部署文档

V 1.0.3

智能安全技术研究中心

**版本更新**

| 日期 | 修订版本 | 修改章节 | 修改描述 | 作者 | 审核 |
| --- | --- | --- | --- | --- | --- |
| 2018.9.27 | V1.0.0 | 全部 | 创建，基于呼和浩特1期 | 朱鑫城 |  |
| 2018.10.23 | V1.0.1 | 4 | 文件服务器安装 | 王健 |  |
| 2018.11.6 | V1.0.2 | 8 | 增加数据平台相关说明 | 王健 |  |
| 2019.01.24 | V1.0.3 | 16 | 增加网关配置说明 | 王健 |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

2台windows （windows-server-2008-R2-企业版）

ip地址:192.168.0.234,192.168.0.235

用户名：Administrator

密码：11111111

4台Ubuntu （Ubuntu14.04）

MySql服务器ip地址:192.168.0.230, 192.168.0.231,

用户名：root0

密码：3edcVFR$

文件服务器ip地址:192.168.0.232, 192.168.0.233

用户名：btdata

密码：bt123456

vnc密码：123456

环境

Java环境 所有机器已安装

Mysql 双主配置，部署在两台Ubuntu上

ip：192.168.0.230, 192.168.0.231

端口：3306

用户名：root

密码：123456

Redis 哨兵模式配置，部署在两台windows上

ip地址:192.168.0.234,192.168.0.235

端口：6379

哨兵节点端口：26379

密码：cigit

RabbitMQ集群，部署在两台windows上

ip地址:192.168.0.234,192.168.0.235

服务端口：5672

web管理平台端口：15672

rabbitmq用户名：root

密码：123456

文件服务器，部署在两台Ubuntu上

ip地址:192.168.0.232,192.168.0.233

服务端口：22122

目录

[1. Java环境部署 5](#_Toc536088736)

[1. Windows 5](#_Toc536088737)

[2. Ubuntu 13](#_Toc536088738)

[2. MySql 15](#_Toc536088739)

[1. MySql安装 15](#_Toc536088740)

[2. MySql双主配置 18](#_Toc536088741)

[3. 修改默认的数据库引擎 22](#_Toc536088742)

[4. 数据调优配置 23](#_Toc536088743)

[5. 修改字符集 23](#_Toc536088744)

[6. 修改最大连接数 24](#_Toc536088745)

[7. binlog和relaylog日志迁移 27](#_Toc536088746)

[3. Redis安装以及哨兵模式配置 32](#_Toc536088747)

[4. 文件服务器fastdfs安装 38](#_Toc536088748)

[5. RabbitMQ安装以及集群配置 43](#_Toc536088749)

[6. Ukey注册 49](#_Toc536088750)

[7. VC++运行环境安装 49](#_Toc536088751)

[8. 全流程安检系统 50](#_Toc536088752)

[9. 全流程安检系统版本更新 52](#_Toc536088753)

[10. 防火墙端口配置 52](#_Toc536088754)

[11. 验证图片是否存储在文件服务器中 59](#_Toc536088755)

[12. 硬盘挂载 62](#_Toc536088756)

[13. redis和redis-sentinel注册成windows服务 64](#_Toc536088757)

[14. Ubuntu 网络配置 64](#_Toc536088758)

[15. 数据平台服务器数据写入 65](#_Toc536088759)

[16. 数据平台网关设置 72](#_Toc536088760)

[17. 网关白名单设置 72](#_Toc536088761)

# Java环境部署

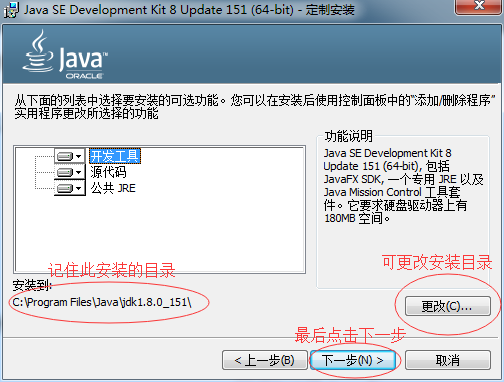
## Windows

**JDK安装**：Java版本为1.8.0\_151，需从官网下载64位版本，直接进行安装，安装过程如下：

1. 双击下载的“jdk-8u151-windows-x64.exe”文件，直接点击“下一步”；



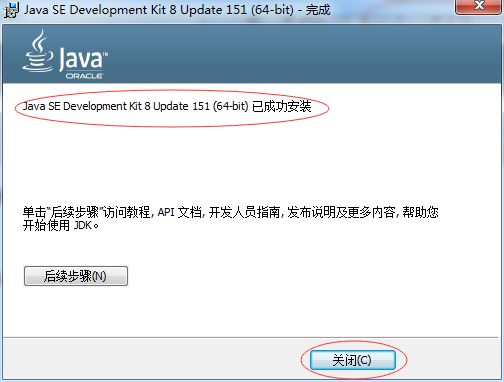
1. 可以选择更改安装目录，一般情况下默认即可，记住安装的路径，点击“下一步”；



1. 目标文件夹一般情况下默认，点击“下一步”进行安装；



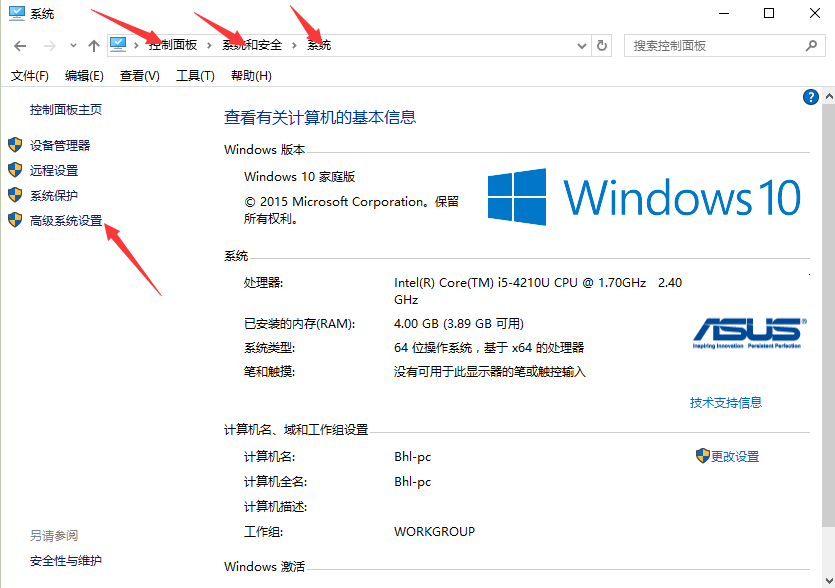
1. 安装完成。安装成功界面如图：



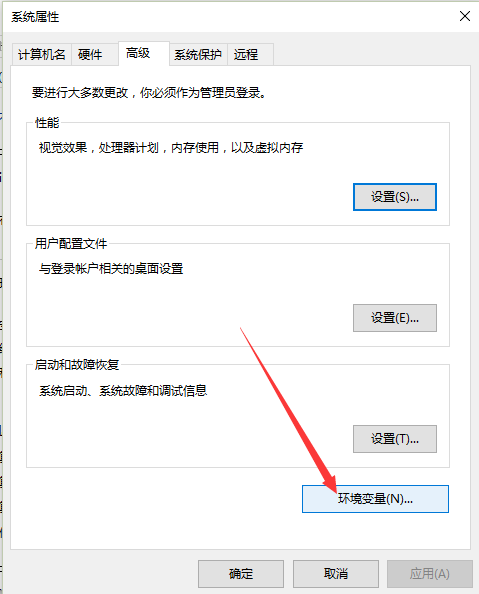
**Java环境配置**：假设JDK安装路径为C:\Program Files\Java。环境配置过程如下：

1. 打开环境变量窗口；

Win10系统打开环境变量窗口方式如下：首先打开控制面板->系统安全->系统->高级系统设置



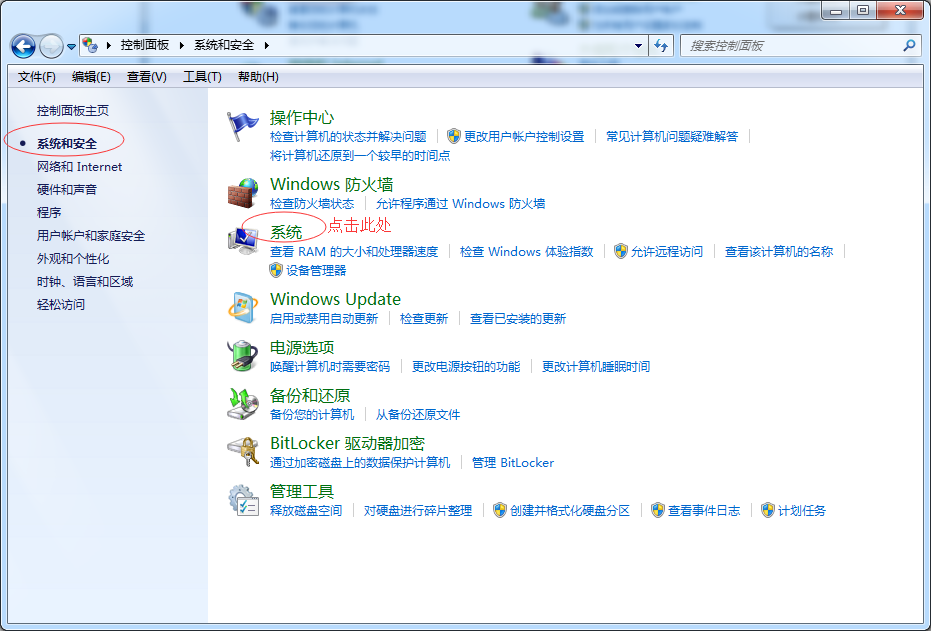
然后点击环境变量



Win7系统打开方式如下：打开“控制面板”之后，找到“系统和安全”，操作如下图：



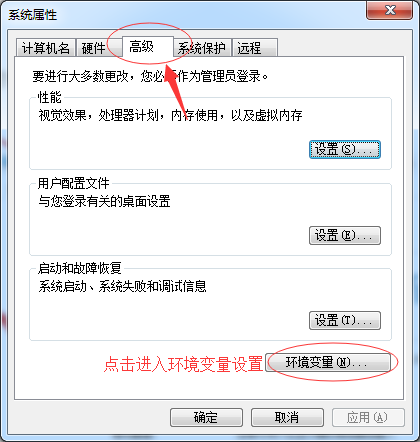
进入“系统和安全”之后，找到“系统”，操作如下图：



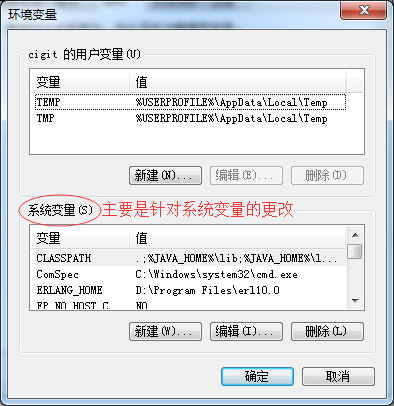
进入“系统”之后，找到“高级系统配置”，操作如下图：



点击“高级系统配置”之后，进入下面页面，找到“环境变量”，具体操作如下图：



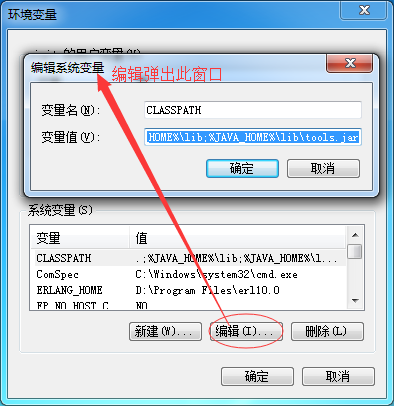
最终进入以下页面：



1. 环境变量配置

可以选择编辑或者新建（存在的就编辑，不存在就新建）；

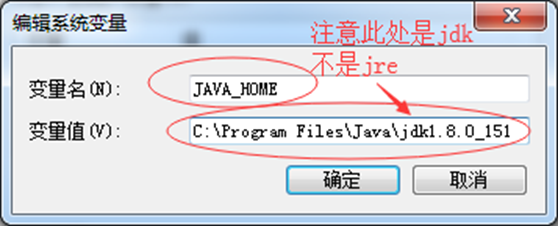




然后在"系统变量"中设置3项属性，JAVA\_HOME,PATH,CLASSPATH(大小写无所谓),若已存在则点击"编辑"，不存在则点击"新建"。

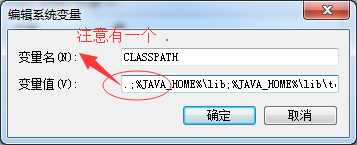
变量设置参数如下：

变量名：JAVA\_HOME。变量值：C:\Program Files\Java\jdk1.8.0\_151

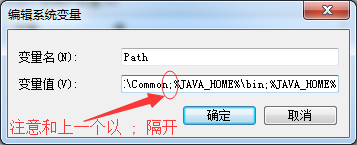


**注意**：这个路径就是自己安装JDK安装包的路径，已当时机器的安装路径为准。

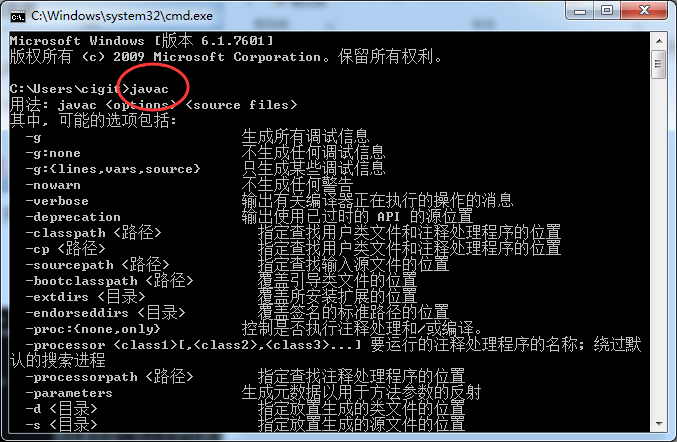
变量名：CLASSPATH变量值：.;%JAVA\_HOME%/lib/dt.jar;%JAVA\_HOME%/lib/tools.jar;记得前面有个"."



