

pandaj的专栏

自己动手，丰衣足食

目录视图 摘要视图 RSS 订阅

个人资料



pandajava

访问： 38221次
积分： 960
等级： BLOG > 3
排名： 千里之外

原创： 55篇
转载： 47篇
译文： 0篇
评论： 5条

文章搜索

文章分类

- java (26)
- JVM (1)
- js (2)
- PHP (4)
- linux (13)
- Spring (2)
- DB (7)
- ExtJS (9)
- 算法 (9)
- Apache Flume (5)
- Apache Hive (1)
- http|servlet (1)
- Work Info (12)
- github (2)
- OS (1)
- interview (2)
- Other (2)

文章存档

- 2016年08月 (1)
- 2016年04月 (2)
- 2016年03月 (1)

【1024程序员节】参加活动领暖心礼品 【观点】有了深度学习，你还学传统机器学习算法？ 【资源库】火爆了的React Native都在研究什么

(9)rocketMQ3.2.6 单机安装 及 web界面监控 部署

2016-03-09 11:26 1929人阅读 评论(0) 收藏 举报

版权声明：本文为博主原创文章，未经博主允许不得转载。

1.

rocketMQ3.2.6安装 <https://github.com/alibaba/RocketMQ>
执行nohup sh mqbroker -n 127.0.0.1:9876 &
报错 java.lang.ExceptionInInitializerError
Caused by: java.net.UnknownHostException: xxxxxxx: 未知的名称或服务

解决方案: http://www.easytd.com/yichangjijin/d_16012708270.html

vi /proc/sys/kernel/hostname
或者
sysctl kernel.hostname=localhost

2.

nohup sh mqnamesrv -n 127.0.0.1:9876 &
The Name Server boot success.

nohup sh mqbroker -n 127.0.0.1:9876 &
The broker[localhost, 192.xxx.xxx.128:10911] boot success. and name server is 127.0.0.1:9876

进程:

```
root 16148 15609 0 12:46 pts/1 00:00:00 sh mqnamesrv -n 127.0.0.1:9876
root 16152 16148 0 12:46 pts/1 00:00:00 sh /usr/local/rocketMq/alibaba-rocketmq/bin/runserver.sh
com.alibaba.rocketmq.namesrv.NamesrvStartup -n 127.0.0.1:9876
root 16154 16152 0 12:46 pts/1 00:00:37 /usr/local/jdk/jdk1.7/bin/java -server -Xms4g -Xmx4g -Xmn2g -
XX:PermSize=128m -XX:MaxPermSize=320m -XX:+UseConcMarkSweepGC -
XX:+UseCMSCompactAtFullCollection -XX:CMSInitiatingOccupancyFraction=70 -
XX:+CMSParallelRemarkEnabled -XX:SoftRefLRUPolicyMSPerMB=0 -XX:+CMSClassUnloadingEnabled -
XX:SurvivorRatio=8 -XX:+DisableExplicitGC -verbose:gc -Xloggc:/root/rmq_srv_gc.log -XX:+PrintGCDetails -
XX:-OmitStackTraceInFastThrow -Djava.ext.dirs=/usr/local/rocketMq/alibaba-rocketmq/bin/../lib -cp
./usr/local/rocketMq/alibaba-rocketmq/bin/../conf/./usr/local/jdk/jdk1.7/lib/dt.jar:/usr/local/jdk/jdk1.7/lib/tools.jar
com.alibaba.rocketmq.namesrv.NamesrvStartup -n 127.0.0.1:9876
```

```
root 16290 15609 0 12:54 pts/1 00:00:00 sh mqbroker -n 127.0.0.1:9876
root 16294 16290 0 12:54 pts/1 00:00:00 sh /usr/local/rocketMq/alibaba-rocketmq/bin/runbroker.sh
com.alibaba.rocketmq.broker.BrokerStartup -n 127.0.0.1:9876
```

2015年12月 (1)
2015年10月 (3)
展开

阅读排行	
(5) MySQL INSERT插入	(2190)
(9)rocketMQ3.2.6 单机安	(1929)
(2) flume 入门学习 Hello	(1645)
(2)Mysql ----- The last pa	(939)
(3)项目导入ext之后 一直	(759)
(2)ExtJs4.2 类的定义、维	(625)
(5) flume ---- 工作记录20	(625)
(4) Debug JDK source 无	(617)
(9) ExtJS4.2 Window的月	(582)
(9) java ---- 集合框架图	(575)

