车联网方案技术开发说明

|  |  |  |  |
| --- | --- | --- | --- |
| **版本** | **修改人** | **修改日期** | **修改内容** |
| V0.9 | 王奇 | 20181222 | 建档 |
| V1.0 | 王奇 | 20200103 | 同步2.0的修改 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

[车联网方案技术开发说明 1](#_Toc28965982)

[1概述 4](#_Toc28965983)

[1.1架构设计图： 4](#_Toc28965984)

[1.2国内测试服服务器配置： 4](#_Toc28965985)

[2中间件技术实施 5](#_Toc28965986)

[2.1 Redis集群： 5](#_Toc28965987)

[2.1.1下载安装Redis 5](#_Toc28965988)

[2.1.2 测试Redis 5](#_Toc28965989)

[2.1.3 配置Redis 5](#_Toc28965990)

[3.1.7 创建集群 8](#_Toc28965991)

[3.1.8 给集群设置密码 9](#_Toc28965992)

[3.1.9 Redis使用方式 11](#_Toc28965993)

[2.2 ActiveMQ： 11](#_Toc28965994)

[2.2.1 ActiveMQ下载安装 11](#_Toc28965995)

[2.2.2 设置ActiveMQ服务开机启动 12](#_Toc28965996)

[2.2.3 设置账户密码 13](#_Toc28965997)

[2.3 ZooKeeper: 13](#_Toc28965998)

[1.下载zookeeper 13](#_Toc28965999)

[2.解压zookeeper 13](#_Toc28966000)

[3. 修改配置文件 14](#_Toc28966001)

[4标识节点 14](#_Toc28966002)

[5.分发zookeeper修改对应端口 15](#_Toc28966003)

[6.启动zookeeper 15](#_Toc28966004)

[7.查看zookeeper运行状态 15](#_Toc28966005)

[2.4 Kafka： 15](#_Toc28966006)

[2.5消费端： 17](#_Toc28966007)

[2.6 Mysql: 17](#_Toc28966008)

[2.6.1 下载Mysql 17](#_Toc28966009)

[2.6.2 安装Mysql 17](#_Toc28966010)

[2.6.3 单机模式 18](#_Toc28966011)

[2.6.4 多实例模式 19](#_Toc28966012)

[2.6.5 Mysql主从： 22](#_Toc28966013)

[2.6.6 其它操作： 22](#_Toc28966014)

[2.7 Hadoop： 23](#_Toc28966015)

[2.8 HBase： 23](#_Toc28966016)

[下载 HBase： 23](#_Toc28966017)

[安装配置 HBase： 23](#_Toc28966018)

[3技术实施2 24](#_Toc28966019)

[3.1 Netty大并发网关 24](#_Toc28966020)

[3.2大并发测试系统 24](#_Toc28966021)

[4 技术实施3 24](#_Toc28966022)

[4.1 接入入口Nginx 24](#_Toc28966023)

[4.2 接入网关JavaAPI 24](#_Toc28966024)

[4.3 FTP服务器 24](#_Toc28966025)

[4.3.1 VSFTP服务器 24](#_Toc28966026)

[5支撑技术实施 24](#_Toc28966027)

[5.1操作系统优化 24](#_Toc28966028)

[5.2产品数据模拟系统 24](#_Toc28966029)

[5.3服务检测系统 24](#_Toc28966030)

[5.3.1 IO检测程序ioSocketCase 24](#_Toc28966031)

[5.4运维脚本 25](#_Toc28966032)

[5.4.1自动运行脚本 25](#_Toc28966033)

[5.4.2服务检测脚本service.sh 25](#_Toc28966034)

[5.5设置Java开发环境 25](#_Toc28966035)

[1.下载Java8 25](#_Toc28966036)

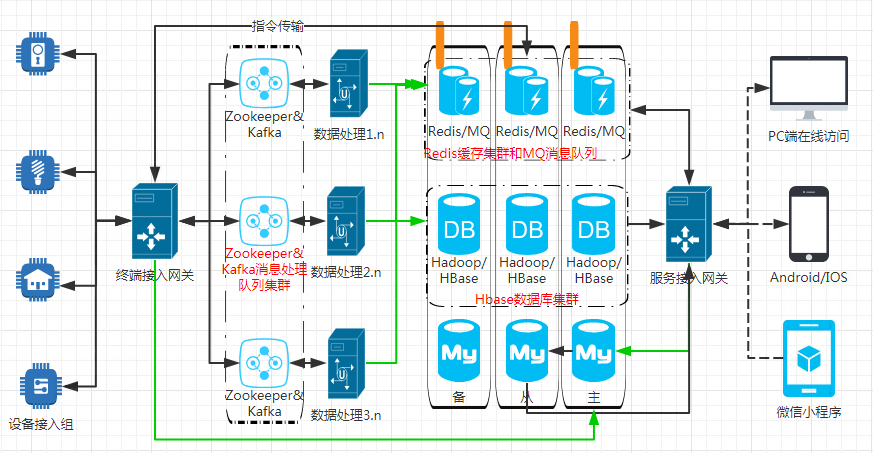
[3.设置Java8环境 25](#_Toc28966037)

[4.更新环境 26](#_Toc28966038)