变量名：Path 变量值：%JAVA\_HOME%/bin;%JAVA\_HOME%/jre/bin;



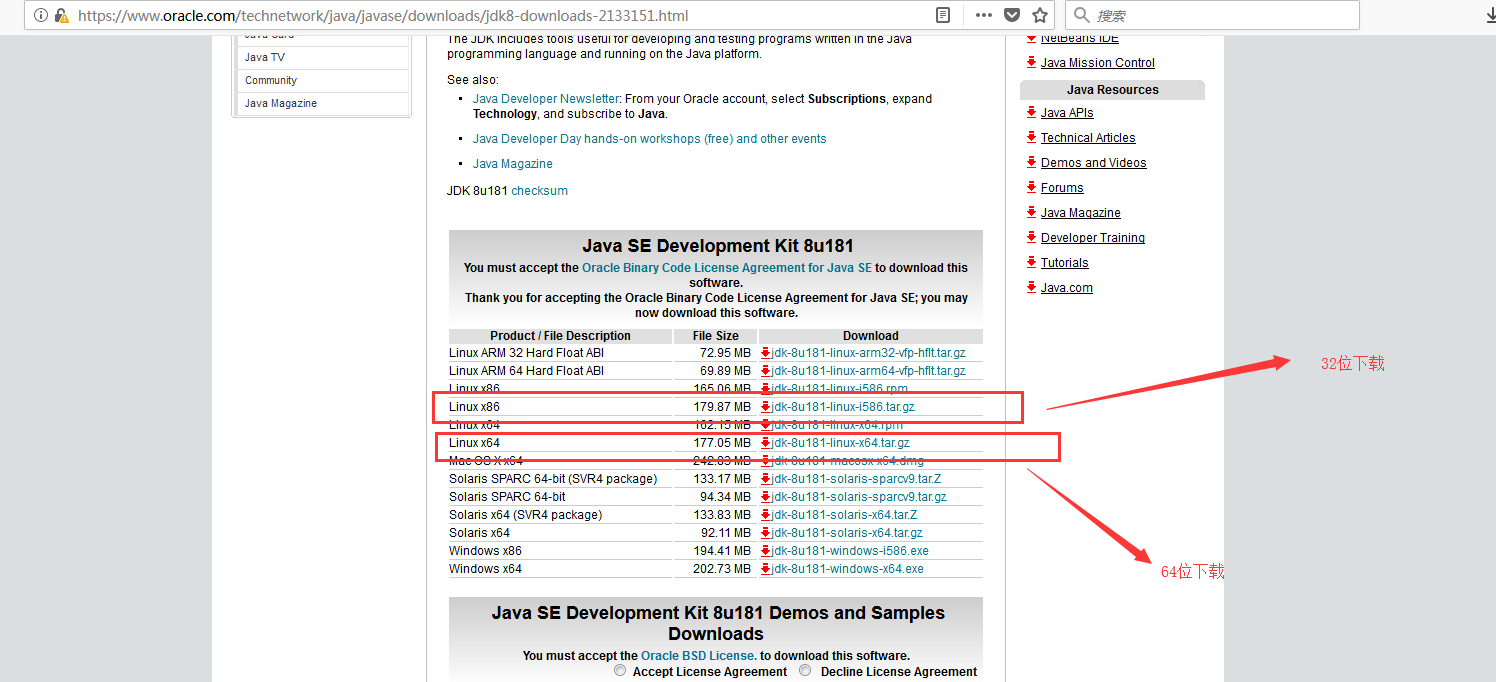
最后一步进行验证，打开windows 控制台，输入javac如出现下图 ，则表明java环境配置成功。



## Ubuntu

1. 下载安装包

官网地址：<https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>



1. 上传至linux 服务器中，在压缩包的目录下使用tar 指令解压缩

tar –zxvf 压缩包名称



1. 使用pwd命令查看当前路径，这里我是解压到java文件夹中的，所以我的路径是这样



1. 配置环境变量

sudo vim /etc/profile

在文档最后添加如下内容 ，编辑完成后 按esc 然后按冒号 输入wq 保存并退出

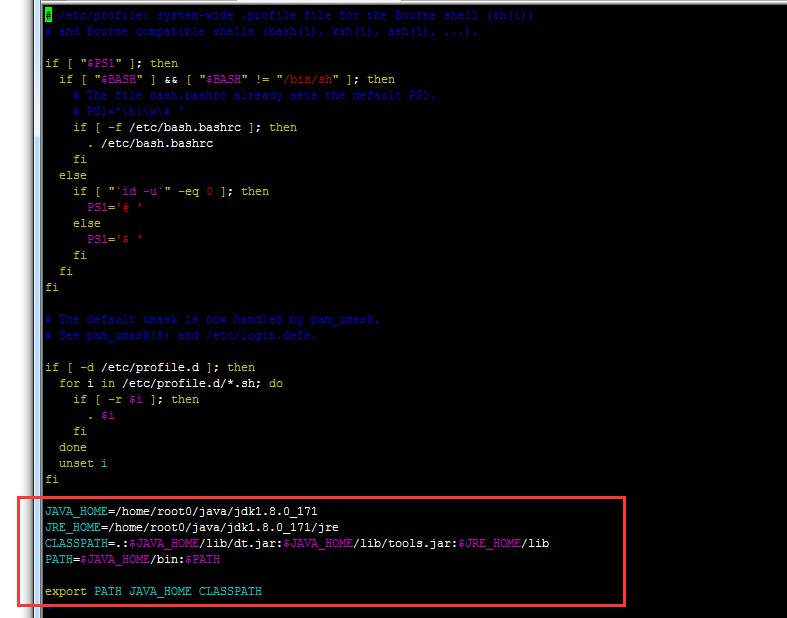
JAVA\_HOME=java的解压路径

JRE\_HOME=$JAVA\_HOME/jre

CLASSPATH=.:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar:$JRE\_HOME/lib

PATH=$JAVA\_HOME/bin:$PATH

export PATH JAVA\_HOME CLASSPATH



1. 使用 source /etc/profile 让刚刚的配置生效



1. 检查安装是否成功

输入javac 如出现下图，则表示安装成功



如果有多台机器需要安装的情况则可以使用scp命令，将安装包和环境变量配置拷贝到目标机器上即可。这里使用192.168.0.231举例

-r 表示发送的是文件夹

第一个参数是要发送的文件夹的位置

@前面的是用户名

@后面的是ip地址

：后面的是拷贝过去的目录路径

$PWD 表示与当前路一致

scp -r /home/root0/java root0@192.168.0.231:$PWD

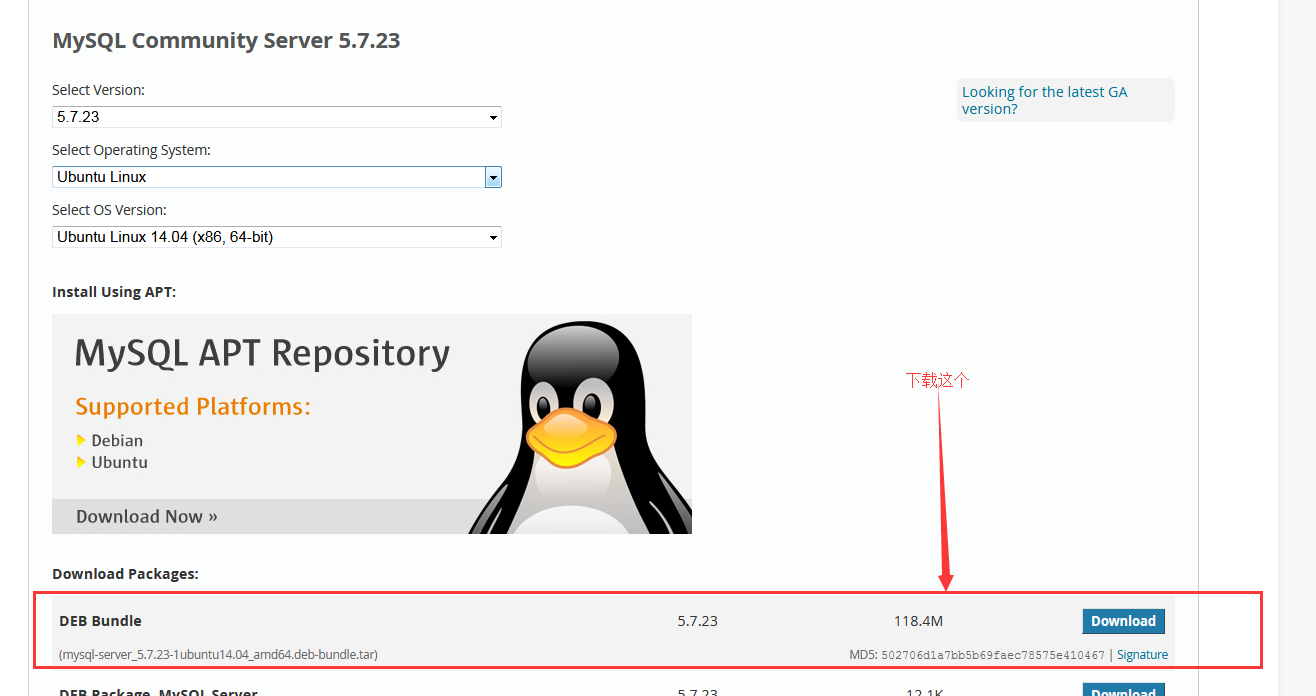
然后 按照步骤4,5配置环境变量

# MySql

## MySql安装

1. 首先在官网下载mysql5.7.21

官网地址：https://dev.mysql.com/downloads/mysql/5.7.html#downloads



1. 上传至linux服务器中，在文件的目录下，使用tar指令解压

sudo tar -xvf mysql-server\_5.7.23-1ubuntu14.04\_amd64.deb-bundle.tar



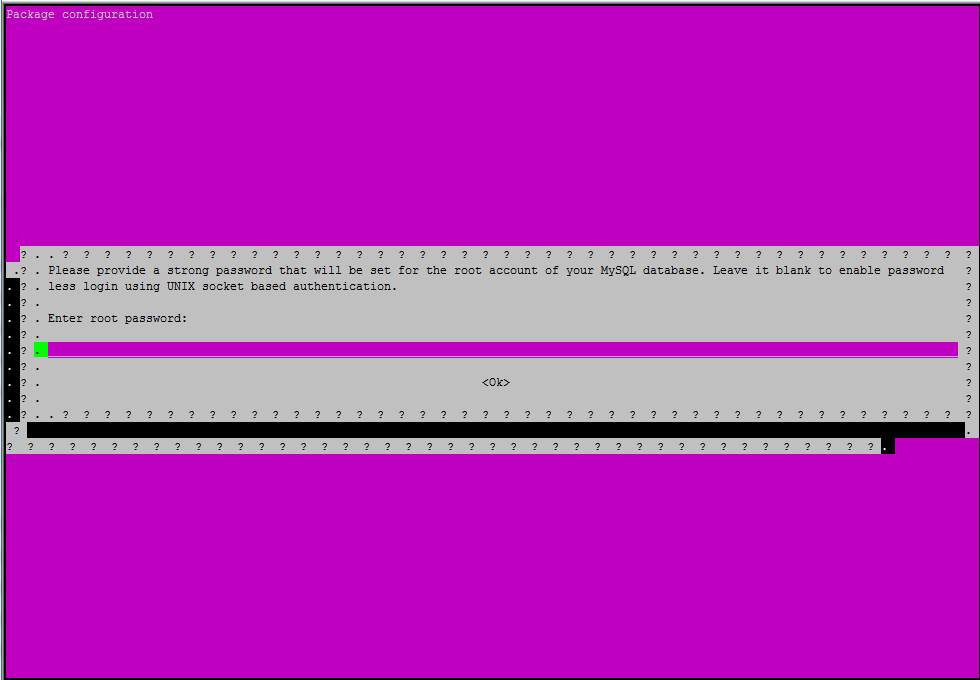
1. 安装所需的依赖环境

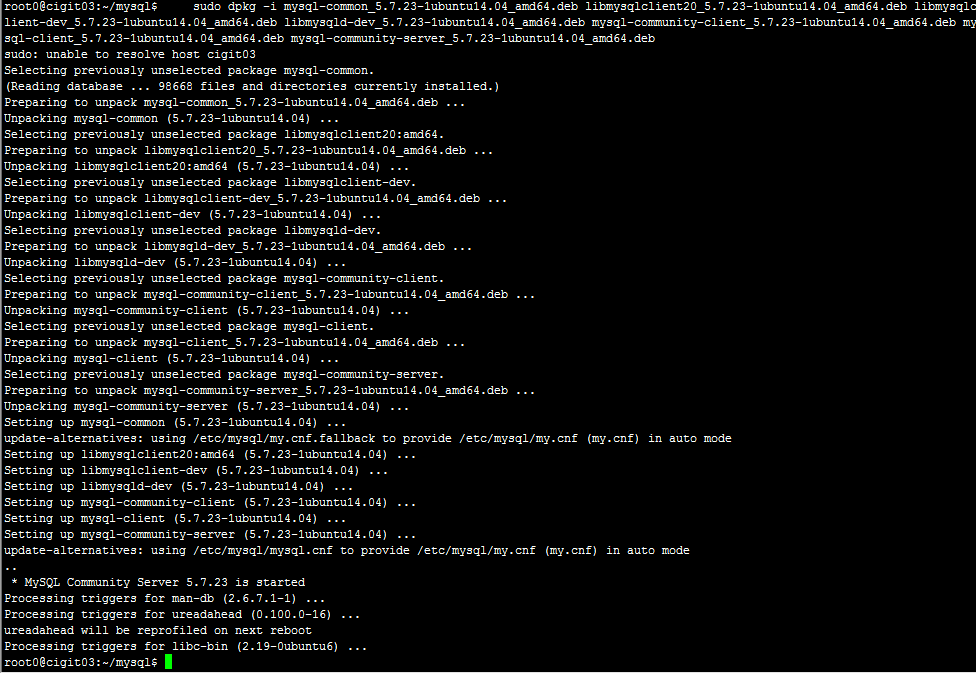
sudo apt-get install –y libmecab2 libaio1

1. 开始安装

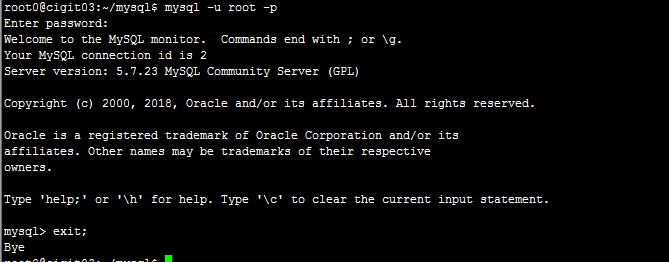
sudo dpkg -i mysql-common\_5.7.23-1ubuntu14.04\_amd64.deb libmysqlclient20\_5.7.23-1ubuntu14.04\_amd64.deb libmysqlclient-dev\_5.7.23-1ubuntu14.04\_amd64.deb libmysqld-dev\_5.7.23-1ubuntu14.04\_amd64.deb mysql-community-client\_5.7.23-1ubuntu14.04\_amd64.deb mysql-client\_5.7.23-1ubuntu14.04\_amd64.deb mysql-community-server\_5.7.23-1ubuntu14.04\_amd64.deb

安装过程中按照提示设置root 密码，这里设置为123456





1. 使用mysql –u root –p 指令进入mysql，如能进入，表示安装成功，输入exit; 退出



## MySql双主配置

服务器：

192.168.0.230

192.168.0.231

两台机器都已按照上述方式安装了MySql5.7 用户root 密码123456

1. 首先给两台机器相互授权

进入192.168.0.230 的mysql 客户端

mysql -u root -p

输入密码进入

然后执行

grant replication slave on \*.\* to 'root'@'192.168.0.231' identified by '123456';

