吧

2、zk常用命令

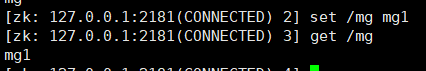
创建 create



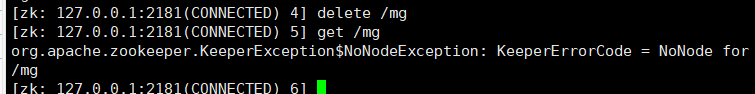
获取 get



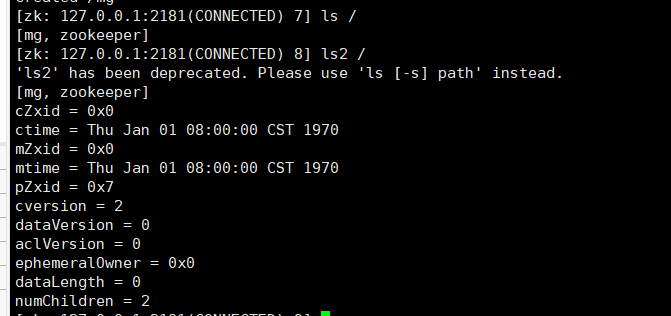
设置 set



删除delete



查询 ls和ls2



三、java中操作zk

1、首先引入依赖

<dependency>  
 <groupId>org.apache.zookeeper</groupId>  
 <artifactId>zookeeper</artifactId>  
 <version>3.6.1</version>  
</dependency>

2、连接zk

@Slf4j  
public class ZKDemo1 implements Watcher

{  
 private static ZooKeeper *zk* = null;  
 private static CountDownLatch *latch* = new CountDownLatch(1);  
 private static Stat *stat* = new Stat();  
 public static void main(String[] args) throws IOException, InterruptedException, KeeperException {  
 String path = "/mg";  
 *zk* = new ZooKeeper("192.168.200.130:2181", 5000, new ZKDemo1());  
 *latch*.await();  
 *log*.info("......成功连接");  
 }  
  
 @SneakyThrows  
 @Override  
 public void process(WatchedEvent watchedEvent) {  
 if (Event.KeeperState.*SyncConnected* == watchedEvent.getState()) {   
  
 if(watchedEvent.getType() == Event.EventType.*None*)  
 {  
 *latch*.countDown();  
 }  
 }  
 }  
}

3、创建节点，并监听

首先看下接下来这段代码

@Slf4j  
public class ZKDemo2 implements Watcher

{  
 private static ZooKeeper *zk* = null;  
 private static CountDownLatch *latch* = new CountDownLatch(1);  
 private static Stat *stat* = new Stat();  
  
 public static void main(String[] args) throws IOException, InterruptedException, KeeperException {  
 String path = "/mg";  
 *zk* = new ZooKeeper("10.0.127.235:2181", 5000, new ZKDemo());  
 *latch*.await();  
 *log*.info("......成功连接");  
  
 if(null!=*zk*.exists(path,false)) {  
 *zk*.delete(path, -1);  
 }  
 String result= *zk*.create(path,"mg".getBytes(), ZooDefs.Ids.*OPEN\_ACL\_UNSAFE*, CreateMode.*PERSISTENT*);  
// zk.getData(path,true,stat);  
 Thread.*sleep*(1000\*10);  
 *zk*.setData(path, ("mg"+System.*currentTimeMillis*()).getBytes(),-1);  
  
 }  
  
 @SneakyThrows  
 @Override  
 public void process(WatchedEvent watchedEvent) {  
 if (Watcher.Event.KeeperState.*SyncConnected* == watchedEvent.getState()) { //zk连接成功通知事件  
  
 if(watchedEvent.getType() == Watcher.Event.EventType.*None*)  
 {  
 *latch*.countDown();  
 }  
 else{  
 System.*out*.println(watchedEvent.getPath()+"......"+new String(*zk*.getData(watchedEvent.getPath(),true,*stat*)));  
 }  
 }  
 }  
}

mg节点被修改了，为什么没有进入监听呢，前边说过zk的监听是一次性的，连接服务器成功后，监听器触发过一次了，需要怎么才能监听节点的修改呢

打开zk.getData(path,true,stat);这行注释掉的代码代码，再试一次，可以了

public byte[] getData(String path, boolean watch, Stat stat)

这个函数watch如果为true，会添加默认的监听器进来。