[5.测试Java8 26](#_Toc28966039)

# 1概述

## 1.1架构设计图：



## 1.2国内测试服服务器配置：

国外正式服务器。

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | **服务器M5** | | | **服务器R5** | | | |
| **CPU /颗数/核数** | | 4 | | | 4 | | | |
| **内存** | | DDR4 16G | | | DDR4 32G | | | |
| **硬盘阵列/容量** | | EBS 150G | | | EBS 16G + SSD 150G | | | |
| **操作系统** | | CentOS7.5 | | | CentOS7.5 | | | |
| **服务器IP** | | 172.31.10.12/ | | | 172.31.12.176/ | | | |
| **部署**  **软件** | Zookeeper1 | Zookeeper2 | Kafka1 | Kafka2 | Zookeeper3 | Kafka3 |  |  |
| Redis2004 | Redis2005 | Redis2006 |  | Redis2001 | Redis2002 | Redis2003 |  |
| 消息端1 | 消息端2 | 消息端3 | 消息端4 | 消息端5 | 消息端6 | ifengstarIO | HK-IO |
| Nginx |  |  |  |  |  |  |  |
|  |  |  |  |  | Redis6379 | Nginx |  |
|  |  |  |  |  | Tomcat | Tomcat | Tomcat |

# 2中间件技术实施

## 2.1 Redis集群：

|  |  |  |  |
| --- | --- | --- | --- |
| **端口** | **关系** | **部署位置** | |
| 2001 | 主  从 | ServerA: 113.106.93.247 | /home/deve/redis-cluster/2000 |
| 2003 | ServerA: 113.106.93.247 | /home/deve/redis-cluster/2001 |
| 2002 | 主  从 | ServerA: 113.106.93.247 | /home/deve/redis-cluster/2002 |
| 2004 | ServerA: 113.106.93.247 | /home/deve/redis-cluster/2003 |
| 2002 | 主  从 | ServerA: 113.106.93.247 | /home/deve/redis-cluster/2004 |
| 2005 | ServerA: 113.106.93.247 | /home/deve/redis-cluster/2005 |

### 2.1.1下载安装Redis

[deve@r730-104 ~]$ cd soft

[deve@r730-104 ~]$wget download.redis.io/releases/redis-5.0.7.tar.gz

[deve@r730-104 ~]$ mkdir -p /home/deve/redis-cluster

[deve@r730-104 ~]$ tar -zxvf redis-5.0.7.tar.gz

[deve@r730-104 ~]$ cd redis-5.0.7

[deve@r730-104 ~]$ make PREFIX=/home/deve/redis-cluster/ install

make PREFIX=/data/m5-module/redis-cluster/ install

[deve@r730-104 ~]$ cd /home/deve/redis-cluster

### 2.1.2 测试Redis

[deve@r730-104 ~]$ yum install gcc tcl

[deve@r730-104 ~]$ make test

### 2.1.3 配置Redis

新建6个文件夹用来存放每一个节点数据。为了方便移植将Redis可执行文件拷贝到每一个文件夹。

#----redis cluster------------------------------------------------------------------

1:修改IP地址，只有这些地址可以访问redis

bind 127.0.0.1 172.31.10.12 172.31.12.176

2.关闭保护模式

# change-cluster-20191226

protected-mode no

3. 修改端口

#port 6379

# change-cluster-20191226

port 2001

4.后台运行

#daemonize no //redis后台运行

# change-cluster-20191226

daemonize yes

5.设置运行PID

#pidfile /var/run/redis\_6379.pid //pidfile文件对应2001,2002,2003,2004,2005,2006

# change-cluster-20191226

pidfile /var/run/redis\_2001.pid

6.设置日志文件位置

#logfile ""

# change-cluster-20191226

logfile "/data/logs/redis/redis-2001.log"

7.是否追加

#appendonly no

# change-cluster-20191226

appendonly yes

8.开启集群

# cluster-enabled yes

# change-cluster-20191226

cluster-enabled yes

9.设置节点名称

# cluster-config-file nodes-6379.conf

# change-cluster-20191226

cluster-config-file nodes-2002.conf

10.设置超时连接时间

# cluster-node-timeout 15000

# change-cluster-20191226

cluster-node-timeout 5000

[deve@r730-104 ~]$ vim start-all.sh

cd /gps/r5-module/redis-cluster/2001/

redis-server redis.conf

cd ../2002

redis-server redis.conf

cd ../2003

redis-server redis.conf

#-----------------------------------------------------------------------------------

