**B52项目实施方案**

**一、目的：**构建稳定、可靠以及安全的项目环境，保证B52项目能够顺利部署，并且平稳运行。保证平台具有一定的处理并发事件的能力，确保B52项目可以顺利投入生产使用。同时，注重数据安全，对项目信息进行保护，加强对数据的容灾备份的考虑，提前做好回滚准备。在确保项目可以顺利实施的前提下，做好对数据的管理。

**二、步骤：**

1、评估项目大小，预测项目访问量

2、规划服务器数量，分配服务器作用及性能

3、根据规划购买云主机以及公网IP和CDN服务

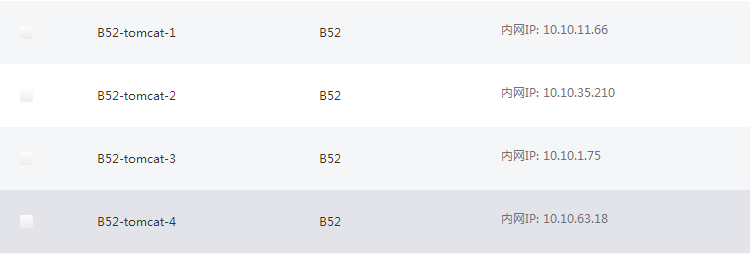
4、协同实施服务器部署

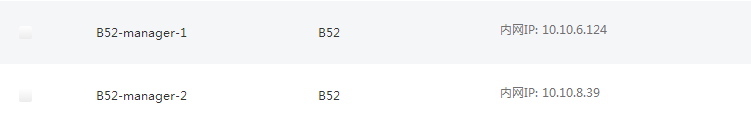
5、部署文档记录整理

**三、详细部署内容：**

1. **Tomcat:**

主机名及主机IP：





部署步骤：

主机信息：CentOS release 6.5 (Final)

Tomcat版本： tomcat-7.0.59

Java版本：jdk-7.76

软件包：apache-tomcat-7.0.59.tar.gz

jdk-7u76-linux-x64.gz

apr-util-1.5.4.tar.gz

apr-iconv-1.2.1.tar.gz

apr-1.5.2.tar.gz

tomcat-native-1.1.33-src.tar.gz

1. 卸载系统原有java环境

# rpm -e --nodeps java-1.7.0-openjdk-1.7.0.65-2.5.1.2.el6\_5.x86\_64  
# rpm -e --nodeps java-1.6.0-openjdk-1.6.0.0-11.1.13.4.el6.x86\_64

1. 安装jdk环境

# tar zxvf jdk-7u76-linux-x64.gz

# mv jdk1.7.0\_76/ /data/java

1. 安装tomcat

#tar zxvf apache-tomcat-7.0.59.tar.gz

#cp -r apache-tomcat-7.0.59 /data/tomcat{1,2,3,4}

1. 安装apr，apr-iconv，apr-util，tomcat-native

#cd apr-1.5.2

#./configure --prefix=/data/apr

#make ;make install

注：安装apr报错rm: cannot remove `libtoolT': No such file or

直接打开/usr/local/src/apr-1.5.2/configure

注释掉：$RM “$cfgfile”

#cd apr-iconv-1.2.1

#./configure --prefix=/data/apr-iconv --with-apr=/data/apr/

#make ;make install

#cd apr-util-1.5

#./configure --prefix=/data/apr-util --with-apr=/data/apr --with-apr-iconv=/data/apr-iconv/bin/apriconv