评论排行	
(1) Linux Top命令参数详	(2)
(16) 正则表达式 学习参	(1)
(7) <IED> eclipse-3.6.1-6	(1)
(2)PHP 学习笔记 from ir	(1)
(2) flume 入门学习 Hello	(0)
(12) sublime 3 + python	(0)
(5)extjs4.2 客户端代理Lc	(0)
(4)ExtJS4.2 数据模型	(0)
(3)Extjs如何动态加载Ext	(0)
(1)ExtJs4.2 HelloWorld	(0)

推荐文章	
* 2016 年最受欢迎的编程语言是什么？	
* Chromium扩展（Extension）的页面（Page）加载过程分析	
* Android Studio 2.2 来啦	
* 手把手教你做音乐播放器（二）技术原理与框架设计	
* JVM 性能调优实战之：使用阿里开源工具 TProfiler 在海量业务代码中精确定位性能代码	

最新评论	
(1) Linux Top命令参数详解 及 Lir pandajava: @Cooling88:问我我也不知道，大多用现查，记住的基本是用的多了。	
(1) Linux Top命令参数详解 及 Lir Coohx: 刚才面试被问到了，没答出来...	
(7) <IED> eclipse-3.6.1-64安装反确定: 好东西，此方法有效	
(2)PHP 学习笔记 from imooc 蓝鸟先森: 果断的收藏了	
(16) 正则表达式 学习参考 wanderer: 以前一直不太懂，看了这篇文章后恍然大悟了，谢谢！	

```
root 16297 16294 5 12:54 pts/1 00:13:55 /usr/local/jdk/jdk1.7/bin/java -server -Xms4g -Xmx4g -Xmn2g -XX:PermSize=128m -XX:MaxPermSize=320m -XX:+UseConcMarkSweepGC -XX:+UseCMSCompactAtFullCollection -XX:CMSInitiatingOccupancyFraction=70 -XX:+CMSParallelRemarkEnabled -XX:SoftRefLRUPolicyMSPerMB=0 -XX:+CMSClassUnloadingEnabled -XX:SurvivorRatio=8 -XX:+DisableExplicitGC -verbose:gc -Xloggc:/root/rmq_bk_gc.log -XX:+PrintGCDetails -XX:+PrintGCDateStamps -XX:-OmitStackTraceInFastThrow -Djava.ext.dirs=/usr/local/rocketMq/alibaba-rocketmq/bin/./lib -cp ./usr/local/rocketMq/alibaba-rocketmq/bin/./conf:./usr/local/jdk/jdk1.7/lib/dt.jar:/usr/local/jdk/jdk1.7/lib/tools.jar com.alibaba.rocketmq.broker.BrokerStartup -n 127.0.0.1:9876
```

3

创建maven项目：<http://www.yiibai.com/maven/m2eclipse-project.html>（代码转载）

```
[java]
01. package com.pjm.mqstudy.recoetmq;
02. import java.util.concurrent.TimeUnit;
03.
04.
05. import com.alibaba.rocketmq.client.exception.MQClientException;
06. import com.alibaba.rocketmq.client.producer.DefaultMQProducer;
07. import com.alibaba.rocketmq.client.producer.SendResult;
08. import com.alibaba.rocketmq.common.message.Message;
09.
10. public class Producer {
11.     public static void main(String[] args) throws MQClientException,
12.         InterruptedException {
13.         /**
14.          * 一个应用创建一个Producer，由应用来维护此对象，可以设置为全局对象或者单例<br>
15.          * 注意：ProducerGroupName需要由应用来保证唯一<br>
16.          * ProducerGroup这个概念发送普通的消息时，作用不大，但是发送分布式事务消息时，比较关键，
17.          * 因为服务器会回查这个Group下的任意一个Producer
18.          */
19.         DefaultMQProducer producer = new DefaultMQProducer("ProducerGroupName");
20.         producer.setNamesrvAddr("192.1xx.xxx.1xx:9876");
21.         producer.setInstanceName("Producer");
22.
23.         /**
24.          * Producer对象在使用之前必须要调用start初始化，初始化一次即可<br>
25.          * 注意：切记不可以每次发送消息时，都调用start方法
26.          */
27.         producer.start();
28.
29.         /**
30.          * 下面这段代码表明一个Producer对象可以发送多个topic，多个tag的消息。
31.          * 注意：send方法是同步调用，只要不抛异常就标识成功。但是发送成功也可能会有多种状态，<br>
32.          * 例如消息写入Master成功，但是Slave不成功，这种情况消息属于成功，但是对于个别应用如果对消息
33.          * 可靠性要求极高，<br>
34.          * 需要对这种情况做处理。另外，消息可能会存在发送失败的情况，失败重试由应用来处理。
35.          */
36.         for (int i = 0; i < 100; i++) {
37.             try {
38.                 {
39.                     Message msg = new Message("TopicTest1", // topic
40.                         "TagA", // tag
41.                         "OrderID001", // key
42.                         ("Hello MetaQ").getBytes()); // body
43.                     SendResult sendResult = producer.send(msg);
44.                     System.out.println(sendResult);
45.                 }
46.             }
47.             {
48.                 Message msg = new Message("TopicTest2", // topic
49.                     "TagB", // tag
50.                     "OrderID0034", // key
51.                     ("Hello MetaQ").getBytes()); // body
52.                 SendResult sendResult = producer.send(msg);
53.                 System.out.println(sendResult);
54.             }
55.             {
56.                 Message msg = new Message("TopicTest3", // topic
57.                     "TagC", // tag
58.                     "OrderID061", // key
59.                     ("Hello MetaQ").getBytes()); // body
```



```

59.         SendResult sendResult = producer.send(msg);
60.         System.out.println(sendResult);
61.     }
62.     } catch (Exception e) {
63.         e.printStackTrace();
64.     }
65.     TimeUnit.MILLISECONDS.sleep(1000);
66. }
67.
68. /**
69.  * 应用退出时，要调用shutdown来清理资源，关闭网络连接，从MetaQ服务器上注销自己
70.  * 注意：我们建议应用在JBoss、Tomcat等容器的退出钩子里调用shutdown方法
71.  */
72. producer.shutdown();
73. }
74. }