[deve@r730-104 ~]$ cp -R 2001 2006

[deve@r730-104 ~]$ cp -R 2001 2002

[deve@r730-104 ~]$ cp -R 2001 2003

[deve@r730-104 ~]$ cp -R 2001 2004

[deve@r730-104 ~]$ cp -R 2001 2005

修改相应的redis.conf

编写启动脚本

[deve@r730-104 ~]$ vim start-all.sh

cd 2006

redis-server redis.conf

cd ../2001

redis-server redis.conf

cd ../2002

redis-server redis.conf

cd ../2003

redis-server redis.conf

cd ../2004

redis-server redis.conf

cd ../2005

redis-server redis.conf

关闭脚本

[deve@r730-104 ~]$ vim stop-all.sh

2006/redis-cli -h 172.0.0.1 -p 2006 shutdown

2001/redis-cli -h 172.0.0.1 -p 2001 shutdown

2002/redis-cli -h 172. 0.0.1 -p 2002 shutdown

2003/redis-cli -h 172. 0.0.1 -p 2003 shutdown

2004/redis-cli -h 172. 0.0.1 -p 2004 shutdown

2005/redis-cli -h 172. 0.0.1 -p 2005 shutdown

如果有需要需要使用redis-cli -h 172.0.0.1 -p 2001 -a youpassword shutdown

[deve@r730-104 ~]$ chmod 777 start-all.sh

[deve@r730-104 ~]$ chmod 777 stop-all.sh

启动所有Redis

[deve@r730-104 ~]$ ./ start-all.sh

[deve@r730-104 ~]$ ps -ef | grep redis

观察子目录下的redis-XXXX.log

### 3.1.7 创建集群

[deve@r730-104 ~]$ ./redis-trib.rb create --replicas 1 172.16.18.104:2000 172.16.18.104:2001 172.16.18.104:2002 172.16.18.104:2003 172.16.18.104:2004 172.16.18.104:2005

>>> Creating cluster

>>> Performing hash slots allocation on 6 nodes...

Using 3 masters:

127.0.0.1:2000

127.0.0.1:2001

127.0.0.1:2002

Adding replica 127.0.0.1:2004 to 127.0.0.1:2000

Adding replica 127.0.0.1:2005 to 127.0.0.1:2001

Adding replica 127.0.0.1:2003 to 127.0.0.1:2002

>>> Trying to optimize slaves allocation for anti-affinity

[WARNING] Some slaves are in the same host as their master

M: d268ffb7f284ff22ad3d9d9877d89489f4f082a4 127.0.0.1:2000

slots:0-5460 (5461 slots) master

M: c576b1175b4035787cd087e8efa2ec8c33fe3352 127.0.0.1:2001

slots:5461-10922 (5462 slots) master

M: 9a0ef38514b5e9114687501e3fe84f7ac183c021 127.0.0.1:2002

slots:10923-16383 (5461 slots) master

S: 4bbee759b49d68ce7dc104a61bdbb05776c718b2 127.0.0.1:2003

replicates d268ffb7f284ff22ad3d9d9877d89489f4f082a4

S: a7fa12ff5dbeb8cabedfda8621b3a4708e119557 127.0.0.1:2004

replicates c576b1175b4035787cd087e8efa2ec8c33fe3352

S: 38300c40240cd80e9eb5e2f1d6cc556a6722f324 127.0.0.1:2005

replicates 9a0ef38514b5e9114687501e3fe84f7ac183c021

Can I set the above configuration? (type 'yes' to accept): yes

>>> Nodes configuration updated

>>> Assign a different config epoch to each node

>>> Sending CLUSTER MEET messages to join the cluster

Waiting for the cluster to join....

>>> Performing Cluster Check (using node 127.0.0.1:2000)

M: d268ffb7f284ff22ad3d9d9877d89489f4f082a4 127.0.0.1:2000

slots:0-5460 (5461 slots) master

1 additional replica(s)

S: a7fa12ff5dbeb8cabedfda8621b3a4708e119557 127.0.0.1:2004

slots: (0 slots) slave

replicates c576b1175b4035787cd087e8efa2ec8c33fe3352

S: 38300c40240cd80e9eb5e2f1d6cc556a6722f324 127.0.0.1:2005

slots: (0 slots) slave

replicates 9a0ef38514b5e9114687501e3fe84f7ac183c021

S: 4bbee759b49d68ce7dc104a61bdbb05776c718b2 127.0.0.1:2003

slots: (0 slots) slave

replicates d268ffb7f284ff22ad3d9d9877d89489f4f082a4

M: c576b1175b4035787cd087e8efa2ec8c33fe3352 127.0.0.1:2001

slots:5461-10922 (5462 slots) master

1 additional replica(s)

M: 9a0ef38514b5e9114687501e3fe84f7ac183c021 127.0.0.1:2002

slots:10923-16383 (5461 slots) master

1 additional replica(s)

[OK] All nodes agree about slots configuration.

>>> Check for open slots...

>>> Check slots coverage...

[OK] All 16384 slots covered.

[deve@r730-104 ~]$

### 3.1.8 给集群设置密码

使用 ./stop-all.sh关闭集群，修改每个节点的配置文件

#----redis cluster------------------------------------------------------------------

bind 127.0.0.1 106.3.226.247 113.106.93.247

daemonize yes //redis后台运行

pidfile /var/run/redis\_2000.pid //pidfile文件对应2000,2001,2002,2003,2004,2005

port 2000 //端口2000,2001,2002,2003,2004,2005

cluster-enabled yes //开启集群 把注释#去掉

cluster-config-file nodes\_2000.conf //集群的配置 配置文件首次启动自动生成

cluster-node-timeout 5000 //请求超时 设置5秒够了

一般来说不需要开启appendonly aof日志文件太大，GPS应用对Redis数据，并没有非常高要求

appendonly no

#appendonly yes //aof日志开启 有需要就开启，它会每次写操作都记录一条日志

dir /home/developer/redis-cluster/2000/

masterauth youpassword //如果需要密码访问Redis请设置密码

requirepass hkredispassword1test

logfile "/home/developer/redis-cluster/2000/redis-2000.log"