#make;make install

#cd tomcat-native-1.1.33-src/jni/native/

#./configure --with-apr=/data/apr --with-java-home=/usr/local/java/

make;make install

添加环境变量(必须是/usr/local/apr/lib这个路径)：

export LD\_LIBRARY\_PATH=$LD\_LIBRARY\_PATH:/usr/local/apr/lib

1. 修改环境变量

#vim /etc/profile

#JAVA

JAVA\_HOME=/usr/local/java

CLASSPATH=$JAVA\_HOME/lib:$JAVA\_HOME/jre/lib

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

export PATH CLASSPATH JAVA\_HOME

export LD\_LIBRARY\_PATH=$LD\_LIBRARY\_PATH:/usr/local/apr/lib

#tomcat1

TOMCAT1\_HOME=/data/tomcat1

CATALINA1\_HOME=/data/tomcat1

CATALINA1\_BASE=/data/tomcat1

export TOMCAT1\_HOME CATALINA1\_HOME CATALINA1\_BASE

#source /etc/profile

1. 修改tomcat配置文件

#vim /data/tomcat/bin/catalina.sh

#添加一行

JAVA\_OPTS="$JAVA\_OPTS -server -Xms2G -Xmx2G -Xmn600M -Xss256k -XX:PermSize=128m -XX:MaxPermSize=256m -XX:SurvivorRatio=1 -XX:MaxTenuringThreshold=10 -XX:+AggressiveOpts -XX:+UseFastAccessorMethods -XX:+UseBiasedLocking -XX:+UseParNewGC -XX:+UseConcMarkSweepGC -XX:CMSInitiatingOccupancyFraction=80 -XX:SoftRefLRUPolicyMSPerMB=0 -XX:+UseCMSCompactAtFullCollection -XX:LargePageSizeInBytes=128m -Dconf\_server\_host=disconf.17money.local -Denv=product"

#vim /data/tomcat1/conf/server.xml

<Connector port="8081"

executor="tomcatThreadPool"

protocol="org.apache.coyote.http11.Http11AprProtocol"

connectionTimeout="20000"

acceptCount="100"

disableUploadTimeout="true"

enableLookups="false"

URIEncoding="UTF-8"

redirectPort="8443" />

1. 启动tomcat

#/data/tomcat/bin/startup.sh

1. **Fastdfs:**

主机名及主机IP



主机信息：CentOS release 6.5 (Final)

版本信息：fastdfs-5.0.5

软件包：fastdfs-5.05.tar.gz

fastdfs-nginx-module\_v1.16.tar.gz

pcre-8.36.tar.gz

zlib-1.2.8.tar.gz

nginx-1.8.0.tar.gz

libfastcommon.zip

环境介绍：

group1:

B52-fdfs-1: 10.10.30.32     tracker1

B52-fdfs-2: 10.10.39.175    tracker2

B52-fdfs-3: 10.10.32.86      storager1    +    nginx(port:9001)

B52-fdfs-4:10.10.33.17       storager2    +    nginx(port:9001)