这样就给192.168.0.231机器上 用户名为root 密码为123456的用户授予了访问权限

然后在192.168.0.231上进行相同的操作，把IP地址修改一下

grant replication slave on \*.\* to 'root'@'192.168.0.230' identified by '123456';

1. 检查是否成功

在 192.168.0.231 上 使用 mysql -h 192.168.0.230 –u root –p 进行登录

若能成功登录表示配置成功。

若出现

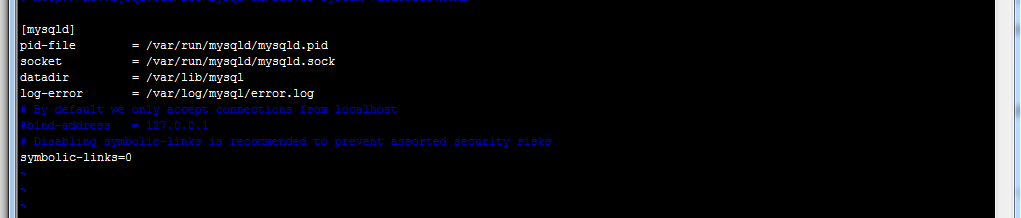
ERROR 2003 (HY000): Can't connect to MySQL server on '192.168.0.230' (111)

则需要修改配置文件

vim /etc/mysql/mysql.conf.d/mysql.cnf

将 bind-address=127.0.0.1 注释掉

如图：



1. 双主配置

修改192.168.0.230 /etc/mysql/my.cnf 文件

sudo vim /etc/mysql/my.cnf

增加内容如下：

[mysqld]

log\_bin = mysql-bin

binlog\_format = mixed

server\_id = 1

read\_only=0

binlog\_do\_db=htbusyinfo

binlog\_do\_db=htdatainfo

binlog\_ignore\_db=information\_schema

binlog\_ignore\_db=mysql

binlog\_ignore\_db=performance\_schema

binlog\_ignore\_db=sys

replicate\_do\_db=htbusyinfo

replicate\_do\_db=htbaseinfo

replicate\_ignore\_db=information\_schema

replicate\_ignore\_db=mysql

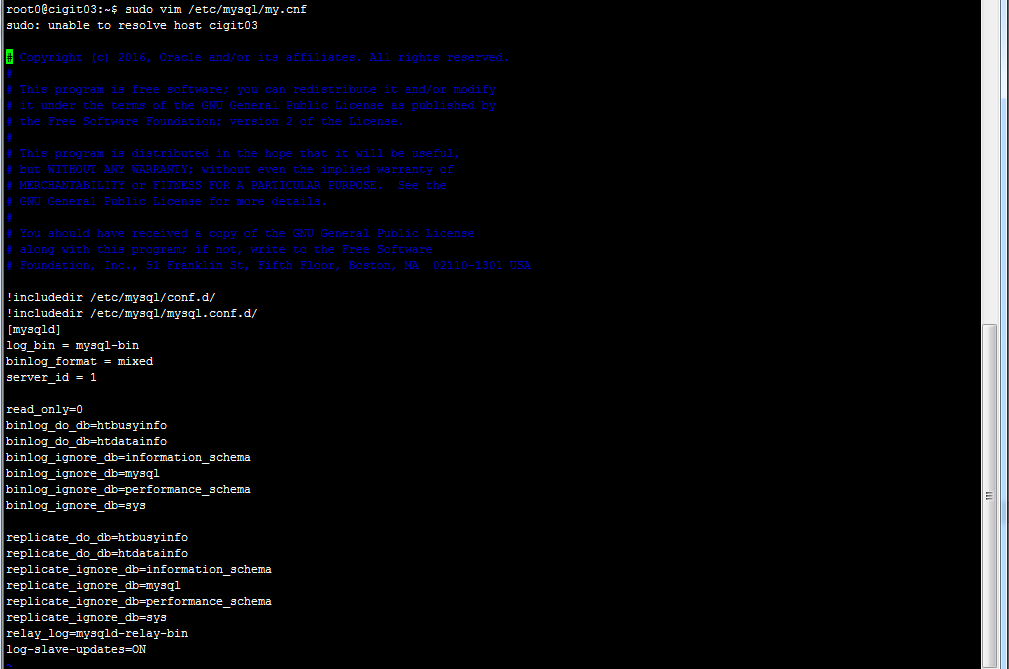
replicate\_ignore\_db=performance\_schema

replicate\_ignore\_db=sys

relay\_log=mysqld-relay-bin

log-slave-updates=ON

如图：



解释一下上面的内容：

binlog\_do\_db 用来表示，只把哪些数据库的改动记录到binary日志中。

binlog-ignore-db 表示，需要忽略哪些数据库。

server-id 必须保证每个服务器不一样

replicate-do-db 可以指定需要复制的数据库

replicate-ignore-db 复制时需要排除的数据库

relay\_log 中继日志的名字

log-slave-updates 意思是中继日志执行之后，这些变化是否需要计入自己的binarylog。

保存并退出

重启mysql

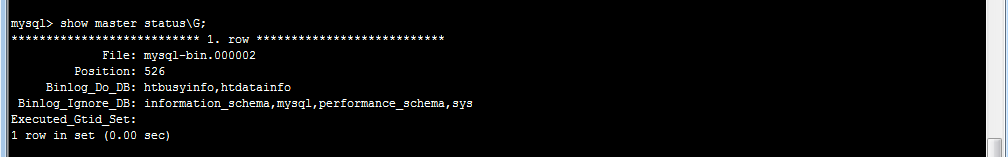
sudo service mysql restart

同理，在192.168.0.231的机器上进行同样的操作，将server\_id=1 改为server\_id=2。然后重启mysql。

在192.168.0.230上进入mysql 客户端

然后输入

show master status\G;



会出现上图内容。

然后在192.168.0.231机器上执行

stop slave;

然后执行：

CHANGE MASTER TO

MASTER\_HOST='192.168.0.230',

MASTER\_USER='root',

MASTER\_PASSWORD='123456',

MASTER\_LOG\_FILE='mysql-bin.000002',

MASTER\_LOG\_POS=526;

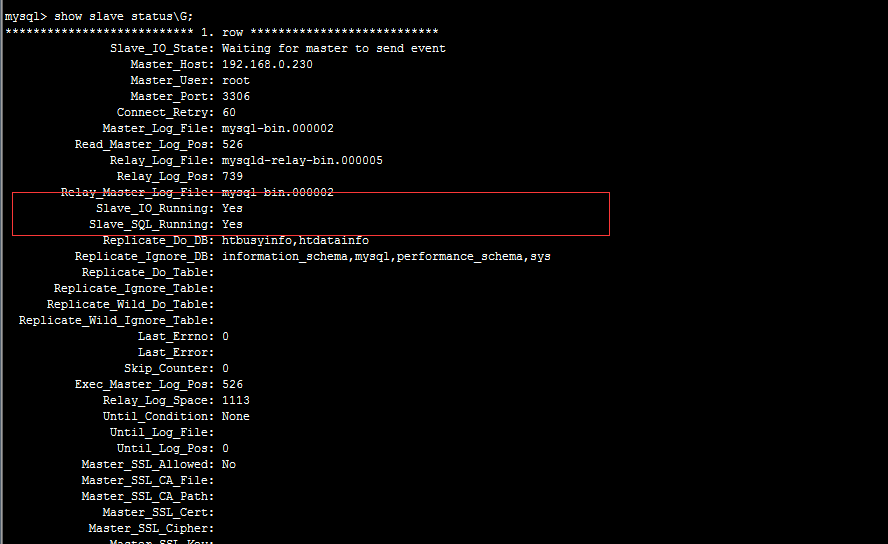
注意这里 ip地址应与master机器的ip地址一致。master\_user 和master\_password 是我们刚刚授权的用户名和密码。master\_log\_file是图中的File。master\_log\_pos是图中的Position。

输入之后，再在192.168.0.231上执行

start slave;

然后输入

show slave status\G;



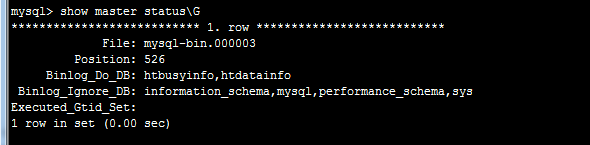
若Slave\_IO\_Running和Slave\_SQL\_Runing的值都为yes则表示以192.168.0.230为主，192.168.0.231为从配置成功

接下来配置以192.168.0.231为主，以192.168.0.230为从。

在上面的步骤中，已经修改了192.168.0.231的my.cnf文件，所以只需要完成后面的操作。

在192.168.0.231上进入mysql 客户端

输入show master status\G;



然后在192.168.0.230上执行

stop slave;

然后执行：

CHANGE MASTER TO

MASTER\_HOST='192.168.0.231',

MASTER\_USER='root',

MASTER\_PASSWORD='123456',

MASTER\_LOG\_FILE='mysql-bin.000003',

MASTER\_LOG\_POS=526;

然后执行：

start slave;

然后输入

show slave status\G;

若Slave\_IO\_Running和Slave\_SQL\_Runing的值都为yes则表示以192.168.0.231为主，192.168.0.230为从配置成功

这个时候双主节点配置完成。

切换数据库引擎为MyISAM

mysql默认引擎为Innodb 所以我们需要切换为MyISAM

修改/etc/mysql/my.cnf 文件

在[mysqld] 下修改\增加为如下内容

default-storage-engine=MyISAM

保存退出，重启mysql服务即可

## 修改默认的数据库引擎

修改mysql配置文件

使用root权限执行命令

sudo vim /etc/mysql/my.cnf

会进入到文件编辑界面

在[mysqld]下添加或修改

default-storage-engine=MyISAM

## 数据调优配置

修改mysql配置文件

使用root权限执行命令

sudo vim /etc/mysql/my.cnf

会进入到文件编辑界面

在[mysqld]下添加或修改

bulk\_insert\_buffer\_size=120M

max\_allowed\_packet=4M

net\_buffer\_length=8k

concurrent\_insert=2

event\_scheduler=ON

重启mysql使配置生效

加入事件，2个小时运行下

analyze table face\_review\_features

## 修改字符集

1. 修改mysql配置文件

使用root权限执行命令

sudo vim /etc/mysql/my.cnf

会进入到文件编辑界面

在[mysql]下添加或修改default-character-set=utf8

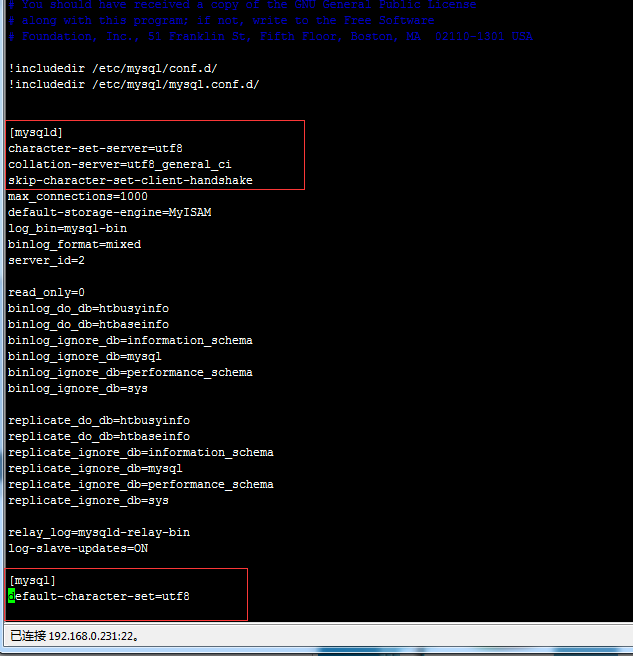
在[mysqld]下添加或修改

character-set-server=utf8

collation-server=utf8\_general\_ci

skip-character-set-client-handshake

如图：



1. 重启mysql,使配置生效

service mysql restart

## 修改最大连接数

1. 修改mysql配置文件

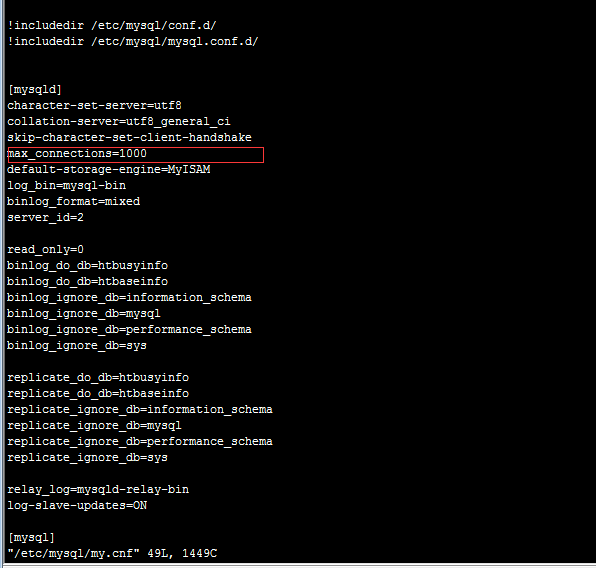
使用root权限执行命令

sudo vim /etc/mysql/my.cnf

会进入到文件编辑界面

在 [mysqld] 下添加或修改max\_connections

如图：



修改红框中的值，然后保存并退出

1. 修改系统配置文件并重启

因为mysql的最大连接数会受到系统的限制，所以需要修改系统的配置来让设置的mysql最大连接数生效

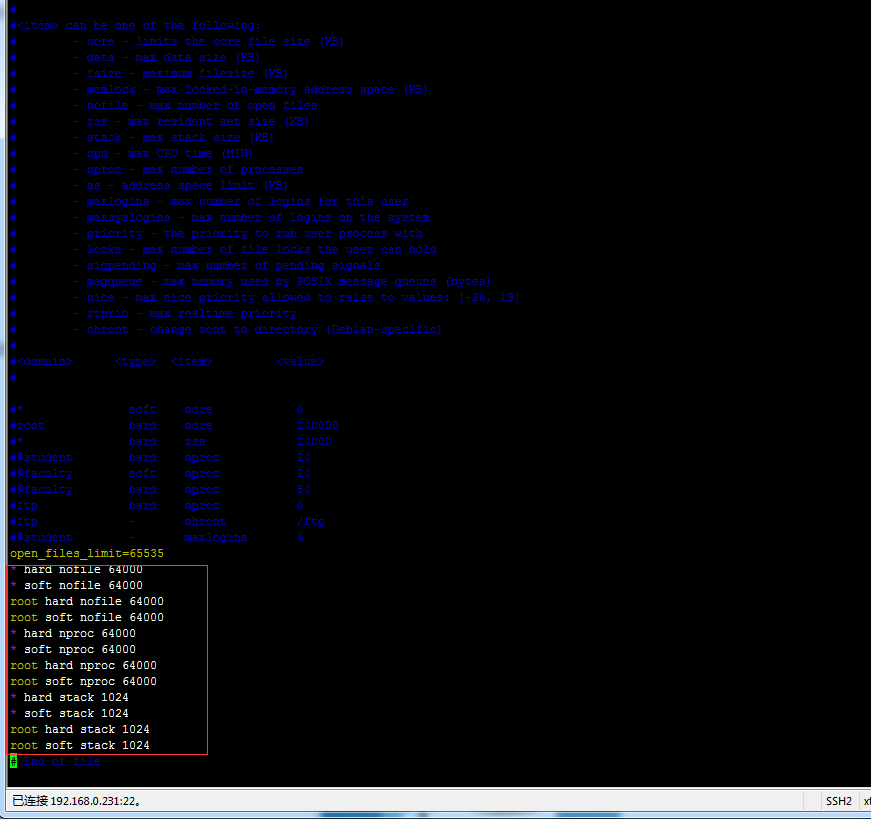
使用root权限执行命令

sudo vim /etc/security/limits.conf

在文件最后添加下列内容：

\* hard nofile 64000  
\* soft nofile 64000  
root hard nofile 64000  
root soft nofile 64000  
\* hard nproc 64000  
\* soft nproc 64000  
root hard nproc 64000  
root soft nproc 64000  
\* hard stack 1024  
\* soft stack 1024  
root hard stack 1024  
root soft stack 1024