#-----------------------------------------------------------------------------------

带密码访问集群

./redis-cli -h 127.0.0.1 -c -p 2000

2000/redis-cli -c -p 2000

auth hkredispassword1test

./redis-cli -h 127.0.0.1 -c -p 2000 -a hkredispassword1test

./redis-cli -h 172.16.18.104 -c -p 2000

auth hkredispassword1test

备注：集群配置时最好只使用一个IP地址。

### 3.1.9 Redis使用方式

## 2.2 ActiveMQ：

|  |  |  |  |
| --- | --- | --- | --- |
| **端口** | **部署位置** | **服务对象** | **Topic路径** |
| ServerA:61616 | /home/deve/module/apache-activemq-5.15.3 |  |  |
| ServerA:8161 |  |  |
| bs.ifengstar.com:61616 | /home/deve/module/apache-activemq-5.15.3 |  |  |
| bs.ifengstar.com: 8161 |  |  |

ActiveMQ是开源组织apache.org旗的一款开源消息中间件。访问网址为：

http://activemq.apache.org/activemq-5158-release.html

### 2.2.1 ActiveMQ下载安装

[deve@r730-104 soft]$ wget http://mirror.bit.edu.cn/apache//activemq/5.15.8/apache-activemq-5.15.8-bin.tar.gz

[deve@r730-104 soft]$ tar -zvxf a apache-activemq-5.15.8-bin.tar.gz -C /home/deve/module/activemq

**启动ActiveMQ**

[root@r730-104 activemq]# /home/deve/module/activemq/bin/activemq start

INFO: Loading '/home/deve/module/activemq//bin/env'

INFO: Using java '/usr/local/jdk1.8.0\_191/bin/java'

INFO: Starting - inspect logfiles specified in logging.properties and log4j.properties to get details

INFO: pidfile created : '/home/deve/module/activemq//data/activemq.pid' (pid '17206')

**查看ActiveMQ状态**

[root@r730-104 activemq]# /home/deve/module/activemq/bin/activemq status

INFO: Loading '/home/deve/module/activemq//bin/env'

INFO: Using java '/usr/local/jdk1.8.0\_191/bin/java'

ActiveMQ not running

**停止ActiveMQ运行**

[root@r730-104 activemq]# /home/deve/module/activemq/bin/activemq stop

INFO: Loading '/home/deve/module/activemq//bin/env'

INFO: Using java '/usr/local/jdk1.8.0\_191/bin/java'

ERROR: No or outdated process id in '/home/deve/module/activemq//data/activemq.pid'

INFO: Removing /home/deve/module/activemq//data/activemq.pid

### 2.2.2 设置ActiveMQ服务开机启动

* 建立软连接：

[root@r730-104 activemq]# ln -s /home/deve/module/activemq/bin/activemq /etc/init.d/activemqd

* 注册为系统服务

[root@r730-104 activemq]# vim /etc/init.d/activemqd

- 添加下面内容到/etc/init.d/activemq脚本

# chkconfig: 345 63 37

# description: Auto start ActiveMQ

JAVA\_HOME=/usr/local/jdk1.8.0\_191

JAVA\_CMD=java1234

\*\*/usr/local/jdk1.8.0\_191/bin/env增加ACTIVEMQ\_HOME,JAVA\_HOME

\*\*echo $PATH

\*\* 增加PATH=

​ export PATH 代替JAVA\_HOME

\*\*\*重复第一步

---------------------------------------示例-----------------------------------------------------------------------------------

!/bin/sh

description: Auto start ActiveMQ

BEGIN INIT INFO

Provides: activemq

Required-Start: remote\_fs network $syslog

Required-Stop: remote\_fs network $syslog

Default-Start: 2 3 4 5

Default-Stop: 0 6

chkconfig: 2345 64 36

Short-Description: Starts ActiveMQ

Description: Starts ActiveMQ Message Broker Server

END INIT INFO

PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/usr/java/jdk1.8.0\_191/bin:/usr/java/jdk1.8.0\_191/jre/bin:/root/bin

export PATH

JAVA\_CMD=java1234

ACTICVEMQ\_HOME=/usr/local/activemq-5.15.3

------------------------------------------------------------------------

* 开启开机自启

[root@r730-104 activemq]# chkconfig activemq on

[root@r730-104 activemq]# reboot

* 以系统服务的方式启动、查看状态和停止服务

[root@r730-104 activemq]# service activemq start

[root@r730-104 activemq]# service activemq status

[root@r730-104 activemq]# service activemq stop

### 2.2.3 设置账户密码

修改activemq.xml

<!-- destroy the spring context on shutdown to stop jetty -->

<shutdownHooks>

<bean xmlns="http://www.springframework.org/schema/beans" class="org.apache.activemq.hooks.SpringContextHook" />

</shutdownHooks>

<!-- 添加访问ActiveMQ的账号密码 -->

​ <plugins>

​ <simpleAuthenticationPlugin>

​ <users>

​ <authenticationUser username="hkadmin" password="hk667" groups="users,admins"/>