1. 配置安装环境

首先卸载自带的libevent

#rpm –e --nodeps libevent-1.4.13-4.el6.x86\_64

#cd /root/tmp

#tar zxvf libevent-2.0.21-stable.tar.gz

# cd libevent-2.0.21-stable

#./configure --prefix=/usr/local/libevent

# make && make install

#echo '/usr/local/libevent/lib/' >> /etc/ld.so.conf #加入动态链接库

#echo '/usr/local/libevent/linclude/' >> /etc/ld.so.conf #加入动态链接库

#ldconfig

安装Libfastcommon

#unzip master.zip

#cd libfastcommon-master/

#./make.sh && ./make.sh install

1. 安装Fastdfs

#tar zxvf fastdfs-5.05.tar.gz

# cd fastdfs-5.05

#./make.sh

#./make.sh install

#cd /etc/fdfs/

#cp storage.conf.sample storage.conf

#cp tracker.conf.sample tracker.conf

#cp client.conf.sample client.conf

1. 配置fastdfs

mkdir -p /data/fsdf/{fsdf-sto,fsdf-tra}

然后在有Tracker的服务器上修改/etc/fdfs/tracker.conf

#vi /etc/fdfs/tracker.conf

bind\_addr=10.10.30.32 ##绑定ip

base\_path= /data/fsdf/fsdf-tra ##指定数据目录

reserved\_storage\_space = 10% ##保留存储空间，小于这个值就不能上传

在Storage服务器上修改/etc/fdfs/storage.conf

#vi /etc/fdfs/storage.conf

disabled=false

group\_name=group1 ##组名，根据所在服务器分配相应的组

bind\_addr=10.10.32.86 ##绑定的IP，127.0.0.1不可以

base\_path= /data/fsdf/fsdf-sto ##数据库目录

store\_path0= /data/fsdf/fsdf-sto ##存储目录

tracker\_server=10.10.30.32:22122 ##Tracker服务IP:PORT 多个Tracker服务器，每行一个

tracker\_server=10.10.39.175:22122 ##如上

修改ls /etc/fdfs/client.conf

base\_path= /data/fsdf/fsdf-sto ##数据库目录

tracker\_server=10.10.30.32:22122 ##Tracker服务IP:PORT 多个Tracker服务器，每行一个

tracker\_server=10.10.39.175:22122 ##如上

Note：可以直接把已经安装了FastDfs的服务器上的配置文件复制过去然后修改

然后分别启动Tracker，Storage服务 ##必须先启动Tracker不然会报错

#fdfs\_trackerd /etc/fdfs/tracker.conf

#fdfs\_storaged /etc/fdfs/storage.conf

1. Fastdfs && nginx集成

4.1 配置安装环境

#cd /root/tmp

安装一些需要的包 yum -y install pcre pcre-devel zlib zlib-devel

#tar zxvf fastdfs-nginx-module\_v1.16.tar.gz

#tar zxvf nginx-1.8.0.tar.gz

#vi fastdfs-nginx-module/src/config

ngx\_addon\_name=ngx\_http\_fastdfs\_module

HTTP\_MODULES="$HTTP\_MODULES ngx\_http\_fastdfs\_module"

NGX\_ADDON\_SRCS="$NGX\_ADDON\_SRCS $ngx\_addon\_dir/ngx\_http\_fastdfs\_module.c"

CORE\_INCS="$CORE\_INCS /usr/include/fastdfs /usr/include/fastcommon/" 去掉原来的local

CORE\_LIBS="$CORE\_LIBS -L/usr/lib64 -lfastcommon -lfdfsclient"

CFLAGS="$CFLAGS -D\_FILE\_OFFSET\_BITS=64 -DFDFS\_OUTPUT\_CHUNK\_SIZE='256\*1024' -DFDFS\_MOD\_CONF\_FILENAME='\"/etc/fdfs/mod\_fastdfs.conf\"'"

4.2 安装nginx模块

#cd nginx-1.8.0

#./configure --prefix=/usr/local/nginx --add-module=/root/tmp/fastdfs-nginx-module/src/

#make

#make install

#cp /root/tmp/fastdfs-nginx-module/src/mod\_fastdfs.conf /etc/fdfs/

#vi /etc/fdfs/mod\_fastdfs.conf

connect\_timeout=2

network\_timeout=30

base\_path= /data/fsdf/fsdf-sto

load\_fdfs\_parameters\_from\_tracker=true

storage\_sync\_file\_max\_delay = 86400

use\_storage\_id = false

storage\_ids\_filename = storage\_ids.conf

tracker\_server=10.10.30.32:22122

tracker\_server=10.10.39.175:22122

storage\_server\_port=23000

group\_name=group1 ##更改为相对应的组名

url\_have\_group\_name = true

store\_path\_count=1

store\_path0= /data/fsdf/fsdf-sto

log\_level=info

log\_filename=

response\_mode=proxy

if\_alias\_prefix=

flv\_support = true

再次修改/etc/fdfs/client.conf文件

http.tracker\_server\_port=9001 ###nginx代理端口

最后一行##include http.conf去掉一个#

#include http.conf

做M00的软连接

ln -s /data/fsdf/fsdf-sto/data/ /data/fsdf/fsdf-sto/data/M00

启动Nginx

Service nginx start

Starting nginx: ngx\_http\_fastdfs\_set pid=31641

有提示类似上面的信息即模块加载成功

测试链接：

<http://10.10.32.86:9001/group1/M00/00/00/CgogVlcR8QuAWZZVAAAAIBA2KMc329_big.log>

