

使用Java API操作zookeeper的acl权限

发布于 2020-09-23 14:17:08

565

0

举报

默认匿名权限

ZooKeeper提供了如下几种验证模式（scheme）：

- digest：Client端由用户名和密码验证，譬如user:password，digest的密码生成方式是Sha1摘要的base64形式
- auth：不使用任何id，代表任何已确认用户。
- ip：Client端由IP地址验证，譬如172.2.0.0/24
- world：固定用户为anyone，为所有Client端开放权限
- super：在这种scheme情况下，对应的id拥有超级权限，可以做任何事情(cdrwa)

注意的是，exists操作和getAcl操作并不受ACL许可控制，因此任何客户端可以查询节点的状态和节点的ACL。

节点的权限（perms）主要有以下几种：

- Create 允许对子节点Create操作
- Read 允许对本节点GetChildren和GetData操作
- Write 允许对本节点SetData操作
- Delete 允许对子节点Delete操作
- Admin 允许对本节点setAcl操作

Znode ACL权限用一个int型的十进制数字perms表示，perms的5个二进制位分别表示setacl、delete、create、write、read。比如0x1f=adcwr，0x1=----r，0x15=a-c-r。

以下示例在创建节点的时候，赋予该节点的权限为默认匿名权限，权限位为adcwr，换成十进制数字表示就是31，十六进制则是0x1f。代码如下：

```
package org.zero01.zk.demo;

import org.apache.zookeeper.*;
import org.apache.zookeeper.data.ACL;
import org.apache.zookeeper.data.Stat;

import java.util.List;

/**
 * @program: zookeeper-connection
 * @description: zookeeper 操作节点acl权限演示
 * @author: 01
 * @create: 2018-04-27 09:20
 */
public class ZKNodeAcl implements Watcher {

    // 集群模式则是多个ip
    private static final String zkServerIps = "192.168.190.128:2181,192.168.190.129:2181,192.168.190.129:2181,192.168.190.129:2181";
    // 超时时间
    private static final Integer timeout = 5000;
    private static ZooKeeper zooKeeper;
    private static Stat stat;

    // Watch事件通知方法
    public void process(WatchedEvent watchedEvent) {
    }

    public static void main(String[] args) throws Exception {
        zooKeeper = new ZooKeeper(zkServerIps, timeout, new ZKNodeAcl());

        // 这样创建的节点是默认匿名权限的：ZooDefs.Ids.OPEN_ACL_UNSAFE
        String result = zooKeeper.create("/testAclNode", "test data".getBytes(), ZooDefs.Ids.OPEN_ACL_UNSAFE, CreateMode.PERSISTENT);

        Thread.sleep(1000);

        // 获取该节点的acl权限信息
        List<ACL> aclList = zooKeeper.getACL("/testAclNode", stat);
        for (ACL acl : aclList) {
            System.out.println("权限scheme id: " + acl.getId());
        }
    }
}
```

端碗吹水

无无

LV1

494

472.1K

1K

文章

累计阅读量

获赞

关注

前往专栏

加入 开发者社群

领取 专属20元代金券

Get大咖技术交流圈



目录

默认匿名权限

自定义用户权限

ip权限

广告

腾讯云

热门网络产品特惠

流量包 0.01元起

立即抢购

领券

```

41 // 获取的是十进制的int型数字
42 System.out.println("权限位: " + acl.getPerms());
43 }
44 // 避免与服务端的连接马上断开
45 Thread.sleep(1000);
46 }
}

```

控制台输出内容如下:

```

1 权限scheme id: 'world','anyone'
2
3 权限位: 31

```

ZooDefs.Ids可以直接使用的权限如下:

```

1 ZooDefs.Ids.OPEN_ACL_UNSAFE // 默认匿名权限, 权限scheme id: 'world','anyone', 权限位: 31 (adcwr)
2 ZooDefs.Ids.READ_ACL_UNSAFE // 只读权限, 权限scheme id: 'world','anyone', 权限位: 1 (r)