​ </users>

​ </simpleAuthenticationPlugin>

​ </plugins>

## 2.3 ZooKeeper:

Zookeeper是基于Java技术开发，运行前需要配置JDK环境，请参见第5章节技术支撑--JDK安装配置

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **编号** | **端口** | **集群IP及端口** | **部署位置** | **数据文件位置** |
| 1 | 2181 | 172.31.10.12:2881:3881 | /data/m5-module/zookeeper1 | /data/m5-module/zookeeper1 |
| 2 | 2182 | 172.31.10.12:2882:3882 | /data/m5-module/zookeeper2 | /data/m5-module/zookeeper2 |
| 3 | 2183 | 172.31.12.176:2883:3883 | /gps/r5-module/zookeeper3 | /data/r5-module/zookeeper3 |
|  |  |  |  |  |

### 1.下载zookeeper

cd /data/m5-module/src

wget https://archive.apache.org/dist/zookeeper/zookeeper-3.5.6/apache-zookeeper-3.5.6.bin.tar.gz

### 2.解压zookeeper

tar -zvxf apache-zookeeper-3.5.6.tar.gz

mv apache-zookeeper-3.5.6 ../zookeeper1

### 3. 修改配置文件

cd ../zookeeper1/conf

cp zoo\_sample.cfg zoo.cfg

vim zoo.cfg

--------------------------------------------

dataDir=/data/m5-module/zookeeper1/data

dataLogDir=/data/m5-module/zookeeper1/logs

clientPort=2181

server.1=172.31.10.12:2881:3881

server.2=172.31.10.12:2882:3882

server.3=172.31.12.176:2883:3883

#如果是三台不同的主机，端口配置可以相同；不同的主机，端口配置需要不同

.........................................................................

dataDir=/data/m5-module/zookeeper2/data

dataLogDir=/data/m5-module/zookeeper2/logs

clientPort=2182

server.1=172.31.10.12:2881:3881

server.2=172.31.10.12:2882:3882

server.3=172.31.12.176:2883:3883

.........................................................................

dataDir=/data/r5-module/zookeeper3/data

dataLogDir=/data/r5-module/zookeeper3/logs

clientPort=2183

server.1=172.31.10.12:2881:3881

server.2=172.31.10.12:2882:3882

server.3=172.31.12.176:2883:3883

---------------------------------------------

### 4标识节点

分别在三台主机的dataDir目录下新建myid文件,并写入对应节点标识，Zookeeper集群通过myid配置的节点通信端口和选举端口来进行节点通信，选举出Leader节点。

---------------------------------------------

cd /data/m5-module/zookeeper1

mkdir data

echo "1" > data/myid

.......................................................

cd /data/m5-module/zookeeper2

mkdir data

echo "2" > data/myid

.......................................................

cd /gps/r5-module/zookeeper3

mkdir data

echo "3" > data/myid

---------------------------------------------

### 5.分发zookeeper修改对应端口

cd /data/m5-module/

cp -r zookeeper1 zookeeper2

scp -r zookeeper1 root@172.31.12.176:/gps/r5-module/zookeeper3

### 6.启动zookeeper

cd /data/m5-module/zookeeper1/

bin/zkServer.sh start

---------------------------------------------

cd /data/m5-module/zookeeper2/

bin/zkServer.sh start

---------------------------------------------

cd /gps/r5-module/zookeeper3/

bin/zkServer.sh start

### 7.查看zookeeper运行状态

bin/zkServer.sh status

/usr/local/jdk8/bin/java

ZooKeeper JMX enabled by default

Using config: /data/m5-module/zookeeper2/bin/../conf/zoo.cfg

Client port found: 2182. Client address: localhost.

Mode: leader

## 2.4 Kafka：

M5: 172.31.10.12; R5:172.31.12.176

|  |  |  |
| --- | --- | --- |
| **端口** | **部署位置** |  |
| M5:9092 | M5:/data/m5-module/kafka1 |  |
| M5:9093 | M5:/data/m5-module/kafka2 |  |
| R5:9092 | R5: /gps/r5-module/kafka3 |  |
|  |  |  |

1.下载默认kafka:

wget [**http://mirrors.tuna.tsinghua.edu.cn/apache/kafka/2.4.0/kafka\_2.12-2.4.0.tgz**](http://mirrors.tuna.tsinghua.edu.cn/apache/kafka/2.4.0/kafka_2.12-2.4.0.tgz)

wget [**https://www-us.apache.org/dist/kafka/2.4.0/kafka\_2.12-2.4.0.tgz**](https://www-us.apache.org/dist/kafka/2.4.0/kafka_2.12-2.4.0.tgz)

tar -zvxf kafka\_2.12-2.4.0.tgz

2.配置kafka:

cd /data/m5-module/kafka1

vim config/server.roperties

----------------------------------------------------------

在broker.id=0 后面增加如下配置：

delete.topic.enable=true

listeners=PLAINTEXT://172.31.10.12:9092

advertised.host.name=172.31.10.12

advertised.port=9092

//avertised.port=9092

advertised.listeners=PLAINTEXT://172.31.10.12:9092

log.dirs=/data/m5-module/kafka1/logs

zookeeper.connect=172.31.12.176:2183,172.31.10.12:2182,172.31.10.12:2181

......................................................................