<http://10.10.33.17:9001/group1/M00/00/00/CgohEVcR8XiAaimjAAAAJyxZr7Q850_big.log>

1. **Zookeeper:**

主机信息：CentOS release 6.5 (Final)

软件版本：zookeeper-3.4.6

软件包：zookeeper-3.4.6.tar.gz

示例主机：

server.1=10.10.112.108:2888:3888

server.2=10.10.111.160:2888:3888

server.3=10.10.101.201:2888:3888

1. 安装zookeeper

#tar zxvf zookeeper-3.4.6.tar.gz  
#mv zookeeper-3.4.6 /data/zookeeper  
#cd /data/zookeeper/  
#mv conf/zoo\_sample.cfg conf/zoo.cfg

2.配置zookeeper集群

tickTime=2000

initLimit=10

syncLimit=5

dataDir=/data/zookeeper/zdata

clientPort=2181

server.1=10.10.112.108:2888:3888

server.2=10.10.111.160:2888:3888

server.3=10.10.101.201:2888:3888

根据上面的server.1 对应的IP地址  新建myid文件，如  
在10.10.112.108上执行  
#echo "1 >> /data/zookeeper/zdata/myid  
其他机器上则  
#echo "2”>> /data/zookeeper/zdata/myid  
#echo "3“>> /data/zookeeper/zdata/myid

3.修改环境变量

#vi /etc/profile  
##ZOOKEEPER  
ZOOKEEPER\_HOME=/data/zookeeper  
export ZOOKEEPER\_HOME  
export PATH=$PATH:$ZOOKEEPER\_HOME/bin:$ZOOKEEPER\_HOME/conf  
#source /etc/profile  
分别启动Zookeeper  
#zkServer.sh start  ##启动Zookeeper

4.查看zookeeper主从信息

启动成功后可查看服务状态就当前等级，等级分别有follower，leader，leader挂了之后会在follower里面推选一个leader  
# zkServer.sh status  
JMX enabled by default  
Using config: /usr/local/zookeeper/bin/../conf/zoo.cfg  
Mode: leader  
或者  
# zkServer.sh status  
JMX enabled by default  
Using config: /usr/local/zookeeper/bin/../conf/zoo.cfg  
Mode: follower

1. **Mongodb**

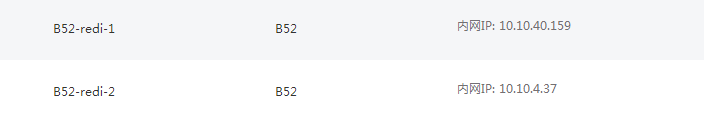
主机信息：CentOS release 6.5 (Final)

Mongodb版本：mongodb-3.0.6

软件包：mongodb-3.0.6-centos6.5.tar.gz  
  
#解压  
# tar zxvf mongodb-3.0.6-centos6.5.tar.gz  
# mv mongodb /usr/local/mongodb  
# mkdir -p /mongodb/data/master  
# cd /usr/local/mongodb/  
# mkdir /etc/mongodb    
# mkdir /mongodb/log  
  
分别在各个服务器上创建相关配置文件  
# vi /etc/mongodb/master.conf  
#master.conf  
dbpath=/mongodb/data/master     ##数据库目录  
logpath=/mongodb/log/  ##日志文件  
pidfilepath=/mongodb/master.pid  ##pid文件  
directoryperdb=true                                ##为每个数据库按照数据库名创建目录  
logappend=true                                         ##以追加的方式记录日志  
replSet=testrs                                             ##replica set的名字  
bind\_ip=10.10.182.53                           ##绑定的IP地址  
port=27017                                                  ##端口  
oplogSize=10000                                               ##mongodb操作日志文件的最大大小。单位为Mb，默认为硬盘剩余空间的5%  
fork=true                                             ##运行在后台  
  