```

自定义用户权限

本节介绍如何自定义用户权限, 这里使用digest进行演示, 因为平时工作中digest是用得最多的。我们都知道digest是使用密文进行设置的, 所以我们需要自定义一个工具类来加密明文密码得到密文密码。代码如下:

```

1 package org.zero01.zk.util;
2
3 import org.apache.zookeeper.server.auth.DigestAuthenticationProvider;
4
5 public class AclUtils {
6
7     public static String getDigestUserPwd(String id) throws Exception {
8         // 加密明文密码
9         return DigestAuthenticationProvider.generateDigest(id);
10    }
11 }

```

然后修改 ZKNodeAcl 类的代码如下:

```

package org.zero01.zk.demo;

import org.apache.zookeeper.*;
import org.apache.zookeeper.data.ACL;
import org.apache.zookeeper.data.Id;
import org.apache.zookeeper.data.Stat;
import org.zero01.zk.util.AclUtils;

import java.util.ArrayList;
import java.util.List;

/**
 * @program: zookeeper-connection
 * @description: zookeeper 操作节点acl权限演示
 * @author: 01
 * @create: 2018-04-27 09:20
 */
public class ZKNodeAcl implements Watcher {

    // 集群模式则是多个ip
    private static final String zkServerIps = "192.168.190.128:2181,192.168.190.129:2181,192.168.190.130:2181";
    // 超时时间
    private static final Integer timeout = 5000;
    private static ZooKeeper zooKeeper;
    private static Stat stat;

    public void process(WatchedEvent watchedEvent) {

    }

    public static void main(String[] args) throws Exception {
        zooKeeper = new ZooKeeper(zkServerIps, timeout, new ZKNodeAcl());

        // 自定义用户认证访问
        List<ACL> acls = new ArrayList<ACL>(); // 权限列表
        // 第一个参数是权限scheme, 第二个参数是加密后的用户名和密码
        Id user1 = new Id("digest", AclUtils.getDigestUserPwd("user1:123456a"));
        Id user2 = new Id("digest", AclUtils.getDigestUserPwd("user2:123456b"));
        acls.add(new ACL(ZooDefs.Perms.ALL, user1)); // 给予所有权限
        acls.add(new ACL(ZooDefs.Perms.READ, user2)); // 只给予读权限
    }
}

```

```

41     acls.add(new ACL(ZooDefs.Perms.DELETE | ZooDefs.Perms.CREATE, user2)); // 多个权限的给予
42
43     // 使用自定义的权限列表去创建节点
44     String result = zooKeeper.create("/testDigestNode", "test data".getBytes(), acls, CreateMode.PERSISTENT);
45     if (result != null) {
46         System.out.println("创建节点: \t" + result + "\t成功...");
47     }
48
49     Thread.sleep(1000);
50
51     // 获取该节点的acl权限信息
52     List<ACL> aclList = zooKeeper.getACL("/testDigestNode", stat);
53     for (ACL acl : aclList) {
54         System.out.println("\n-----\n");
55         System.out.println("权限scheme id: " + acl.getId());
56         System.out.println("权限位: " + acl.getPerms());
57     }
58     Thread.sleep(1000);
59 }

```

控制台输出信息如下:

```

1  创建节点:    /testDigestNode 成功...
2
3  -----
4
5  权限scheme id: 'digest','user1:TQYTqd46qVVbWp0d02tL05qb+JM=
6
7  权限位: 31
8
9  -----
10
11 权限scheme id: 'digest','user2:CV4ED0rE6SxA3h/DN/WyScDMbCs=
12
13 权限位: 1
14
15  -----
16
17 权限scheme id: 'digest','user2:CV4ED0rE6SxA3h/DN/WyScDMbCs=
18
19 权限位: 12

```

我们可以到[服务器上](#)查看该节点的ACL信息是否一致:

```

1  [zk: localhost:2181(CONNECTED) 15] getAcl /testDigestNode
2  'digest','user1:TQYTqd46qVVbWp0d02tL05qb+JM=
3  : cdrwa
4  'digest','user2:CV4ED0rE6SxA3h/DN/WyScDMbCs=
5  : r
6  'digest','user2:CV4ED0rE6SxA3h/DN/WyScDMbCs=
7  : cd
8  [zk: localhost:2181(CONNECTED) 16]