cd /gps/r5-module/kafka3

vim config/server.roperties

在broker.id=2 后面增加如下配置：

delete.topic.enable=true

listeners=PLAINTEXT://172.31.12.176:9092

advertised.host.name=172.31.12.176

advertised.port=9092

advertised.listeners=PLAINTEXT://172.31.12.176:9092

log.dirs=/data/m5-module/kafka1/logs

zookeeper.connect=172.31.12.176:2183,172.31.10.12:2182,172.31.10.12:2181

----------------------------------------------------------

3.启动kafka

----------------------------------------------------------

cd /data/m5-module/kafka1

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

......................................................................

cd /data/m5-module/kafka2

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

......................................................................

cd /gps/r5-module/kafka3

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

----------------------------------------------------------

4.查看当前服务器中的所有topic

cd /gps/r5-moduld/kafka3

bin/kafka-topics.sh --zookeeper 172.31.12.176:2183 --list

5.创建集群

bin/kafka-topics.sh --zookeeper 172.31.12.176:2183 --create --replication-factor 3 --partitions 1 --topic first

6.发送信息

cd /gps/r5-moduld/kafka3

bin/kafka-console-producer.sh --broker-list 172.31.12.176:9092 --topic first

>hello world

>angry bird

7.消费消息

bin/kafka-console-consumer.sh --bootstrap-server 172.31.12.176:9092 --topic first --from-beginning

8.删除topic

后台启动kafka

nohup bin/kafka-server-start.sh config/server.properites &

关闭kafka

cd /gps/r5-moduld/kafka3

bin/kafka-server-stop.sh stop

## 2.5消费端：

|  |  |  |
| --- | --- | --- |
| **IP及端口** | **部署位置** |  |
| M5:2181 | /data/m5-module/app/ifengConsumer |  |
| M5:2182 | /data/m5-module/app/ifengConsumer |  |
|  | /data/m5-module/app/ifengConsumer |  |
|  | /data/m5-module/app/ifengConsumer |  |
|  | /data/m5-module/app/ifengConsumer |  |
|  | /data/m5-module/app/ifengConsumer |  |

## 2.6 Mysql:

|  |  |  |  |
| --- | --- | --- | --- |
| **端口** | **类型** | **部署地址** |  |
| D1:3306 | 主 | /data/mysql |  |
| D2:3306 | 从 | /data/mysql |  |

### 2.6.1 下载Mysql

wget https://cdn.mysql.com//Downloads/MySQL-5.7/mysql-5.7.24-linux-glibc2.12-x86\_64.tar.gz

### 2.6.2 安装Mysql

* 解压：tar -zxvf mysql-5.7.24-linux-glibc2.12-x86\_64.tar.gz -C /usr/local
* 增加环境变量：

echo 'export PATH=$PATH:/usr/local/mysql/bin' >> /etc/profile

source /etc/profile

* 为centos添加mysql用户组和mysql用户(-s /bin/false参数指定mysql用户仅拥有所有权，而没有登录权限):
* groupadd mysql
* useradd -r -g mysql -s /bin/false mysql2.6.3 配置Mysql
* 进入安装mysql软件的目录： cd /usr/local/mysql
* 修改当前目录拥有者为新建的mysql用户： chown -R mysql:mysql ./

### 2.6.3 单机模式

* 1、修改my.cnf

|  |
| --- |
| vim /etc/my.conf  #-------my.cnf-----------#  # For advice on how to change settings please see  # http://dev.mysql.com/doc/refman/5.7/en/server-configuration-defaults.html  [mysqld]  #  # Remove leading # and set to the amount of RAM for the most important data  # cache in MySQL. Start at 70% of total RAM for dedicated server, else 10%.  # innodb\_buffer\_pool\_size = 128M  #  # Remove leading # to turn on a very important data integrity option: logging  # changes to the binary log between backups.  # log\_bin  #  # Remove leading # to set options mainly useful for reporting servers.  # The server defaults are faster for transactions and fast SELECTs.  # Adjust sizes as needed, experiment to find the optimal values.  # join\_buffer\_size = 128M  # sort\_buffer\_size = 2M  # read\_rnd\_buffer\_size = 2M  #datadir=/var/lib/mysql  user=mysql  datadir=/home/mysql/8610/mysql  port=8610  #socket=/var/lib/mysql/mysql.sock  socket=/home/mysql/8610/mysql.sock  # Disabling symbolic-links is recommended to prevent assorted security risks  symbolic-links=0  #log-error=/var/log/mysqld.log  log-error=/home/mysql/8610/mysqld.log  sql\_mode=STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION  pid-file=/home/mysql/8610/mysqld.pid  [client]  socket=/home/mysql/8610/mysql.sock  #--------end my.cnf----------------# |