  
#vi /etc/mongodb/slaver.conf  
#slaver.conf  
dbpath=/mongodb/data/slaver  
logpath=/mongodb/log/slaver.log  
pidfilepath=/mongodb/slaver.pid  
directoryperdb=true  
logappend=true  
replSet=testrs  
bind\_ip=10.10.185.73  
port=27017  
oplogSize=10000  
fork=true  
noprealloc=true  
  
#vi  /etc/mongodb/arbiter.conf  
#arbiter.conf  
dbpath=/mongodb/data/arbiter  
logpath=/mongodb/log/arbiter.log  
pidfilepath=/mongodb/arbiter.pid  
directoryperdb=true  
logappend=true  
replSet=testrs  
bind\_ip=10.10.175.32  
port=27017  
oplogSize=10000  
fork=true  
noprealloc=true  
  
然后分别启动mongodb  
/usr/local/mongodb/bin/mongod -f /etc/mongodb/master.conf  
/usr/local/mongodb/bin/mongod -f /etc/mongodb/slaver.conf  
/usr/local/mongodb/bin/mongod -f /etc/mongodb/arbiter.conf  
注意：可能启动需要一段时间  
当提示：child process started successfully, parent exiting   即启动成功  
  
配置主，备，仲裁节点  
可以连接三个节点中的任意一个  
# /usr/local/mongodb/bin/mongo  10.10.175.32:27017  
MongoDB shell version: 3.0.2  
connecting to: 10.10.175.32:27017/test  
Server has startup warnings:   
2015-05-11T16:02:04.623+0800 I CONTROL  [initandlisten] \*\* WARNING: You are running this proce user, which is not recommended.  
2015-05-11T16:02:04.623+0800 I CONTROL  [initandlisten]   
2015-05-11T16:02:04.623+0800 I CONTROL  [initandlisten]   
2015-05-11T16:02:04.623+0800 I CONTROL  [initandlisten] \*\* WARNING: /sys/kernel/mm/transparentled is 'always'.  
2015-05-11T16:02:04.623+0800 I CONTROL  [initandlisten] \*\*        We suggest setting it to 'ne  
2015-05-11T16:02:04.623+0800 I CONTROL  [initandlisten]   
2015-05-11T16:02:04.623+0800 I CONTROL  [initandlisten] \*\* WARNING: /sys/kernel/mm/transparentag is 'always'.  
2015-05-11T16:02:04.623+0800 I CONTROL  [initandlisten] \*\*        We suggest setting it to 'ne  
2015-05-11T16:02:04.623+0800 I CONTROL  [initandlisten]   
>use admin  
>config= {\_id:"testrs", members:  
... [ {\_id:0,host:'10.10.175.32:27017',priority:2},  
... {\_id:1,host:'10.10.182.53:27017',priorit:1},  
... {\_id:2,host:'10.10.185.37:27017',arbiterOnly:true}] };  
##config 可以是任意的名字，priority设置优先级，数字越大的优先级越高，arbiterOnly:true 为设置仲裁节点，并且一定为true，注意大小写  
> rs.initiate(config)  ##初始化设置  
在提示“OK” 后即配置成功  
>rs.status()  ##查看集群状态  
> rs.conf()   ##查看配置文件  
  
State：1表示可以读写，2表示不能读写  
Health：1表示正常，2表示异常  
对于Replica Set来说，Secondary 节点默认是不可读的。  
常用命令  
testrs:PRIMARY> use test;  ##切换到test数据库，没有会自动创建  
testrs:PRIMARY> show dbs;  ##显示数据库  
testrs:PRIMARY>  db.test.find()  ##相当于select \* from database；  
db.test.insert({“name”:”foobar”,”age”:25})  ##插入数据  
  