```

[java]



```

01. package com.pjm.mqstudy.recoetmq;
02. import java.util.List;
03.
04. import com.alibaba.rocketmq.client.consumer.DefaultMQPushConsumer;
05. import com.alibaba.rocketmq.client.consumer.listener.ConsumeConcurrentlyContext;
06. import com.alibaba.rocketmq.client.consumer.listener.ConsumeConcurrentlyStatus;
07. import com.alibaba.rocketmq.client.consumer.listener.MessageListenerConcurrently;
08. import com.alibaba.rocketmq.client.exception.MQClientException;
09. import com.alibaba.rocketmq.common.message.MessageExt;
10.
11. public class PushConsumer {
12.
13.     /**
14.      * 当前例子是PushConsumer用法，使用方式给用户感觉是消息从RocketMQ服务器推到了应用客户端。<br>
15.      * 但是实际PushConsumer内部是使用长轮询Pull方式从MetaQ服务器拉消息，然后再回调用户Listener方法
16.      */
17.     public static void main(String[] args) throws InterruptedException,
18.         MQClientException {
19.
20.         /**
21.          * 一个应用创建一个Consumer，由应用来维护此对象，可以设置为全局对象或者单例<br>
22.          * 注意：ConsumerGroupName需要由应用来保证唯一
23.          */
24.         DefaultMQPushConsumer consumer = new DefaultMQPushConsumer(
25.             "ConsumerGroupName");
26.         consumer.setNamesrvAddr("192.1xx.xxx.xxx:9876");
27.         consumer.setInstanceName("Consumer");
28.
29.         /**
30.          * 订阅指定topic下tags分别等于TagA或TagC或TagD
31.          */
32.         consumer.subscribe("TopicTest1", "TagA || TagC || TagD");
33.
34.         /**
35.          * 订阅指定topic下所有消息<br>
36.          * 注意：一个consumer对象可以订阅多个topic
37.          */
38.         consumer.subscribe("TopicTest2", "*");
39.
40.         consumer.registerMessageListener(new MessageListenerConcurrently() {
41.
42.             /**
43.              * 默认msgs里只有一条消息，可以通过设置consumeMessageBatchMaxSize参数来批量接收消息
44.              */
45.             public ConsumeConcurrentlyStatus consumeMessage(
46.                 List<MessageExt> msgs, ConsumeConcurrentlyContext context) {
47.                 System.out.println(Thread.currentThread().getName()
48.                     + " Receive New Messages: " + msgs.size());
49.
50.                 MessageExt msg = msgs.get(0);
51.                 if (msg.getTopic().equals("TopicTest1")) {
52.                     // 执行TopicTest1的消费逻辑
53.                     if (msg.getTags() != null && msg.getTags().equals("TagA")) {
54.                         // 执行TagA的消费
55.                         System.out.println(new String(msg.getBody()));
56.                     } else if (msg.getTags() != null
57.                         && msg.getTags().equals("TagC")) {
58.                         // 执行TagC的消费
59.                     } else if (msg.getTags() != null

```

```

58.         && msg.getTags().equals("TagD")) {
59.             // 执行TagD的消费
60.         }
61.     } else if (msg.getTopic().equals("TopicTest2")) {
62.         System.out.println(new String(msg.getBody()));
63.     }
64.
65.     return ConsumeConcurrentlyStatus.CONSUME_SUCCESS;
66. }
67. });
68.
69. /**
70.  * Consumer对象在使用之前必须要调用start初始化，初始化一次即可<br>
71.  */
72. consumer.start();
73.
74. System.out.println("Consumer Started.");
75. }
76. }

```

4

创建topic

```
[root@PjmLinuxCentOs6 bin]# sh mqadmin updateTopic -n 192.168.164.128:9876 -c DefaultCluster -t TopicTest1
```

create topic to 192.xxx.xxx.xxx:10911 success.