* 2、在mysql用户目录下建立数据文件夹

cd /home/mysql/ && mkdir -p 8610

chown mysql:mysql 8610

chmod 777 8610

cd /home/mysql/8610/ && mkdir mysql

chown mysql:mysql mysql

chmod 777 mysql

touch mysqld.log

chown mysql:mysql mysqld.log

chmod 777 mysqld.log

* 3、初始化数据库

cd /usr/local/mysql

bin/mysqld --initialize --user=mysql --basedir=/usr/local/mysql --datadir=/home/mysql/8610/mysql/

在初始化后需要记录root用户的随机密码。如果命令行没有给出，请查阅日志文件。

* 4、开启mysql服务，命令如下：

./support-files/mysql.server start

* 5、将mysql进程放入系统进程中，命令如下：

cp support-files/mysql.server /etc/init.d/mysqld

* 6、重新启动mysql服务，命令如下：

service mysqld restart

* 7、使用随机密码登录mysql数据库，命令如下：

mysql -u root -p

等待系统提示，输入随机密码，即可登录

* 8、进入mysql操作行，为root用户设置新密码（fhxt&clw715#）：

alter user 'root'@'localhost' identified by 'fhxt&clw715#';

* 9、设置允许远程连接数据库，命令如下：

update user set user.Host='%' where user.User='root';

* 10、新建用户并设置远程访问权限

#grant all privileges on \*.\* to gpsadmin@localhost identified by "1qaz&619" ;

grant all privileges on \*.\* to gpsadmin@"%" identified by "1qaz&619" ;

select host,user from user; 　　//查询mysql中所有用户权限

* 11、刷新权限，命令如下：

flush privileges;

### 2.6.4 多实例模式

* 1、修改my.cnf

|  |
| --- |
| vim /etc/my.conf  #-------my.cnf-----------#  vim /etc/my.conf  [client]  port=3306  socket=/tmp/mysql.sock  [mysqld\_multi]  mysqld = /usr/local/mysql/bin/mysqld\_safe  mysqladmin = /usr/local/mysql/bin/mysqladmin  log = /home/mysql/mysqld\_multi.log  [mysqld]  user=mysql  basedir = /usr/local/mysql  sql\_mode=NO\_ENGINE\_SUBSTITUTION,STRICT\_TRANS\_TABLES  [mysqld3306]  mysqld=mysqld  mysqladmin=mysqladmin  datadir=/home/mysql/mysql-3306/mysql  port=3306  server\_id=3306  socket=/tmp/mysql-3306.sock  log-output=file  slow\_query\_log = 1  long\_query\_time = 1  slow\_query\_log\_file = /home/mysql/mysql-3306/log/slow.log  log-error = /home/mysql/mysql-3306/log/error.log  log-bin = /home/mysql/mysql-3306/log/mysql3306-bin  binlog-ignore-db = mysql  [mysqld8610]  mysqld=mysqld  mysqladmin=mysqladmin  datadir=/home/mysql/mysql-8610/mysql  port=8610  server\_id=8610  socket=/tmp/mysql-8610.sock  log-output=file  slow\_query\_log = 1  long\_query\_time = 1  slow\_query\_log\_file = /home/mysql/mysql-8610/log/slow.log  log-error = /home/mysql/mysql-8610/log/error.log  log-bin = /home/mysql/mysql-8610/log/mysql8610-bin  replicate-ignore-db=mysql  relay-log = slave-relay-bin  relay-log-index = slave-relay-bin.index  read\_only  #--------end my.cnf----------------# |

* 2、在mysql用户目录下建立数据文件夹

|  |
| --- |
| cd /home/mysql/ && mkdir -p 8610  chown mysql:mysql 8610  chmod 777 8610  cd /home/mysql/8610/ && mkdir mysql  chown mysql:mysql mysql  chmod 777 mysql  touch mysqld.log  chown mysql:mysql mysqld.log  chmod 777 mysqld.log |
| cd /home/mysql/ && mkdir -p 3619  chown mysql:mysql 3619  chmod 777 3619  cd /home/mysql/3619/ && mkdir mysql  chown mysql:mysql mysql  chmod 777 mysql  touch mysqld.log  chown mysql:mysql mysqld.log  chmod 777 mysqld.log |
| cd /home/mysql/ && mkdir -p 3306  chown mysql:mysql 3306  chmod 777 3306  cd /home/mysql/3306/ && mkdir mysql  chown mysql:mysql mysql  chmod 777 mysql  touch mysqld.log  chown mysql:mysql mysqld.log  chmod 777 mysqld.log |

* 3、将mysql进程放入系统进程中，命令如下：

cp support-files/mysql.server /etc/init.d/mysqld

* 4、初始化数据库

cd /usr/local/mysql

|  |
| --- |
| mysqld --initialize --user=mysql --basedir=/usr/local/mysql --datadir=/home/mysql/3306/mysql/ |
| mysqld --initialize --user=mysql --basedir=/usr/local/mysql --datadir=/home/mysql/3916/mysql/ |
| mysqld --initialize --user=mysql --basedir=/usr/local/mysql --datadir=/home/mysql/8610/mysql/ |