如图：



保存并退出

然后

sudo vim /etc/pam.d/su

将pam\_limits.so这一行注释掉

然后

sudo vim /etc/pam.d/common-session

在最后加上

session require pam\_limits.so

最后配置 /etc/profile

在最后一行加上

ulimit –SHn 40960

输入下面的命令重启系统，让刚刚的配置生效

sudo reboot

1. 重启mysql

系统重启之后，输入下面的命令重启mysql

sudo service mysql restart

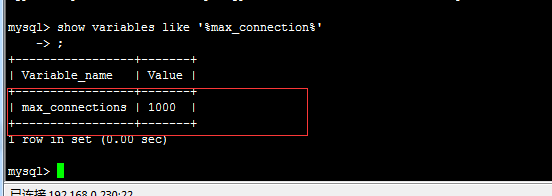
然后进入mysql 客户端

mysql -u root -p 这一步需要输入密码

进入客户端后，输入命令查看最大连接数

show variables like '%max\_connection%';

如图：



可以看到当前最大连接数是1000

## binlog和relaylog日志迁移

1. 停止mysql服务

使用命令 sudo service mysql stop 停止mysql服务

1. 新建存放日志的文件并迁移日志

在根目录下新建文件夹

mkdir /mysqllog

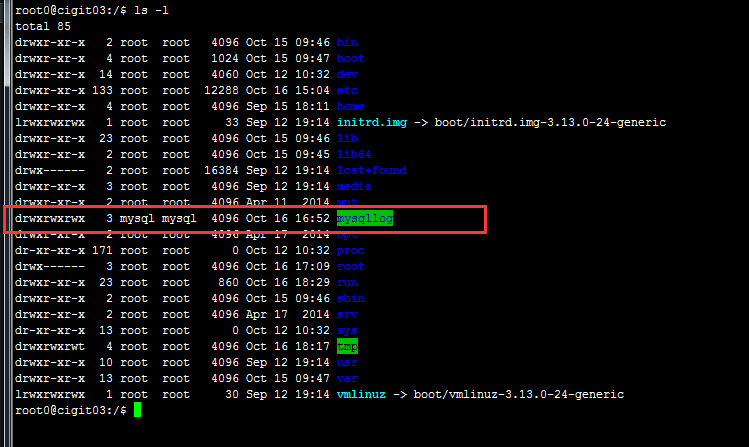
因为mysql的日志文件目录需要由mysql用户来操作，所以我们将该目录的用户和用户组改为mysql

chown –R mysql:mysql /mysqllog

然后赋予读写权限

chmod a+rwx /mysqllog

然后在根目录下使用命令 ls -l 查看文件夹权限是否修改成功



1. 迁移日志

将原目录的所有文件拷贝至新文件夹中

binlog和relaylog的位置在/var/lib/mysql目录下

sudo cp -rp /var/lib/mysql/ /mysqllog/

日志迁移完成

1. 修改my.cnf配置文件

mysql 的默认读取配置文件是

/etc/my.cnf > /etc/mysql/my.cnf

所以优先修改/etc/my.cnf 如果/etc下没有这个配置文件，则修改/etc/mysql/my.cnf

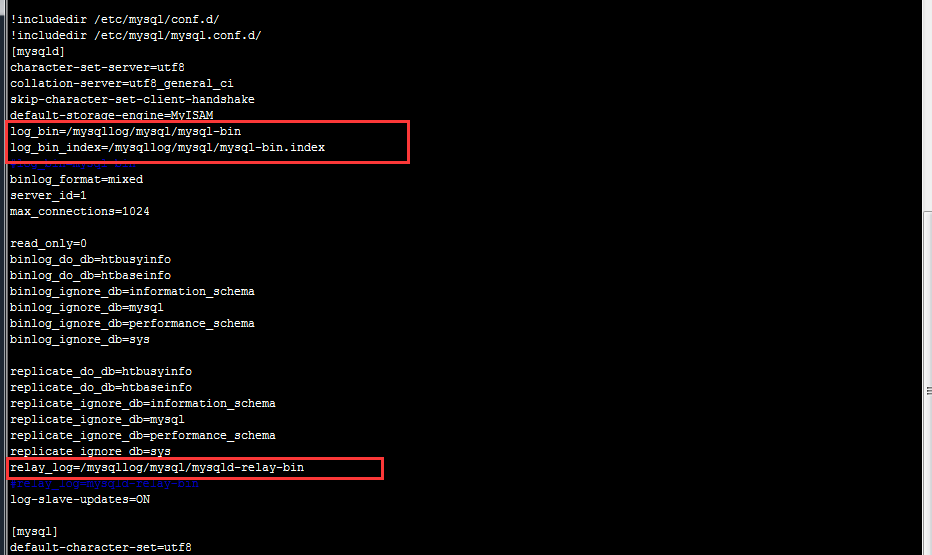
sudo vim /etc/mysql/my.cnf

如图



然后将log\_bin 以及relay\_log的修改为新的路径

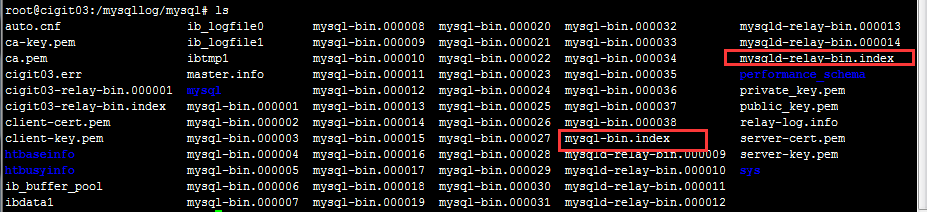
如图



1. 修改binlog路径

mysql主从备份是按照index文件中的文件目录来进行读写的。

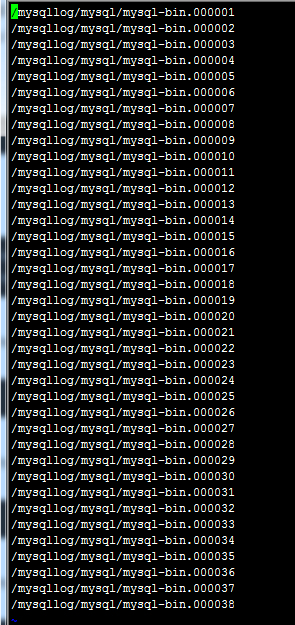
所以我们要修改迁移后的index文件



sudo vim mysql-bin.index

将里面的相对路径全部修改为新的路径（并且是绝对路径）

如图

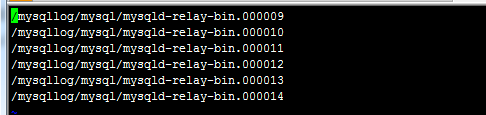


1. 修改relaylog路径

图同上

sudo vim mysqld-relay-bin.index

将里面的相对路径全部修改为新的路径（并且是绝对路径）如图



1. 修改master.info 和relay.info

这两个文件记录的是正在处理的文件信息

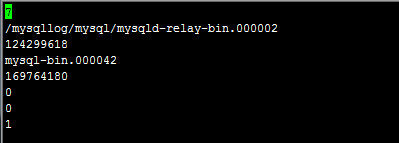
同样也是修改路径

sudo vim master.info

如图，将相对路径修改为新的绝对路径



sudo vim relay-log.info



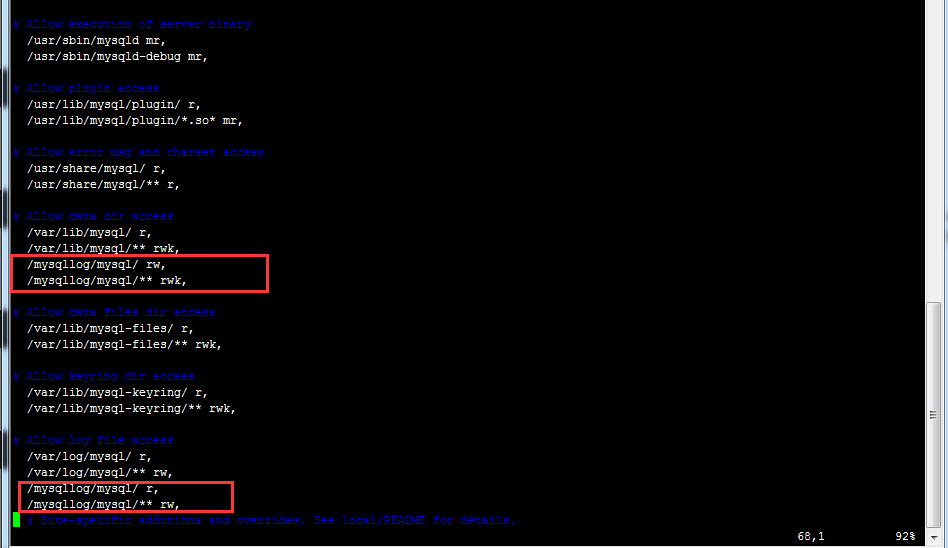
1. 修改apparmor程序对mysql的权限控制

按照正常流程，到上一步就应该结束了，但是我们的系统是Ubuntu，Ubuntu有一个程序apparmor专门来管理其他程序的对文件的读写权限，如果我们不修改配置就会一直报没有权限的错误，而启动不了mysql。