TopicConfig [topicName=TopicTest1, readQueueNums=8, writeQueueNums=8, perm=RW-, topicFilterType=SINGLE_TAG, topicSysFlag=0, order=false]

运行Producer程序，系统报错 no route info of this topic rocketmq

原因linux的系统防火墙没有对外开放相关端口，停用防火墙

1) 重启后生效

开启: chkconfig iptables on

关闭: chkconfig iptables off

2) 即时生效，重启后失效

开启: service iptables start

关闭: service iptables stop

再次运行Producer程序:

```

8:04:36.789 [NettyClientWorkerThread_1] DEBUG io.netty.util.ResourceLeakDetector - -
Dio.netty.leakDetectionLevel: simple
18:04:37.001 [NettyClientSelector_1] DEBUG io.netty.util.internal.Cleaner0 - java.nio.ByteBuffer.cleaner():
available
SendResult [sendStatus=SEND_OK, msgId=C0A8A48000002A9F0000000000000000,
messageQueue=MessageQueue [topic=TopicTest1, brokerName=localhost, queueId=0], queueOffset=0]
SendResult [sendStatus=SEND_OK, msgId=C0A8A48000002A9F0000000000000094,
messageQueue=MessageQueue [topic=TopicTest2, brokerName=localhost, queueId=0], queueOffset=0]
SendResult [sendStatus=SEND_OK, msgId=C0A8A48000002A9F0000000000000129,
messageQueue=MessageQueue [topic=TopicTest3, brokerName=localhost, queueId=0], queueOffset=0]
SendResult [sendStatus=SEND_OK, msgId=C0A8A48000002A9F00000000000001BD,
messageQueue=MessageQueue [topic=TopicTest1, brokerName=localhost, queueId=1], queueOffset=0]
SendResult [sendStatus=SEND_OK, msgId=C0A8A48000002A9F0000000000000251,
messageQueue=MessageQueue [topic=TopicTest2, brokerName=localhost, queueId=1], queueOffset=0]
SendResult [sendStatus=SEND_OK, msgId=C0A8A48000002A9F00000000000002E6,
messageQueue=MessageQueue [topic=TopicTest3, brokerName=localhost, queueId=1], queueOffset=0]
.....

```

运行Consumer程序:

Consumer Started.