```

如果我们需要操作一个设置了digest权限的节点, 那么就需要登录用于相应权限的用户才行。在代码上也是如此, 我们需要先通过addAuthInfo添加用户信息(账户:明文密码)才能够操作其拥有权限的节点。以下示例演示如何使用addAuthInfo添加用户信息并操作相应的节点, 修改main方法的代码如下:

```

...
public class ZKNodeAcl implements Watcher {
    ...
    public static void main(String[] args) throws Exception {
        zooKeeper = new ZooKeeper(zkServerIps, timeout, new ZKNodeAcl());

        // 注册过的用户必须通过addAuthInfo才能操作节点, 参考命令行 addauth
        zooKeeper.addAuthInfo("digest", "user1:123456a".getBytes());
        String result = zooKeeper.create("/testDigestNode/testOneNode", "test data".getBytes(), ZooDefs.Perms.PERSISTENT);
        if (result != null) {
            System.out.println("创建子节点: \t" + result + "\t成功...");
        }
        // 获取节点数据
        byte[] data = zooKeeper.getData("/testDigestNode/testOneNode", false, stat);
        System.out.println(new String(data));
        // 设置节点数据
        zooKeeper.setData("/testDigestNode/testOneNode", "new test data".getBytes(), 0);
        Thread.sleep(1000);
    }
}

```

领券

```
    }  
}
```

控制台输出信息如下：

```
1 | 创建子节点:    /testDigestNode/testOneNode 成功...  
2 | test data
```

ip权限

以上我们简单演示了默认匿名权限以及自定义的digest权限，下面简单演示下自定义ip权限的设置方式。修改 ZKNodeAcl 类中的main方法代码如下：

```
1 | ...  
2 | public class ZKNodeAcl implements Watcher {  
3 |     ...  
4 |     public static void main(String[] args) throws Exception {  
5 |         zooKeeper = new ZooKeeper(zkServerIps, timeout, new ZKNodeAcl());  
6 |  
7 |         // ip方式的acl  
8 |         List<ACL> aclsIP = new ArrayList<ACL>(); // 权限列表  
9 |         // 第一个参数是权限scheme，第二个参数是ip地址  
10 |         Id ipId1 = new Id("ip", "192.168.190.1");  
11 |         aclsIP.add(new ACL(ZooDefs.Perms.ALL, ipId1)); // 给予所有权限  
12 |  
13 |         // 使用自定义的权限列表去创建节点  
14 |         String result = zooKeeper.create("/testIpNode", "this is test ip node data".getBytes(), a  
15 |         if (result != null) {  
16 |             System.out.println("创建节点: \t" + result + "\t成功...");  
17 |         }  
18 |  
19 |         Thread.sleep(1000);  
20 |  
21 |         // 获取该节点的acl权限信息  
22 |         List<ACL> aclList = zooKeeper.getACL(result, stat);  
23 |         for (ACL acl : aclList) {  
24 |             System.out.println("\n-----\n");  
25 |             System.out.println("权限scheme id: " + acl.getId());  
26 |             System.out.println("权限位: " + acl.getPerms());  
27 |         }  
28 |         Thread.sleep(1000);  
29 |  
30 |         // 获取节点数据，验证ip是否有权限  
31 |         byte[] data = zooKeeper.getData("/testIpNode", false, stat);  
32 |         System.out.println("\n-----\n");  
33 |         System.out.println(result + " 节点当前的数据为: " + new String(data));  
34 |     }  
35 | }
```

控制台输出信息如下：

```
1 | 创建节点:    /testIpNode 成功...  
2 |  
3 | -----  
4 |  
5 | 权限scheme id: 'ip,'192.168.190.1  
6 |  
7 | 权限位: 31  
8 |  
9 | -----  
10 |  
11 | /testIpNode 节点当前的数据为: this is test ip node data
```

本文参与 [腾讯云自媒体分享计划](#)，分享自作者个人站点/博客。
原始发表：2018-04-27，如有侵权请联系 cloudcommunity@tencent.com 删除

[前往查看](#)

tcp/ip

评论

[登录](#) 后参与评论

推荐

jxl的使用总结（java操作Excel）

java

jxl.jar是通过java操作Excel表格的工具类库； jxl.jar包：链接：<http://pan.baidu.com/s/1o8qFJHw> 密码：5jqy 1：通过模拟实现创建一个表格，然后模拟添加到表格中数据，实际开发过程中都是通过从...