修改该文件对mysql的权限配置

sudo vim /etc/apparmor.d/usr.sbin.mysqld

如图：



可以看到图中的蓝色字，allow xxx acces

根据这些注释我们需要修改

allow log file access下的目录和allow data dir access下的目录

按照他的写法，我们把新的路径添加进去，如上图

然后保存并退出

最后重启一下 apparmor

sudo service apparmor restart

1. 重启mysql

sudo service mysql restart

启动成功表示配置正常。

没有启动成功，则查看错误日志，位置在/var/log/mysql/error.log

# Redis安装以及哨兵模式配置

以下列服务器为例（windows-server-2008-R2-企业版）：

192.168.0.234

192.168.0.235

所需软件

redis.3.2

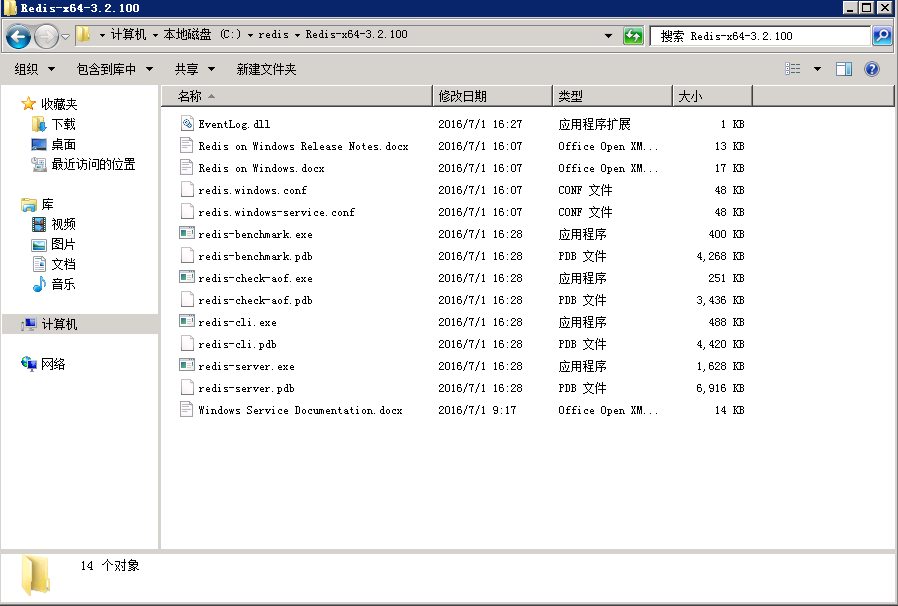


下载地址：

部署步骤：

配置主节点（以192.168.0.234为主节点）：

1. 解压压缩包，解压后内容如下：



1. 修改配置文件 redis.windows[[1]](#endnote-0).conf，修改内容如下（IP地址根据实际情况做修改）：

port 6379

requirepass cigit

bind 192.168.0.234 //此ip地址修改为自己电脑的IP地址

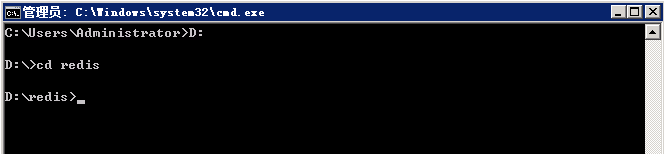
protected-mode yes

masterauth cigit

masterauth 当该节点为主节点时无用，当该节点为从节点时，会尝试连接主节点，需要提供主节点的密码

1. 启动主节点

在命令行窗口中进入到该redis的安装目录中，如图（我是安装到D:/redis）

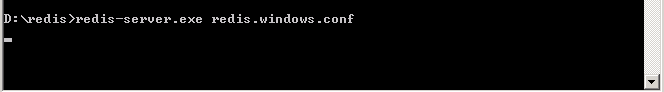


然后输入

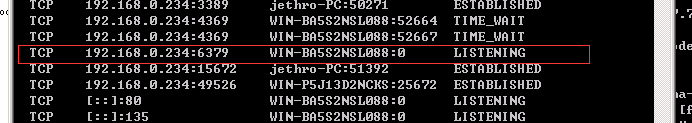
redis-server.exe redis.windows.conf

表示以这个配置文件的配置来启动redis

如图



然后另开一个命令行窗口 输入 nestat –a 可以看到



表示主节点启动成功

配置从节点（以192.168.0.235为从节点）：

1. 步骤1同 主节点配置的步骤1
2. 修改配置文件redis.windows.conf

内容如下：

port 6379

bind 192.168.0.235 //此ip地址修改为自己电脑的IP地址

slaveof 192.168.0.234 6379

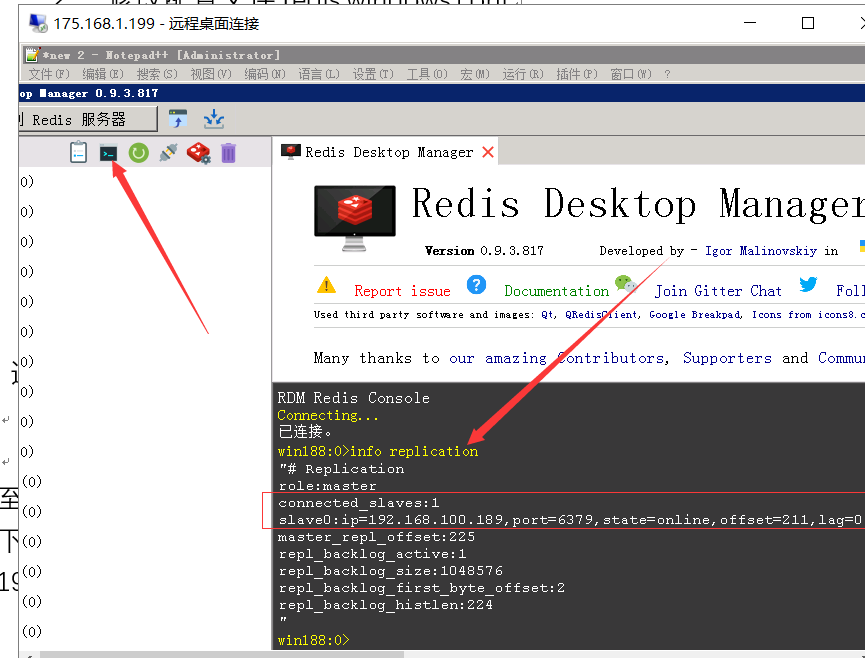
protected-mode no

requirepass cigit

masterauth cigit

1. 启动从节点 方法同启动主节点

通过redismanage 调用命令：info replication查看当前状态



至此主从配置完成

下面配置哨兵，需要在两台机器上都配置哨兵节点

192.168.0.234：

1. 在redis的安装目录中新建哨兵的配置文件 sentinel-26379.conf

内容如下：

#Sentinel 节点的端口

port 26379

dir ./

logfile ""

protected-mode no

bind 192.168.0.234

#这个节点监控192.168.0.234:6379这个主节点

# 1代表判断主节点失败至少需要1个哨兵节点同意

# mymaster 是给主节点起的别名

sentinel monitor mymaster 192.168.0.234 6379 1

sentinel auth-pass mymaster cigit

#每个Sentinel节点都要定期PING命令来判断Redis数据节点和其余Sentinel节

#点是否可达，如果超过5000毫秒且没有回复，则判定不可达

sentinel down-after-milliseconds mymaster 5000

#当Sentinel节点集合对主节点故障判定达成一致时，Sentinel领导者节点会做故

#障转移操作，选出新的主节点，原来的从节点会向新的主节点发起复制操作，限#制每次向新的主节点发起复制操作的从节点个数为1

sentinel parallel-syncs mymaster 1

#故障转移超时时间为180000毫秒

sentinel failover-timeout mymaster 180000

然后在192.168.0.235机器上

在redis的安装目录中新建哨兵的配置文件 sentinel-26379.conf

内容如下

#Sentinel 节点的端口

port 26379

dir ./

logfile ""

protected-mode no

bind 192.168.0.235

#这个节点监控192.168.0.234:6379这个主节点

# 1代表判断主节点失败至少需要1个哨兵节点同意

# mymaster 是给主节点起的别名

sentinel monitor mymaster 192.168.0.234 6379 1

sentinel auth-pass mymaster cigit

#每个Sentinel节点都要定期PING命令来判断Redis数据节点和其余Sentinel节

#点是否可达，如果超过5000毫秒且没有回复，则判定不可达

sentinel down-after-milliseconds mymaster 5000

#当Sentinel节点集合对主节点故障判定达成一致时，Sentinel领导者节点会做故

#障转移操作，选出新的主节点，原来的从节点会向新的主节点发起复制操作，限#制每次向新的主节点发起复制操作的从节点个数为1

sentinel parallel-syncs mymaster 1

#故障转移超时时间为180000毫秒

sentinel failover-timeout mymaster 180000

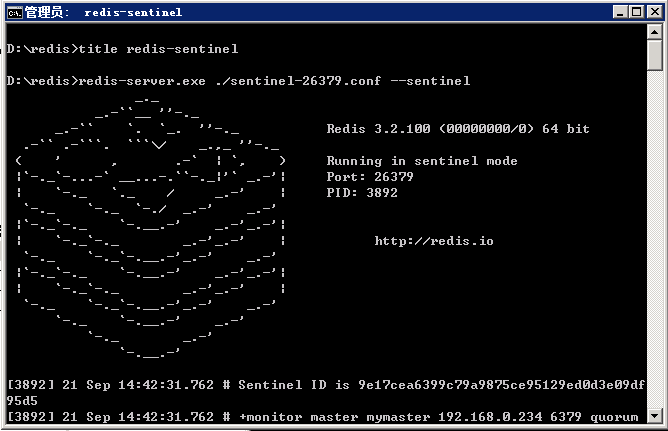
1. 启动哨兵节点

在命令行窗口中进入到该redis的安装目录中，如图（我是安装到D:/redis）

输入 redis-server.exe sentinel-26379.conf –sentinel

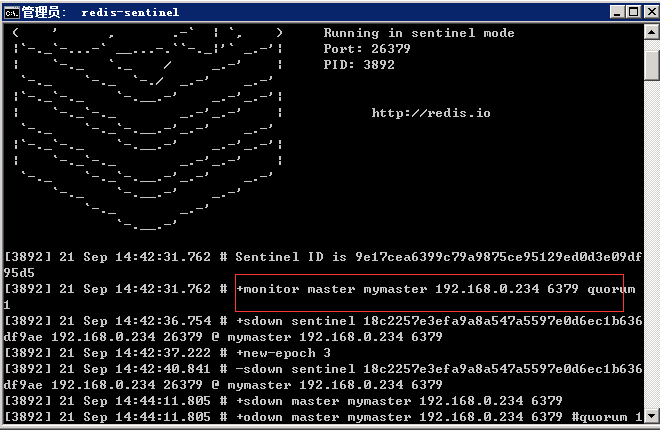
最小化命令行窗口。

启动成功如图：

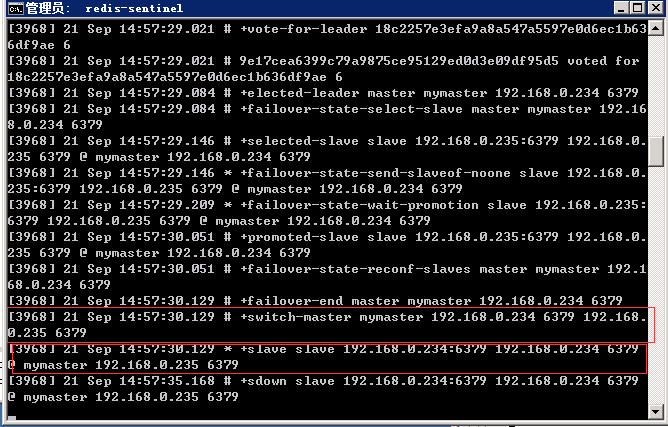


1. 检查是否配置成功

在两台机器都配置好哨兵节点后。可以看到哨兵节点的控制台的输出如下图：



现在主节点时192.168.0.234，我们关掉234机器上的redis后可以看到哨兵控制台的输出。如下图



红圈的意思就是 切换当前的主节点 从192.168.0.234 切换到192.168.235

当我们重新启动后，可以在管理控制台看到。

进入管理控制台的办法是

在redis的安装目录下 进入命令行窗口

输入

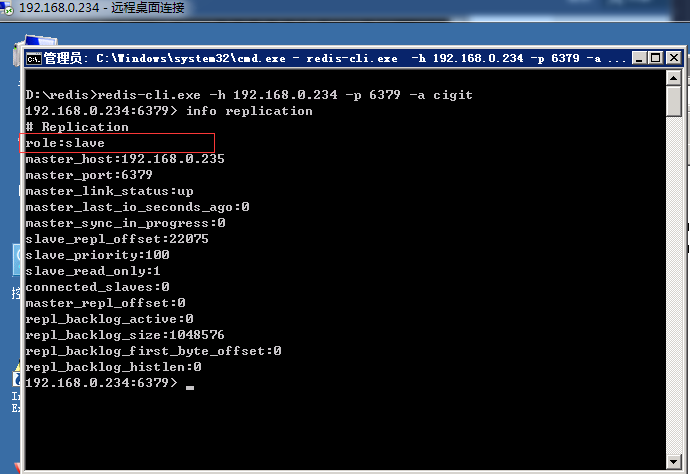
redis-cli.exe –h 192.168.0.234 –p 6379 –a cigit

这样就进入了管理控制台

输入

info replication 可以看到当前节点信息。

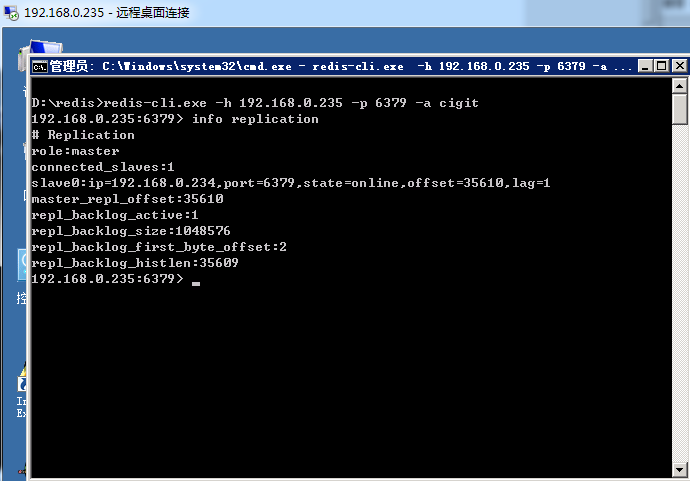
如图



可以看到红圈部分 当前节点是从节点

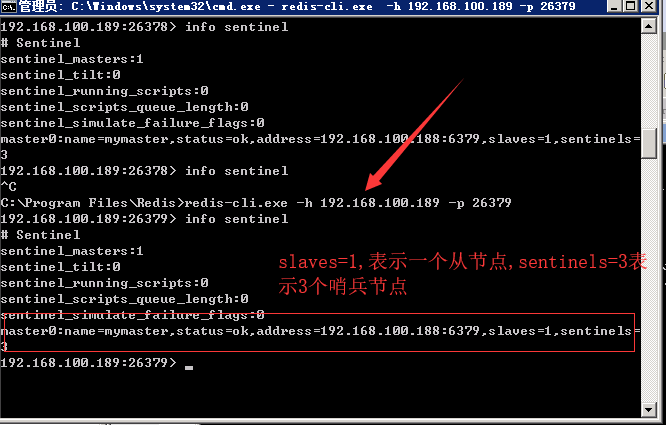
在去192.168.0.235机器上查看

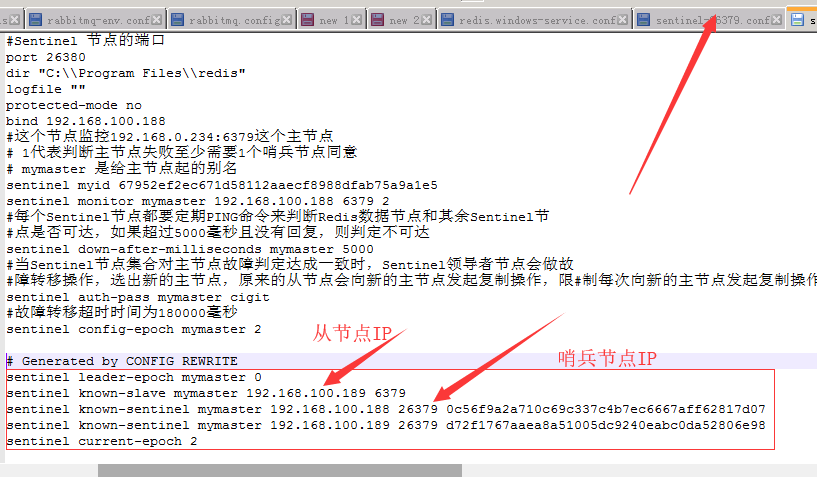
如图：



当前节点为主节点。

检查哨兵是否启动成功





# 文件服务器fastdfs安装

拷贝安装文件fastdfs-5.11.tar.gz和libfastcommon-1.0.36.tar.gz到Ubuntu服务器。

* 1. 安装libfastcommon文件

解压文件：

tar -zxvf libfastcommon-1.0.7



编译



安装文件



* 1. 安装fastdfs文件

解压文件

tar -zxvf fastdfs-5.11.tar.gz



编译文件



安装文件



* 1. 配置tracker服务

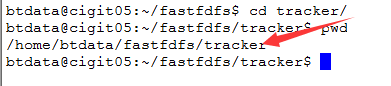
1. 建立tracker目录

例如：在/home/btdata目录下建立一个fastdfs文件夹





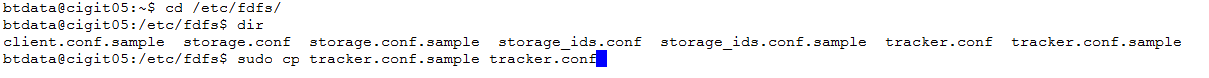
利用pwd命令打印当前的目录，并记录下当前的路径：/home/btdata/fastfdfs/tracker