*在初始化后需要记录root用户的随机密码。如果命令行没有给出，请查阅日志文件。*

* 5、启动mysql集群，命令如下：

/usr/local/mysql/bin/mysqld\_multi start

* 6、查看集群：

/usr/local/mysql/bin/mysqld\_multi report

Reporting MySQL servers

MySQL server from group: mysqld3306 is running

MySQL server from group: mysqld8610 is running

[root@r430-102 mysql]# ss -tupln | grep mysqld

tcp LISTEN 0 80 :::8610 :::\* users:(("mysqld",pid=93909,fd=23))

tcp LISTEN 0 80 :::3306 :::\* users:(("mysqld",pid=93906,fd=23))

* 7、进入mysql操作行，为root用户设置新密码（fhxt&clw715#）：

查看单实例：/usr/local/mysql/bin/mysqld\_multi report 3306

查看监听端口：ss -tulpn | grep mysqld

修改密码：

mysql -S /tmp/mysql-3306.sock -p

set password=password('123456');

set password=password('fhxt&clw715#');

* 9、设置允许远程连接数据库，命令如下：

update user set user.Host='%' where user.User='root';

* 10、新建用户并设置远程访问权限

#grant all privileges on \*.\* to gpsadmin@localhost identified by "1qaz&619" ;

grant all privileges on \*.\* to gpsadmin@"%" identified by "1qaz&619" ;

select host,user from user; 　　//查询mysql中所有用户权限

* 11、刷新权限，命令如下：

flush privileges;

### 2.6.5 Mysql主从：

### 2.6.6 其它操作：

查询数据库中的存储过程

方法一：select `name` from mysql.proc where db = 'your\_db\_name' and `type` = 'PROCEDURE'

方法二：show procedure status;

查看存储过程或函数的创建代码

show create procedure proc\_name;

show create function func\_name;

## 2.7 Hadoop：

目前暂时单接点HBase，没有架设Hadoop

|  |  |  |  |
| --- | --- | --- | --- |
| **端口** | **类型** | **部署地址** |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## 2.8 HBase：

|  |  |  |  |
| --- | --- | --- | --- |
| **端口** | **类型** | **部署地址** |  |
| ServerA:启动时随机 | 单点 | /home/deve/hbase |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

### 下载 HBase：

wget http://mirrors.hust.edu.cn/apache/hbase/2.1.2/hbase-2.1.2-bin.tar.gz

### 安装配置 HBase：

# 3技术实施2

## 3.1 Netty大并发网关

## 3.2大并发测试系统

# 4 技术实施3

## 4.1 接入入口Nginx

## 4.2 接入网关JavaAPI

## 4.3 FTP服务器

### 4.3.1 VSFTP服务器

查看是否已经安装VSFTPD

rpm -qa | grep vsftpd

安装Vsftpd

yum -y install vsftpd

设置开机自动启动

chkconfig vsftpd on

service vsftpd on

# 5支撑技术实施

## 5.1操作系统优化

## 5.2产品数据模拟系统

## 5.3服务检测系统

### 5.3.1 IO检测程序ioSocketCase

AWS R5地址：/home/centos/monitor/go/ioSocketCase

检测对像包含使用字符串协议的ifengstarIO和使用二进制协议的HKIO；

每隔一分钟分别向这两个IO发送一包数据，如果连续失败三次则发送短信进行提醒。

## 5.4运维脚本

### 5.4.1自动运行脚本

AWS crontab -e （使用centos账户运行）

|  |  |  |  |
| --- | --- | --- | --- |
| **检测项** | **时间** | **系统账户** | **脚本路径** |
| Service.sh | 间隔10m | centos | /home/centos/monitor/go/service.sh |
| 内存释放 | 每日6点 | centos | /home/centos/monitor/go/memory/freemem.sh |
|  |  |  |  |

### 5.4.2服务检测脚本service.sh

AWS R5地址：/home/centos/monitor/go/service.sh

主要功能检测系统相关服务是否在运行，如果检测到没有运行，则自动启动相关服务。

|  |  |  |  |
| --- | --- | --- | --- |
| **检测项** | **时间** | **系统账户** | **脚本路径** |
| ioSocketCase | 间隔10m | centos | /home/centos/monitor/go/ioSocketCase |
|  |  |  |  |
|  |  |  |  |

#!/bin/bash

num=`ps -eml | grep -w ioSocketCase | grep -v grep |wc -l `

if [ $num -eq 0 ] ; then

echo "!restart ioSocketCase!";

cd /home/centos/monitor/go/ && nohup ./ioSocketCase &

fi

## 5.5设置Java开发环境

### 1.下载Java8

当前最新版本下载地址：http://www.oracle.com/technetwork/java/javase/downloads/index.html

历史版本下载地址：　　http://www.oracle.com/technetwork/java/javase/archive-139210.html

2.安装Java8

tar -zvxf jdk-8u221-linux-x64.tar.gz

### 3.设置Java8环境

vim /etc/profile

在export PATH USER LOGNAME MAIL HOSTNAME HISTSIZE HISTCONTROL 前面增加：

JAVA\_HOME=/usr/local/jdk8

export PATH=$PATH:$JAVA\_HOME/bin

### 4.更新环境

source /etc/profile

### 5.测试Java8

[root@r730-105]java -version

java version "1.8.0\_221"

java(TM) SE Runtime Environment (build 1.8.0\_221-b11)

java HotSpot(TM) 64-Bit Server VM (build 25.221-b11, mixed mode)