  
不要仲裁节点：  
参考：  
<http://www.lanceyan.com/tech/mongodb/mongodb_repset1.html>  
  
config = { \_id:"testrs", members:[  
... {\_id:0,host:" 10.10.182.53:27017"},  
... {\_id:1,host:" 10.10.182.53:27018"}]}  
  
rs.initiate(config);  
  
  
  
cfg={ \_id:"deve", members: [ {\_id:0,host:'10.10.185.39:27017',priority:2}, {\_id:1,host:'10.10.182.53:27017',priority:1}]}  
  
rs.reconfig(cfg, {"force":true})

1. **Redis:**

主机名及主机IP：



主机信息：CentOS release 6.5 (Final)

Redis版本：redis-2.8.21

软件包：redis-2.8.21.tar.gz

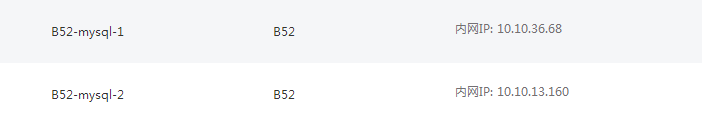
jemalloc-3.6.0.tar.gz

1.安装步骤

安装jemalloc  
cd jemalloc-3.6.0  
./configure  
make; make install  
echo '/usr/local/lib' > /etc/ld.so.conf.d/local.conf  
ldconfig  
  
安装redis  
# tar zxvf redis-3.0.1.tar.gz ;mv redis-3.0.1 /usr/local/redis  
# cd /usr/local/redis/  
# yum install gcc gcc-c++ kernel-devel  
# make MALLOC=libc  
# make install  
  
基本配置：  
daemonize yes  ##后台  
port 7000       #端口  
cluster-node-timeout 5000     ##超时时间  
appendonly yes                        ##每次操作会记录一条log  
  
启动redis服务  
# redis-server /etc/redis/redis.conf  
服务测试  
#redis-cli -p 7000  连接本地  
127.0.0.1:7000> set name abs   
# redis-cli -h 172.16.51.223 -p 7000  连接从服务器  
> get name  
"abs"  
  
创建启动脚本  
在Master：  
安装keepalived  
yum -y install keepalived  
  
定义keepalived配置文件  
! Configuration File for keepalived  
  
global\_defs {  
router\_id LVS\_DEVEL  
}  
vrrp\_script chk\_redis {  
script "/etc/keepalived/scripts/redis\_check.sh"   
interval 2   
}  
  
vrrp\_instance VI\_1 {  
state BACKUP  
interface eth0  
unicast\_peer {  
10.10.40.159  
}  
nopreempt  
virtual\_router\_id 51  
priority 100  
advert\_int 5  
authentication {  
auth\_type PASS  
auth\_pass changeme@2015  
}  
  
track\_script {  
chk\_redis   
}  
virtual\_ipaddress {  
10.10.4.37 dev eth0   
}  
notify\_master /etc/keepalived/scripts/redis\_master.sh  
notify\_backup /etc/keepalived/scripts/redis\_backup.sh  
notify\_fault /etc/keepalived/scripts/redis\_fault.sh  
notify\_stop /etc/keepalived/scripts/redis\_stop.sh   
}  
  
创建keepalive执行脚本文件夹  
mkdir /etc/keepalived/scripts/  
  
创建相关脚本  
vi /etc/keepalived/scripts/redis\_check.sh  
#!/bin/bash  
ALIVE=`/usr/local/bin/redis-cli PING`  
if [ "$ALIVE" == "PONG" ]; then  
echo $ALIVE  
exit 0  
else  
echo $ALIVE  
exit 1  
fi  
  
vi /etc/keepalived/scripts/redis\_master.sh  
#!/bin/bash  
###/etc/keepalived/scripts/redis\_master.sh  
REDISCLI="/usr/local/bin/redis-cli"  
LOGFILE="/var/log/keepalived-redis-state.log"  
pid=$$  
  