1. 修改tracker配置文件

进入/etc/fdfs目录，通过dir可以看到tracker.conf.sample文件，调用

sudo cp tracker.conf.sample tracker.conf 拷贝一份。



使用sudo vim tracker.conf命令修改相应的配置文件



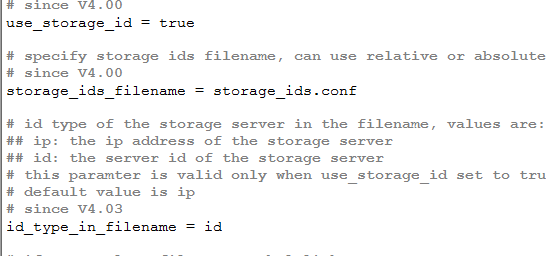
修改值base\_path为刚才创建的tracker目录：



修改use\_storage\_id=true,

修改storage\_ids\_filename为storage\_ids.conf

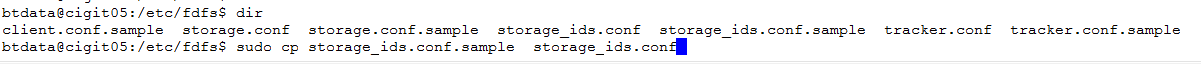
修改id\_type\_in\_filename=id



1. 创建storage\_ids.conf文件

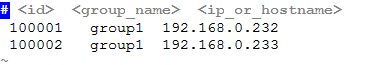
进入cd /etc/fdfs

sudo cp storage\_ids.conf.sample storage\_ids.conf



1. 修改storage\_ids.conf文件





group\_name都使用group1，表示都是一组的数据，会自动同步

ip\_or\_hostname写文件服务器的地址，如现在文件服务器有232和233两台服务器。

1. 建立对应的软连接

使用ln -s 建立软链接：

ln -s /usr/bin/fdfs\_trackerd /usr/local/bin

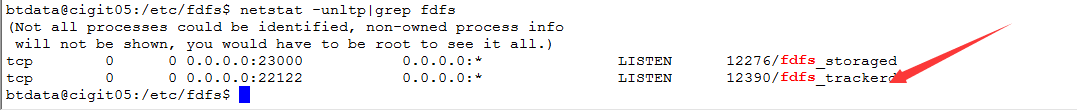
ln -s /usr/bin/stop.sh /usr/local/bin

ln -s /usr/bin/restart.sh /usr/local/bin

1. 启动服务：service fdfs\_trackerd start

查看监听：netstat -unltp|grep fdfs

看到22122端口正常被监听后，这时候就算 Tracker服务安装成功！

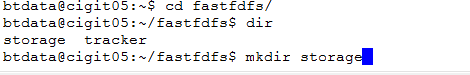


为了搭建集群，同样在另外一台服务器上面，配置同样的tracker服务。

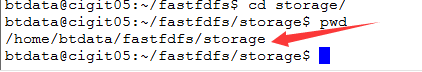
* 1. 搭建storage服务器

1. 建立storage目录

例如：在/home/btdata目录下建立一个fastdfs文件夹



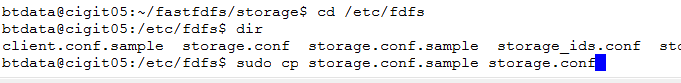
利用pwd命令打印当前的目录，并记录下当前的路径：/home/btdata/fastfdfs/storage



1. 修改storage配置文件

进入/etc/fdfs目录，通过dir可以看到storage.conf.sample文件，调用

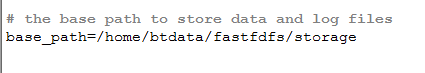
sudo cp storage.conf.sample storage.conf 拷贝一份。



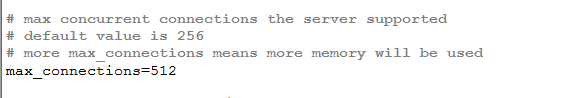
使用sudo vim storage.conf命令修改相应的配置文件



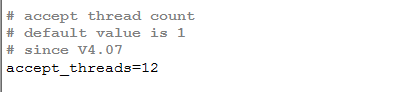
修改base\_path=/home/btdata/fastfdfs/storage



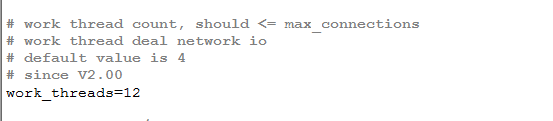
修改max\_connections=512



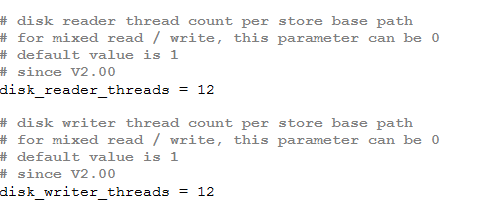
修改accept\_threads=12



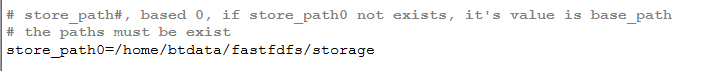
修改work\_threads=12



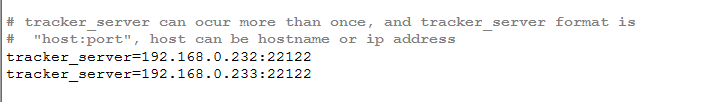
修改disk\_reader\_threads=12,dist\_writer\_threads=12



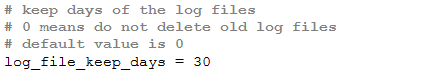
修改store\_path0为/home/btdata/fastfdfs/storage



修改tracker\_server的地址为配置的文件服务器的IP地址，因是集群所以有两个



修改log\_file\_keep\_days=30



退出保存配置文件

同样把另外的文件服务器修改为同样的配置

1. 建立软连接

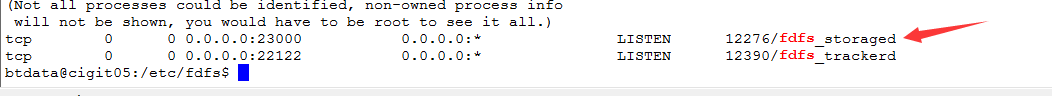
ln -s /usr/bin/fdfs\_storaged /usr/**local**/bin

1. 启动服务

service fdfs\_storaged start

1. 查看服务器是否启动成功

netstat -unltp|grep fdfs



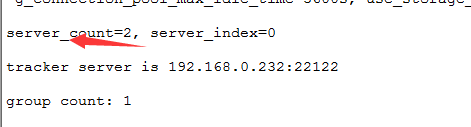
查看文件服务器的整个的状态：

在任一个文件服务器里面执行如下命令：

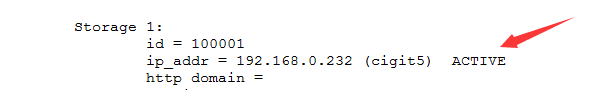
fdfs\_monitor /etc/fdfs/storage.conf



查看下图的server\_count=2，说明有storage



可以看到storage1和storage2的状态都是active表明，文件服务器启动成功。





# RabbitMQ安装以及集群配置

服务器（windows-server-2008-R2-企业版）：

192.168.0.234 用户：Administrator

192.168.0.235 用户：Administrator

所需依赖：

er-lang21.0.1环境

RabbitMQ3.7.7

安装erlang：

1. 下载erlang
2. 双击安装

安装rabbitmq:

1. 下载rabbitmq
2. 双击安装

配置步骤：

注意：安装rabbitmq集群的主机名不能相同，因为rabbitmq是用主机名来命名节点，如果相同则集群会误认为这个节点是自己，则不能构成集群。

1. 按照正常步骤在机器上安装er-lang环境以及rabbitmq程序。

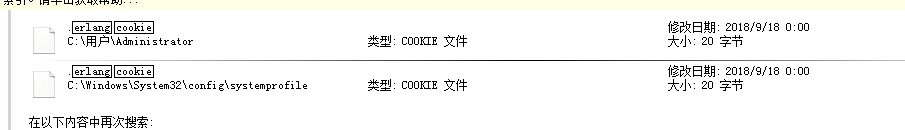
安装完成后停掉所有机器上的rabbitmq服务。

1. 在其中一台机器上(这里以192.168.0.234为例)的C:\用户\Administrator 目录下 拷贝.erlang.cookie文件

因为rabbitmq集群是以erlang为基础来通信的，所以要保证集群中所有机器上的.erlang.cookie 文件的内容一致。

将192.168.0.234 机器上的.erlang.cookie文件拷贝至 192.168.0.235机器的

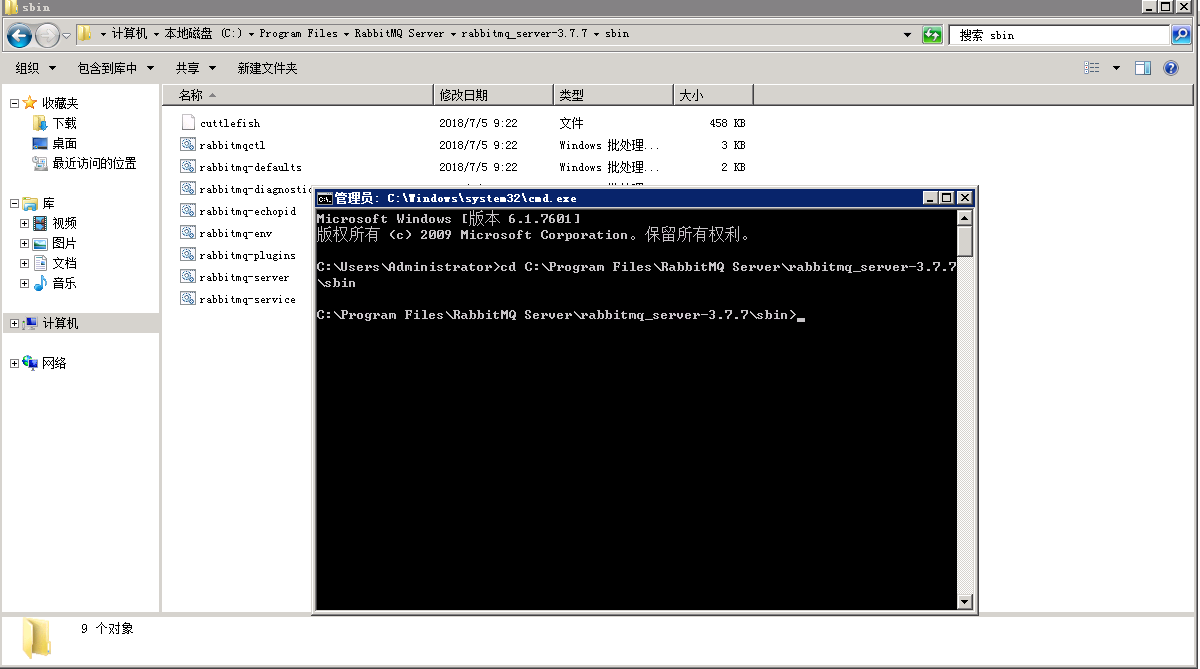
C：\用户\Administrator 目录 和C：\Windows 以及 C：\Windows\System32\config\systemprofile 这三个目录中



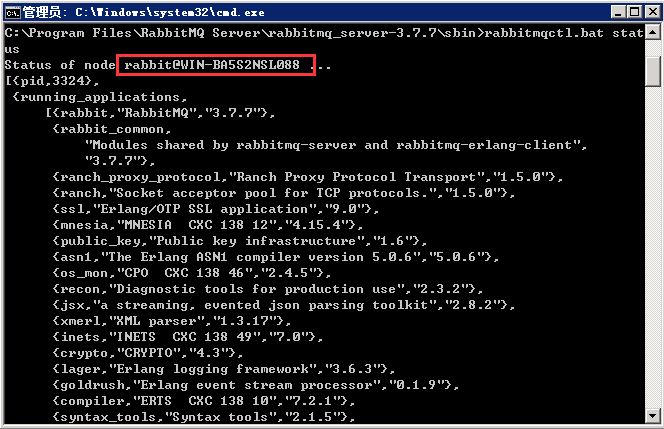
至此，er-lang环境准备完毕，一定要关闭rabbitmq后再进行拷贝，拷贝完成后启动rabbitmq服务

1. 在192.168.0.234机器上的rabbitmq安装目录中的sbin目录下启动命令行

或者在命令行中进入该目录



输入 rabbitmqctl.bat status 查看当前节点情况，如下图，记下红线圈住的节点名称

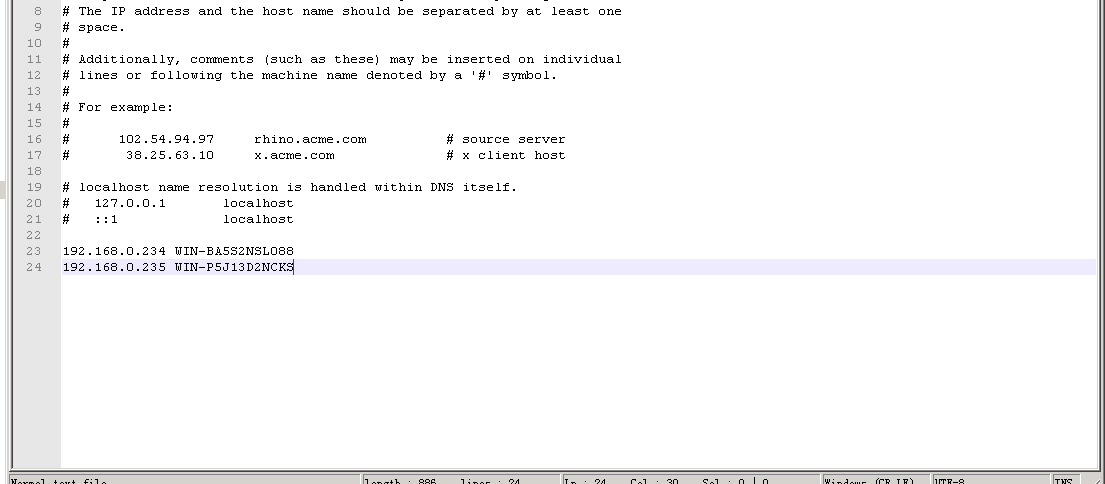


在192.168.0.235进行同样的操作，记住节点名称

1. 修改hosts文件

修改192.168.0.234机器上的C:\Windows\System32\drivers\etc 目录下的hosts文件

内容如下：

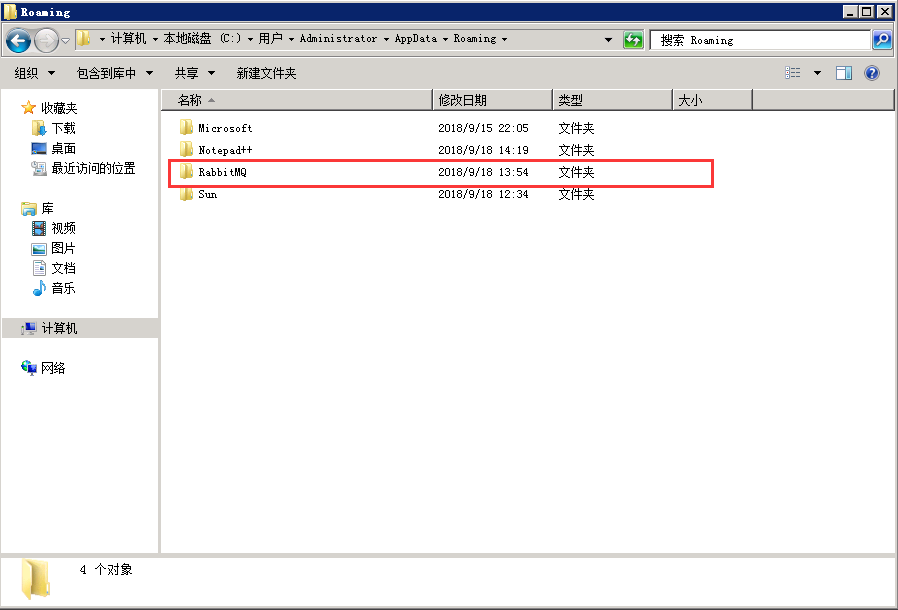


对192.168.0.235机器的hosts文件修改也是如此

1. 修改rabbit配置文件

192.168.0234 ：

在 C:\Users\Administrator\AppData\Roaming 中（需要设置隐藏文件夹可见）的rabbitMQ文件夹中



新建rabbitmq-env.conf文件

内容如下，根据实际情况修改

NODENAME=WIN-BA5S2NSL088

NODE\_IP\_ADDRESS=192.168.0.234

NODE\_PORT=5672

RABBITMQ\_MNESIA\_BASE=C:\Users\Administrator\AppData\Roaming\RabbitMQ\db

RABBITMQ\_LOG\_BASE=C:\Users\Administrator\AppData\Roaming\RabbitMQ\log

NODENAME为查看到的节点名称，ip地址改为自己机器的地址，两个目录根据自己实际情况修改

再新建rabbitmq.config文件，内容如下，根据实际情况做修改：

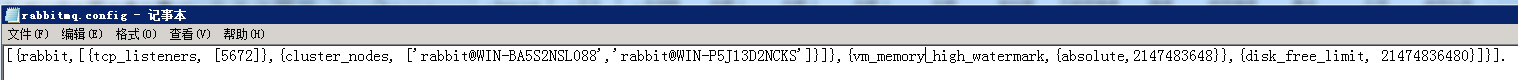
[

{rabbit,[{tcp\_listeners,[5672]},{cluster\_nodes,['rabbit@WIN-188','rabbit@WIN-189']}

,{vm\_memory\_high\_watermark,{absolute,2147483648}},{disk\_free\_limit,21474836480}

]}].