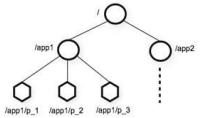


别先生 2018-01-02 729 0

一脸懵逼学习Zookeeper（动物园管理员）---》高度可靠的分布式协调服务

zookeeper 分布式

1：Zookeeper是一个分布式协调服务；就是为用户的分布式应用程序提供协调服务 A、zookeeper是为别的分布式程序服务的 B、Zookeeper本身就是一个分布式程序（只要有半数以...



别先生 2018-01-02 544 0

HBase伪分布式安装(HDFS)+ZooKeeper安装+HBase数据操作+HBase架构体系

数据库 hadoop hbase 分布式 zookeeper

HBase1.2.2伪分布式安装(HDFS)+ZooKeeper-3.4.8安装配置+HBase表和数据操作+HBase的架构体系+单例安装,记录了在Ubuntu下对HBase1.2.2的实践操作,HBase的安装到数据库表的操作.包含内...



王小雷 2018-01-02 1.2K 0

Zookeeper客户端API之读取子节点内容（九）

zookeeper api

本篇博客介绍一下Zookeeper原声API读取节点内容的使用方法。 读取节点内容方法介绍 方法 Zookeeper提供了两个方法来获取节点内容，同步获取和异步获取： public byte[] getData(String path, boolean watch, Stat stat) public void getData(final String...

程序新视界 2018-01-08 907 0

Zookeeper客户端API之创建节点（七）

zookeeper api

本篇博客主要讲一下使用Zookeeper原生的客户API来创建一个数据节点。 创建数据节点方法 Zookeeper提供了两个创建数据节点的方法。 同步创建数据节点方法： public String create(final String path, byte data[], List<ACL> acl, CreateMode createMode)...

程序新视界 2018-01-08 1.6K 0

Zookeeper客户端API之读取子节点列表（八）

zookeeper api

本篇博客介绍一下Zookeeper原生客户端API提供的获取子节点列表方法。 获取子节点列表方法 方法 Zookeeper原生客户端API提供了以下8中获取子节点列表的方法，每个方法的使用说明参考注释内容： /** * 返回指定路径下面的子节点列表。 * 如果watch...

程序新视界 2018-01-08 1.4K 0

Zookeeper之开源客户端ZkClient

zookeeper 开源 api maven

ZkClient是由Datameer的工程师开发的开源客户端，对Zookeeper的原生API进行了包装，实现了超时重连、Watcher反复注册等功能。 ZKClient版本及源码 maven依赖 ZkClient目前有两个不同artifactId的系列。 其中最早的0.1版本maven依赖如下： ...

程序新视界 2018-01-08 3.3K 0

zookeeper：分布式协调服务浅析

zookeeper 分布式 java

领券

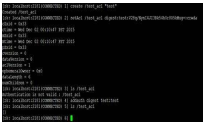
由于之前了解一个分布式框架（dubbo）时，其中涉及到zookeeper，因此今天就先来大概的介绍一下zookeeper。zookeeper是一个用来管理大量的主机的分布式协调服务。

挖掘大数据 2018-01-10 613 0

Zookeeper权限管理与Quota管理

其他
Zookeeper的ACL机制和Quota机制网上资料较少，这里做一个总结，以供大家参考。1 Zookeeper ACL ZooKeeper的权限管理亦即ACL控制功能通过Server、Client两端协调完成：Server端：一个...

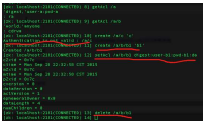
三丰SanFeng 2018-01-16 982 0



ZooKeeper 笔记(5) ACL(Access Control List)访问控制列表

zookeeper access 分布式 存储
zk做为分布式架构中的重要中间件，通常会在上面以节点的方式存储一些关键信息，默认情况下，所有应用都可以读写任何节点，在复杂的应用中，这不太安全，ZK通过ACL机制来解决访问权限问题...

菩提树下的杨过 2018-01-19 2.1K 0



ZooKeeper 笔记(1) 安装部署及hello world

zookeeper apache html java
先给一堆学习文档，方便以后查看 官网文档地址大全： OverView(概述)
http://zookeeper.apache.org/doc/r3.4.6/zookeeperOver.html Getting Started(开始入门)...