echo "`date +'%Y-%m-%d:%H:%M:%S'`|$pid|state:[slaver]" >> $LOGFILE  
echo "`date +'%Y-%m-%d:%H:%M:%S'`|$pid|state:[slaver] Run 'SLAVEOF 10.10.40.159 6379'" >> $LOGFILE  
$REDISCLI SLAVEOF 10.10.40.159 6379 >> $LOGFILE 2>&1  
echo "`date +'%Y-%m-%d:%H:%M:%S'`|$pid|state:[slaver] wait 10 sec for data sync from old master" >> $LOGFILE  
sleep 10  
echo "`date +'%Y-%m-%d:%H:%M:%S'`|$pid|state:[slaver] data rsync from old mater ok..." >> $LOGFILE  
echo "`date +'%Y-%m-%d:%H:%M:%S'`|$pid|state:[master] Run slaveof no one,close master/slave" >> $LOGFILE  
$REDISCLI SLAVEOF NO ONE >> $LOGFILE 2>&1  
echo "`date +'%Y-%m-%d:%H:%M:%S'`|$pid|state:[master] wait other slave connect...." >> $LOGFILE  
  
  
vi /etc/keepalived/scripts/redis\_backup.sh  
#!/bin/bash  
###/etc/keepalived/scripts/redis\_backup.sh  
REDISCLI="/usr/local/bin/redis-cli"  
LOGFILE="/var/log/keepalived-redis-state.log"  
pid=$$  
  
echo "`date +'%Y-%m-%d:%H:%M:%S'`|$pid|state:[master]" >> $LOGFILE  
echo "`date +'%Y-%m-%d:%H:%M:%S'`|$pid|state:[master] Being slave state..." >> $LOGFILE 2>&1  
echo "`date +'%Y-%m-%d:%H:%M:%S'`|$pid|state:[master] wait 10 sec for data sync from old master" >> $LOGFILE  
sleep 10  
echo "`date +'%Y-%m-%d:%H:%M:%S'`|$pid|state:[master] data rsync from old mater ok..." >> $LOGFILE  
echo "`date +'%Y-%m-%d:%H:%M:%S'`|$pid|state:[slaver] Run 'SLAVEOF 10.10.40.159 6379'" >> $LOGFILE  
$REDISCLI SLAVEOF 10.10.40.159 6379 >> $LOGFILE 2>&1  
echo "`date +'%Y-%m-%d:%H:%M:%S'`|$pid|state:[slaver] slave connect to 10.10.40.159 ok..." >> $LOGFILE

1. **Mysql:**

主机名及主机IP：



主机信息：CentOS release 6.5 (Final)

Mysql版本：mysql-5.6.24

软件包：mysql-5.6.24.zip

jemalloc-3.6.0.tar.gz

1.安装步骤：

安装编译环境

# yum -y install patch cmake make gcc gcc-c++ gcc-g77 flex bison file libtool libtool-libs kernel-devel ncurses-devel

2.安装jemalloc

#cd jemalloc-3.6.0

#./configure

#make && make install

#echo '/usr/local/lib' > /etc/ld.so.conf.d/local.conf

#ldconfig

3.编译安装mysql

#mkdir -p /data/mysql/data  
#useradd mysql  
#gourpadd mysql  
#unzip mysql-5.6.24.zip  
#cd mysql-5.6.24  
#cmake -DCMAKE\_INSTALL\_PREFIX=/data/mysql -DMYSQL\_DATADIR=/data/mysql/data -DSYSCONFDIR=/data/mysql -DWITH\_MYISAM\_STORAGE\_ENGINE=1 -DWITH\_INNOBASE\_STORAGE\_ENGINE=1 -DWITH\_MEMORY\_STORAGE\_ENGINE=1 -DWITH\_PARTITION\_STORAGE\_ENGINE=1 -DMYSQL\_UNIX\_ADDR=/data/mysql/sock/mysqld.sock -DDEFAULT\_CHARSET=utf8 -DDEFAULT\_COLLATION=utf8\_general\_ci -DEXTRA\_CHARSETS:STRING=utf8,gbk -DWITH\_DEBUG=0 -DCMAKE\_EXE\_LINKER\_FLAGS="-ljemalloc" -DWITH\_SAFEMALLOC=OFF