如图：



192.168.0.235 配置步骤同上

1. 加入集群

在192.168.0.234机器上的rabbitmq安装目录中的sbin目录下启动命令行

或者在命令行中进入该目录

关闭应用

rabbitmqctl.bat stop\_app

然后重置

rabbitmqctl.bat reset

然后启动

rabbitmqctl.bat start\_app

然后在192.168.0.235机器上

rabbitmqctl.bat stop\_app

rabbitmqctl.bat reset

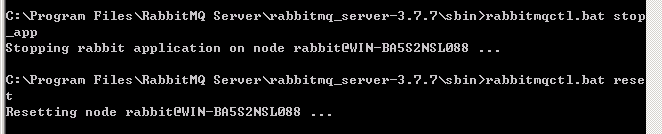
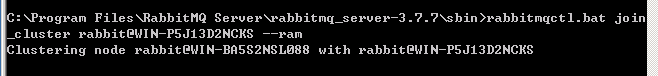
rabbitmqctl.bat join\_cluster rabbit@节点名称 --ram

--ram 表示这个节点作为内存节点而加入的，不加这个则为默认的磁盘节点

然后启动

rabbitmqctl.bat start\_app

过程如图：

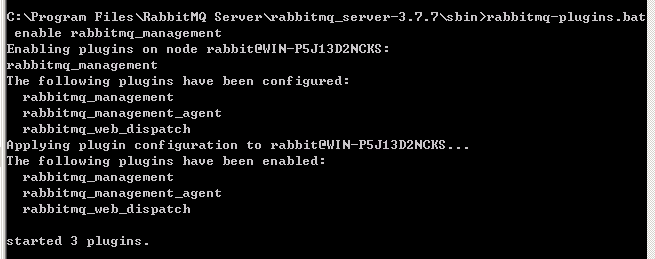
 

1. 配置集群

首先为两台机器开启web端管理界面

在命令行中 输入 rabbitmq-plugins.bat enable rabbitmq\_management

如图：



然后添加一个root用户

命令 rabbitmqctl.bat add\_user 用户名 密码



赋予root用户管理员角色

命令 rabbitmqctl.bat set\_user\_tags 用户名 权限



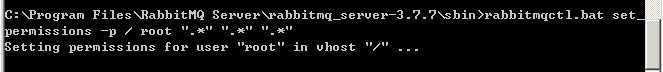
检查当前用户管理员角色是否配置成功，下图则表示成功



然后给root用户赋予 virtul host 的读写权限

rabbitmqctl.bat set\_permissions -p virtulhostName 用户名 ".\*" ".\*" ".\*"

如图：



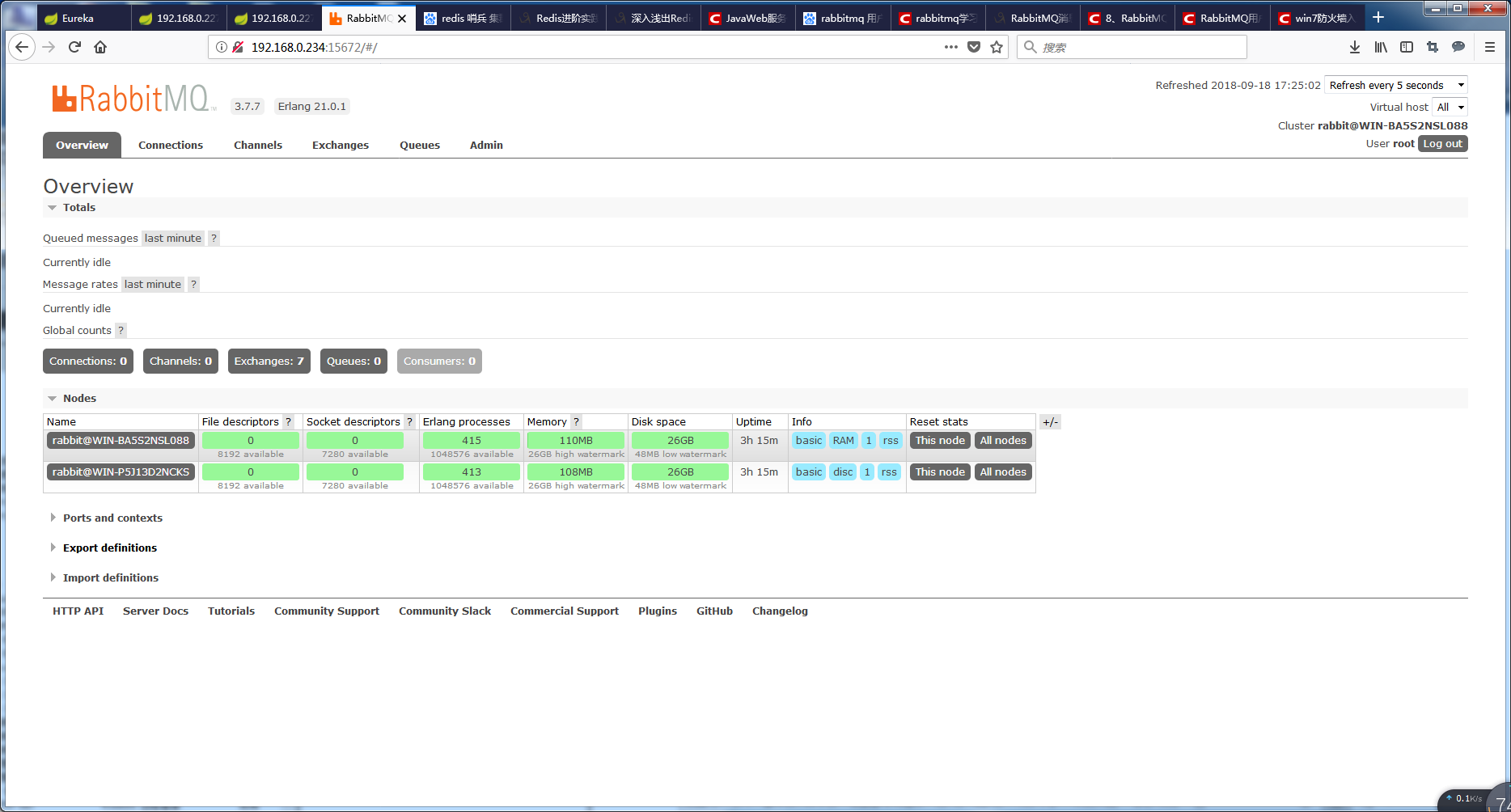
1. 设置HA

在浏览器中 输入 IP地址：15672

这里为 192.168.0.234:15672

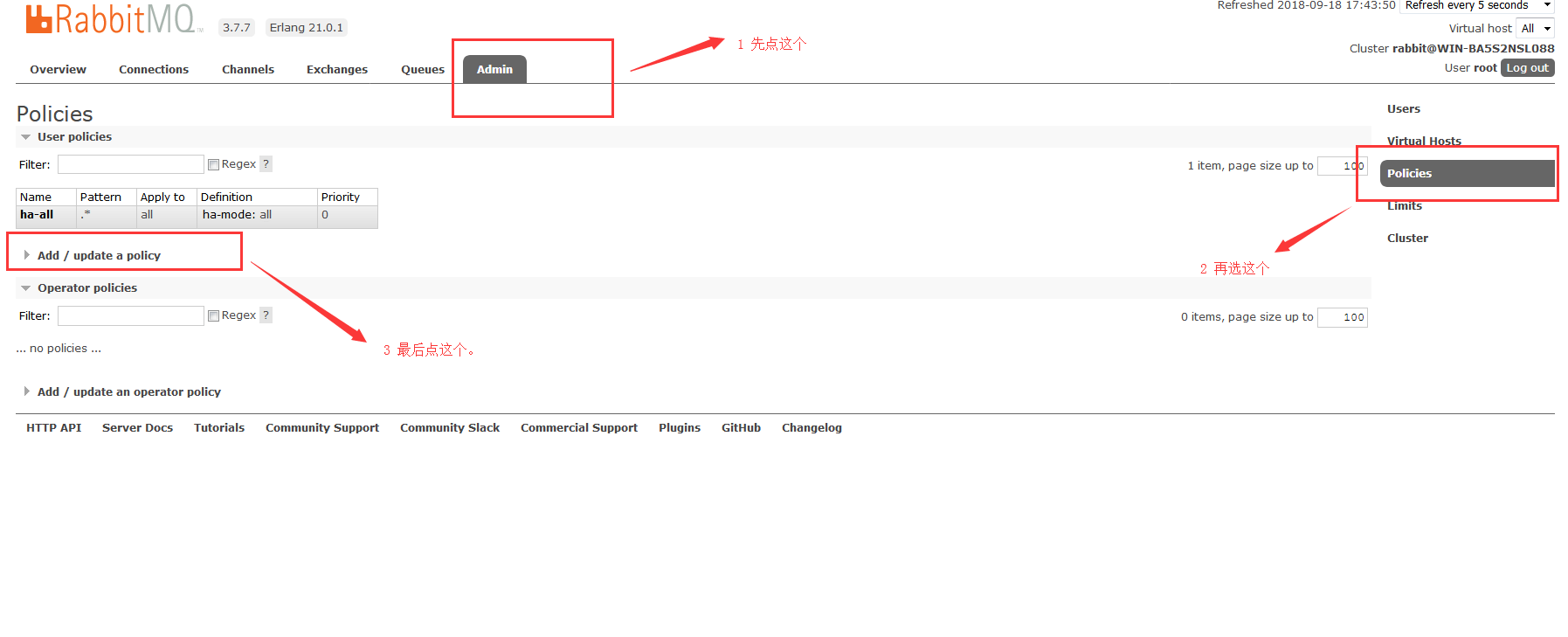


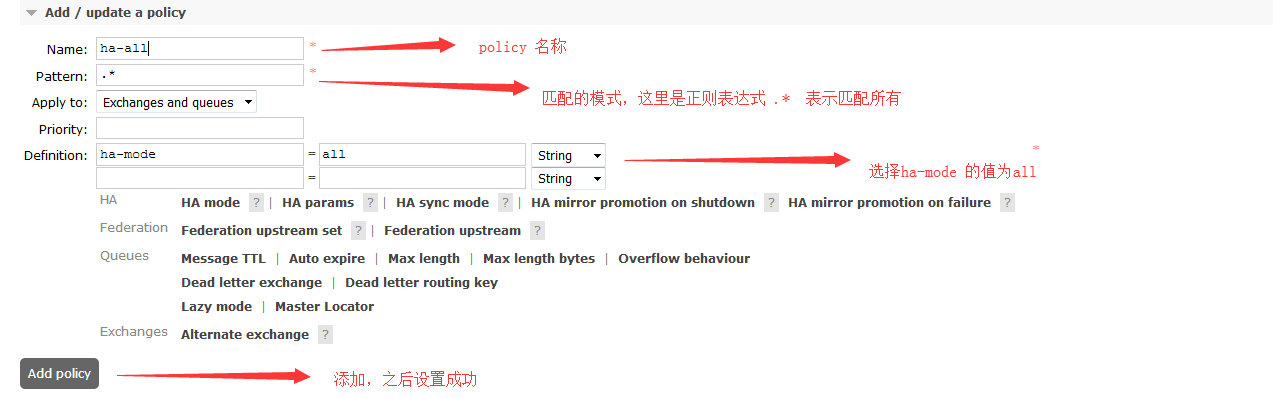
使用 root用户登录，登录成功后



可以看到集群的情况

然后配置高可用，按图步骤





集群中的所有机器都需要配置一样的用户名和密码以及权限，按照步骤7来配置。

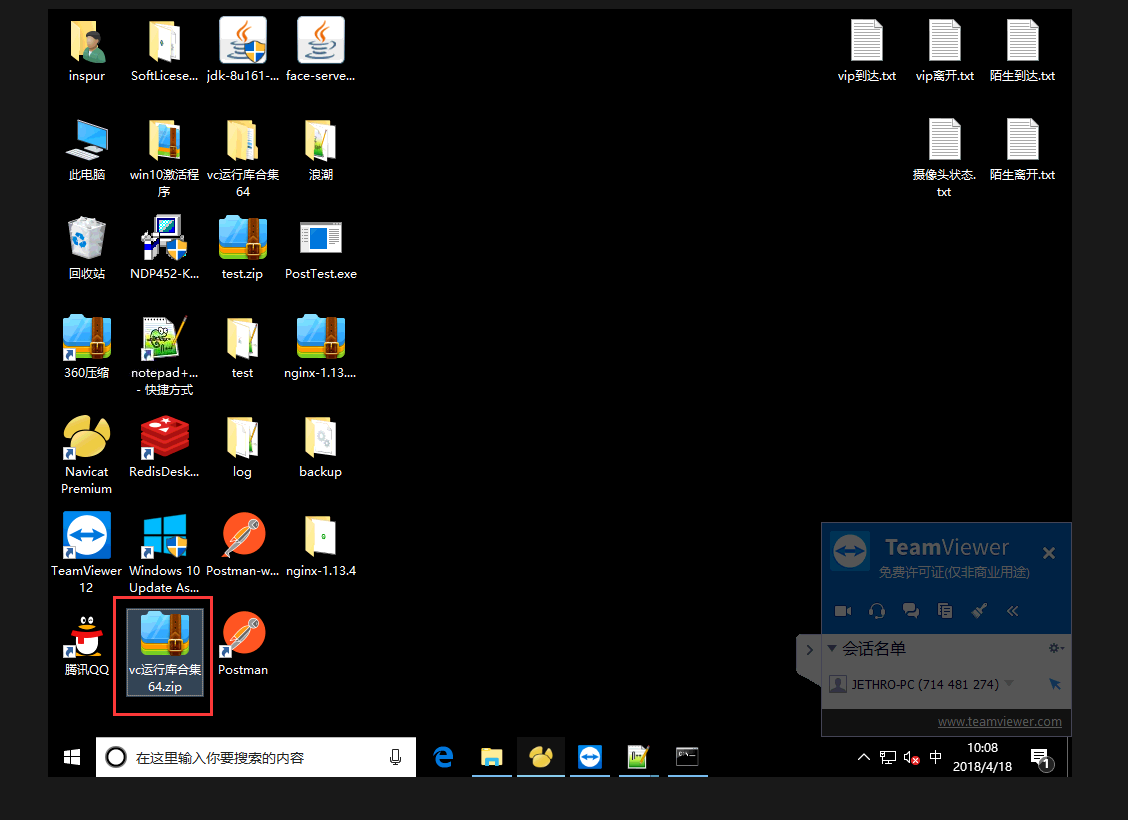
配置过程中若步骤完全正确，但是集群配置不成功，检查防火墙的出入站规则，是否允许rabbitmq的这些端口被外部访问