菩提树下的杨过 2018-01-19 653 0

Windows安装和使用zookeeper

windows zookeeper 分布式 hadoop apache
之前整理过一篇文章《zookeeper 分布式锁服务》，本文介绍的 Zookeeper 是以 3.4.5 这个稳定版本为基础，最新的版本可以通过官网 http://hadoop.apache.org/zookeeper/来获取，Zookeeper 的安装...

张善友 2018-01-19 1.1K 0



Hadoop快速入门

hadoop 分布式 大数据
传说中的Hadoop，我终于来对着你唱"征服"了，好可爱的小象，！J总的来说，hadoop的思路比较简单(map-reduce)，就是将任务分开进行，最后汇总。但这个思路实现起来，比较复杂，但相对于几年前Intel等硬件公司提出的网格运算等方式，显得更加...

用户1216676 2018-01-24 1.5K 0

ZooKeeper快速入门

docker 分布式 zookeeper
“一入Java深似海”，本文将重点介绍分布式相关理论以及Zookeeper的基础应用，深入内容之后会持续更新。？分布式基本概念
分布式架构：在集中式的系统环境中，可以简单通过事务(ACID)保证数据的一致性；而在分布式系统环境中，由于缺少全局时...

用户1216676 2018-01-24 1.1K 0

Dubbo（二）认识Zookeeper

数据库 dubbo zookeeper
前言 在昨天，我们给大家基本介绍了Dubbo，文中反复提到了Zookeeper，那么它到底是什么呢，这篇文章我们将从Dubbo层面去了解Zookeeper，不做全面讲解，毕竟这是Dubbo教程啊~ Zookeeper...

生活创客 2018-01-30 650 0



ZooKeeper构建分布式锁（选译）

zookeeper 分布式
作者：Scott Leberknight 译者：java达人 来源：
http://www.sleberknight.com/blog/sleberkn/entry/building_a_distributed_lock_revisited (点击阅读...)

java达人 2018-01-31 536 0



大数据干货系列（四）-ZooKeeper总结

分布式

ZooKeeper总结


一、本质 ZooKeeper是一个为分布式应用提供一致性服务的软件。 二、ZooKeeper解决了什么问题 1.分布式系统的一致性问题 2.分布式系统的容灾容错 3.分布式系统的执行顺序问题 4...

企鹅号小编

2018-02-01

699

0



Zookeeper ACL权限配置及zkclient示例

zookeeper

分布式

zk做为分布式架构中的重要中间件，通常会在上面以节点的方式存储一些关键信息，默认情况下，所有应用都可以读写任何节点，在复杂的应用中，这不太安全，ZK通过ACL机制来解决访问权限问题，详见官网文档： ...

斯武、风晴

2018-03-01

4K

0

5个不为人知的Java API使用技巧

java

api

程序员都喜欢使用API！ 例如为app应用构建API或作为微服务架构体系的一部分。当然，使用API的前提是能让你的工作变得更轻松。为了简化开发和提高工作效率所作出的努力，有时也意味着需要寻找新的类库或者过程（或者减少过程）。对于很多开发...

BestSDK

2018-03-01

699

0

使用HDFS客户端java api读取hadoop集群上的信息

java

hadoop

本文介绍使用hdfs java api的配置方法。 1、先解决依赖， pom <dependency> <groupId>org.apache.hadoop</groupId> <artifactId>hadoop-client</artifactId> <version>2.7.2</version> <scope>provided</scope> </dependency> 2、配置文件，存放h

用户1225216

2018-03-05

4.4K

2

加载更多

社区

活动

资源

关于

腾讯云开发者

专栏文章

自媒体分享计划

技术周刊

社区规范

扫码关注腾讯云开发者

阅读清单

邀请作者入驻

社区标签

免责声明

腾讯云开发者

互动问答

自荐上首页

开发者手册

联系我们

技术沙龙

技术竞赛

开发者实验室

友情链接

技术视频

团队主页

扫码关注腾讯云开发者

腾讯云TI平台

领取腾讯云代金券

热门产品

域名注册
云存储

云服务器
视频直播

区块链服务

消息队列

网络加速

云数据库

域名解析

热门推荐

人脸识别
SSL 证书

腾讯会议
语音识别

企业云

CDN加速

视频通话

图像分析

MySQL 数据库

更多推荐

数据安全
网站监控

负载均衡
数据迁移

短信

文字识别

云点播

商标注册

小程序开发