ConsumeMessageThread_1 Receive New Messages: 1

ConsumeMessageThread_2 Receive New Messages: 1

Hello MetaQ

Hello MetaQ

ConsumeMessageThread_3 Receive New Messages: 1

Hello MetaQ

ConsumeMessageThread_4 Receive New Messages: 1

Hello MetaQ

ConsumeMessageThread_5 Receive New Messages: 1

Hello MetaQ

ConsumeMessageThread_6 Receive New Messages: 1

Hello MetaQ

ConsumeMessageThread_7 Receive New Messages: 1

Hello MetaQ

ConsumeMessageThread_8 Receive New Messages: 1

Hello MetaQ

ConsumeMessageThread_9 Receive New Messages: 1

Hello MetaQ

ConsumeMessageThread_10 Receive New Messages: 1

Hello MetaQ

ConsumeMessageThread_11 Receive New Messages: 1

Hello MetaQ

ConsumeMessageThread_12 Receive New Messages: 1

Hello MetaQ

ConsumeMessageThread_13 Receive New Messages: 1

Hello MetaQ

ConsumeMessageThread_17 Receive New Messages: 1

Hello MetaQ

ConsumeMessageThread_19 Receive New Messages: 1

Hello MetaQ

ConsumeMessageThread_14 Receive New Messages: 1

Hello MetaQ

ConsumeMessageThread_16 Receive New Messages: 1

Hello MetaQ

ConsumeMessageThread_18 Receive New Messages: 1

Hello MetaQ

ConsumeMessageThread_15 Receive New Messages: 1

Hello MetaQ

ConsumeMessageThread_1 Receive New Messages: 1

Hello MetaQ

ConsumeMessageThread_20 Receive New Messages: 1

Hello MetaQ

5搭建web监控界面

<https://github.com/rocketmq/rocketmq-console>

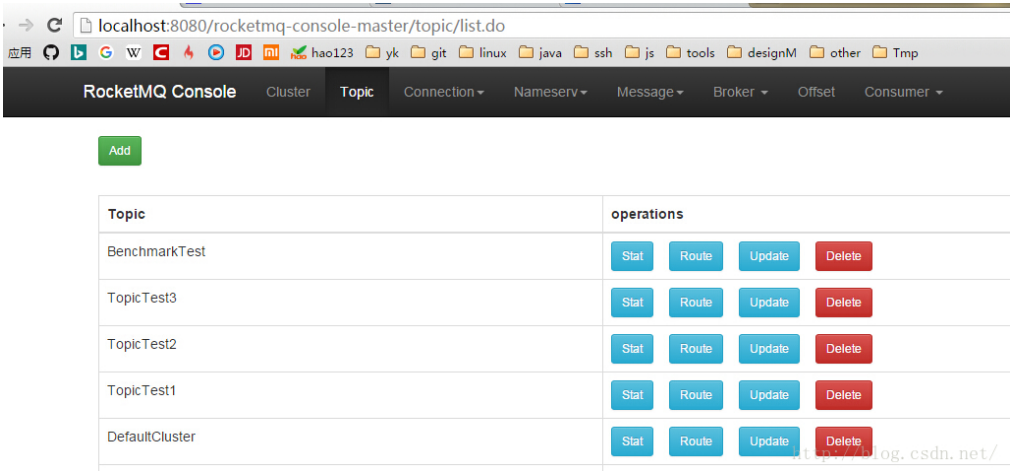
config.properties配置 mq地址端口

Tomcat启动找不到maven里的类 抛ClassNotFoundException

参考 <http://www.yihaomen.com/article/java/471.htm>

1. 右键单击工程项目 -> 点击 properties
2. 选择 Deployment Assembly
3. 点击 Add -> Java Build Path Entries -> Next
4. 选择 Maven Dependencies -> Finish -> Apply -> OK

5. Clean project and server. 重启server



顶 踩
0 0

上一篇 (8) 一个死循环一直不停的new对象过程中的垃圾回收的情况
下一篇

(10)json字符转 转JavaBean，且bean中还有List<T>属性，重点@JsonProperty("tags") [jackson]

参考知识库



.NET 知识库
1019 关注 | 774 收录



Linux 知识库
4742 关注 | 3212 收录



Java 知识库
16838 关注 | 1336 收录



Java EE 知识库
7715 关注 | 721 收录



Java SE 知识库
14627 关注 | 459 收录

猜你在找

企业Web服务器Nginx应用实战
windows server 2008 r2服务器操作系统
[颠覆传统所学]web开发级Centos颠覆实战(上)
[微信公众号_独立知识点]微信WEB开发
Python Django Web开发入门

9ECLIPSE中部署工程报Undefined exploded
监控WebLogic910的项目部署到Tomcat报
监控WebLogic910的项目部署到Tomcat报
比cacti更好的linux单机监控Monitorix的安装与配
个人hadoop学习总结Hadoop集群+HBase集群

查看评论

暂无评论

您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

全部主题

Hadoop

AWS

移动游戏

Java

Android

iOS

Swift

智能硬件

Docker

OpenStack

VPN

Spark

ERP

IE10

Eclipse

CRM

JavaScript

数据库

Ubuntu

NFC

WAP

jQuery

BI

HTML5

Spring

Apache

.NET

API

HTML

SDK

IIS

Fedora

XML

LBS

Unity

Splashtop

UML

components

Windows Mobile

Rails

QEMU

KDE

Cassandra

CloudStack

FTC

coremail

OPhone

CouchBase

云计算

iOS6

Rackspace

Web App

SpringSide

Maemo

Compuware

大数据

aptech

Perl

Tornado

Ruby

Hibernate

ThinkPHP

HBase

Pure

Solr

Angular

Cloud Foundry

Redis

Scala

Django

Bootstrap

公司简介 | 招贤纳士 | 广告服务 | 银行汇款帐号 | 联系方式 | 版权声明 | 法律顾问 | 问题报告 | 合作伙伴 | 论坛反馈

网站客服 杂志客服 微博客服 webmaster@csdn.net 400-600-2320 | 北京创新乐知信息技术有限公司 版权所有 | 江苏乐知网络技术有限公司 提供商务支持
京 ICP 证 09002463 号 | Copyright © 1999-2016, CSDN.NET, All Rights Reserved 