#make ;make install

#/data/mysql/scripts/mysql\_install\_db --basedir=/data/mysql --datadir=/data/mysql/data --defaults-file=/data/mysql/my.cnf --user=mysql

#chown -R mysql.mysql /data/mysql/

#cp /data/mysql/support-files/my-default.cnf /etc/my.cnf # 添加mysql配置文件

#cp /data/mysql/support-files/mysql.server /etc/init.d/mysql #添加启动脚本

4.修改环境变量

#vi /root/.bash\_profile #修改环境变量

PATH=$PATH:$HOME/bin:/data /mysql/bin:/data /mysql/lib

#source /root/.bash\_profile

修改配置mysql

#vim /data/mysql/my.cnf

具体配置查阅svn

#mysqladmin -u root password 'jjjr@2015' #设置管理账号密码

#service mysql start #启动mysql

5.配置主从

#mysql –uroot –p

Use mysql

grant all privileges on \*.\* to root@"%" identified by "jjjr@2015";

update user set Password = password('xxxxxx') where User='root';

flush privileges;

Mysql 主从，双方都执行一下命令

create user 'slave'@'%' identified by 'jjjrQWER@2015'; 创建用户

grant replication slave , replication client on \* . \* to 'slave'@'%' identified by 'jjjrQWER@2015'; 授权

一定要在配置完上面两句在做change，不然，什么时候都不成功。

show master status\G;

计算机生成了可选文字:
Pos I t Ion: 
Executed_Gti d _ set : 
bi nlog. 000002 
500 
1 row in set (0.02 sec) 

Change master to master\_host='10.10.36.68',master\_user='slave',master\_port=3306,master\_password='1111',master\_log\_file='binlog.000002',master\_log\_pos=500;

#对方ip,对方的file值,对方的position值

start slave;

show slave status\G;

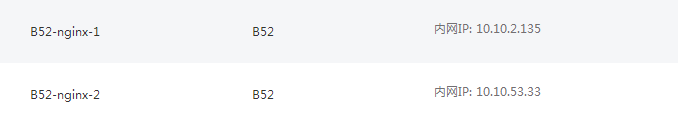
计算机生成了可选文字:
Relay_Master_Log_Fi1e: bi nlog. 000003 
sl ng : 
ng: Ye 

都是yes就说明成功了

Slave\_IO\_Running:connecting 授权有问题用mysql -uslave -h10.10.13.160 -p 测试

1. **Nginx:**

主机名及主机IP：



主机信息：CentOS release 6.5 (Final)

Nginx版本：nginx-1.8.0

软件包：nginx-1.8.0.tar.gz

jemalloc-3.6.0.tar.gz

安装步骤

安装编译环境

yum -y install gcc gcc+ gcc-c++ openssl openssl-devel pcre-devel zlib-devel

useadd www

2． 安装jemalloc

#cd jemalloc-3.6.0

#./configure

#make && make install

#echo '/usr/local/lib' > /etc/ld.so.conf.d/local.conf

#ldconfig

编译安装nginx

#cd nginx-1.8.0

#./configure --prefix=/data/nginx --user=www --group=www --with-http\_stub\_status\_module --with-http\_ssl\_module --with-http\_gzip\_static\_module --with-ld-opt="-ljemalloc" --with-cc-opt='-O2'

#make&make install

配置nginx

#vim /data/nginx/conf/nginx.conf

具体配置见svn

1. 启动nginx

#Service nginx start