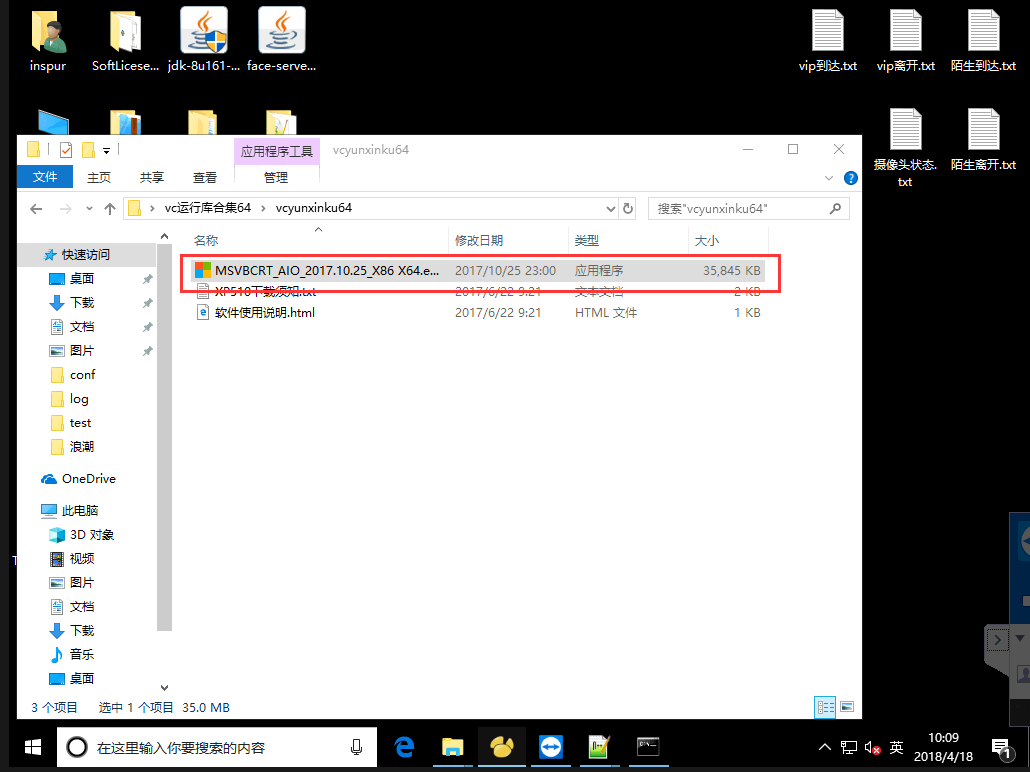
# Ukey注册

利用uKey注册软件，获取当前的注册码，向相关负责人申请，相应的Ukey的识别码，进行注册。

# VC++运行环境安装

下载VC64位运行库合集，运行解压后的exe文件直接安装即可。





# 全流程安检系统

全流程安检系统一共由8个服务模块组成，分别是：数据服务，1：1人证对比服务，预安检，安检，复核，配置中心，注册中心， 以及网关。

如图：



其中最主要的服务为以下几个服务：

数据存储服务：对接数据平台的接口，以及为预安检，安检，复核等服务提供数据写入服务。除复核特征的写入操作外，所有的数据写入操作都是在这个服务中完成的

1:1人证比对：为预安检，安检提供1:1人证对比服务。

预安检：执行预安检流程，调用1:1人证对比服务进行人证比对，并返回结果，也会把结果数据发送至数据平台，使其进行写入操作

安检：执行安检流程，调用1:1人证对比服务进行人证比对，并返回结果，也会把结果数据发送至数据平台，使其进行写入操作。同时将比对成功的人脸特征写入人脸复核库中

复核：执行复核流程，并返回相应结果

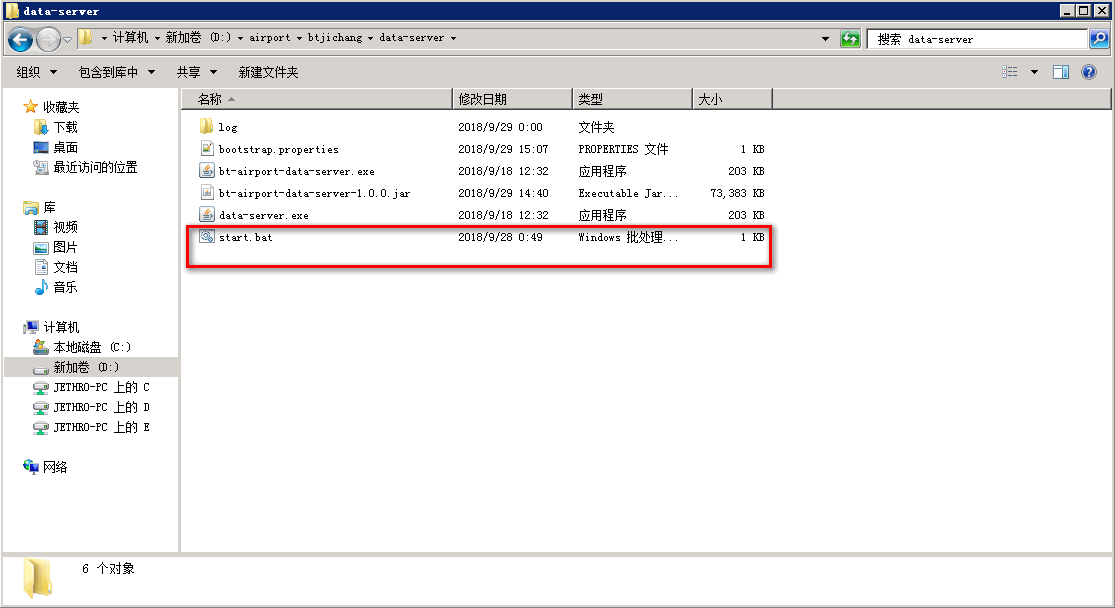
数据平台服务：用于提供数据平台的查询接口，包含预安检，安检，复核，旅客分析，通道分析等。

网关：提供白名单，请求正确性校验服务。

在部署完成后，可以使用批量启动脚本启动所有服务

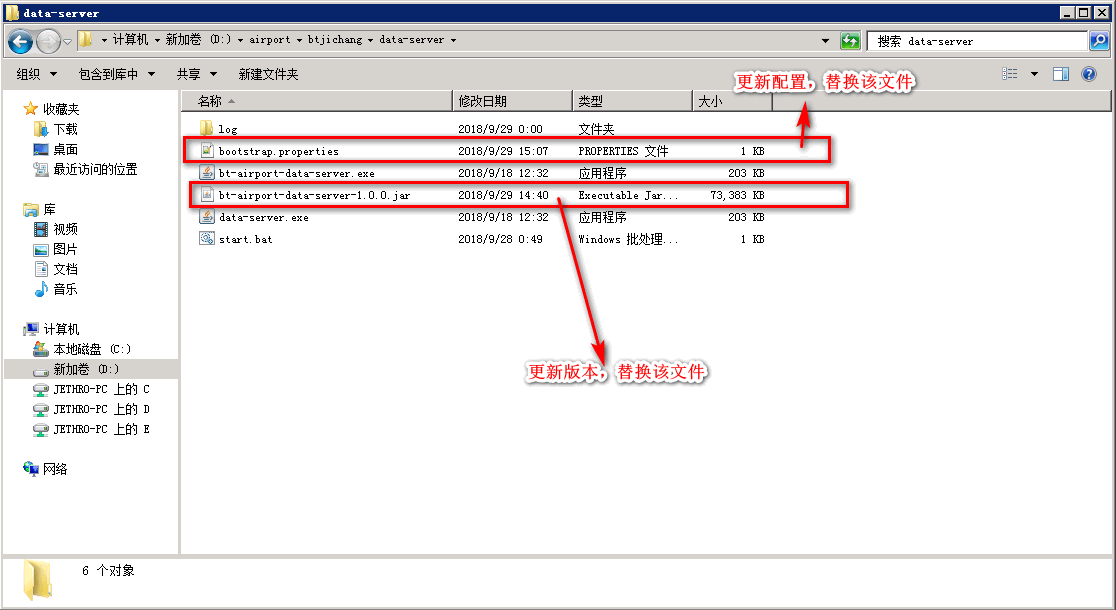
如需要单独重启某个服务，需要进入那个服务的文件夹中运行start.bat 即可

以data-server为例，如图，双击start.bat即可重启：



# 全流程安检系统版本更新

更新某个服务的系统版本，需要直接复制替换编译生成之后的jar包到对应服务的文件夹中。如果需要更新配置文件，则需要替换掉bootstrap.properties。以data-server为例，如图：

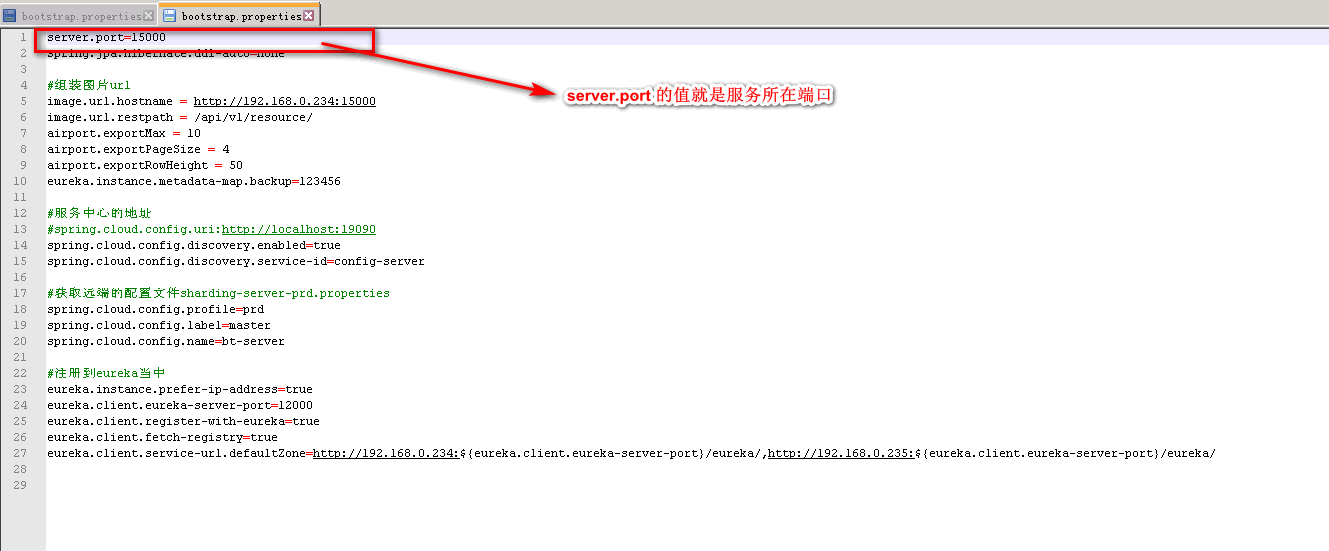


# 防火墙端口配置

如果一切服务正常配置但是还是无法连接上，那么我们需要配置防火墙的入站规则

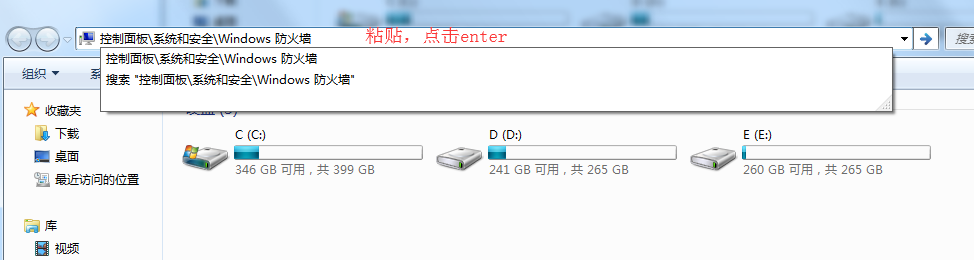
以data-server 为例：

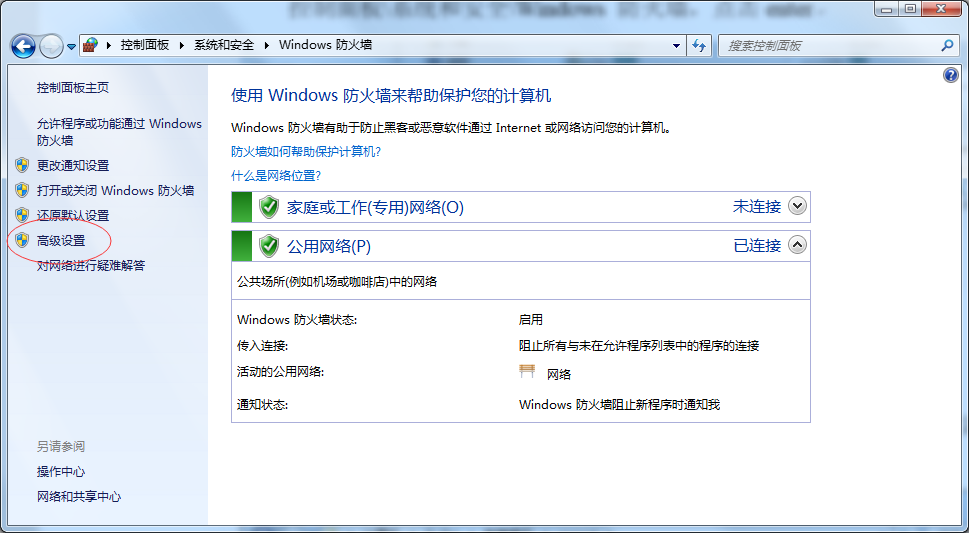
先查询服务所在端口，打开bootstrap.properties文件，如图



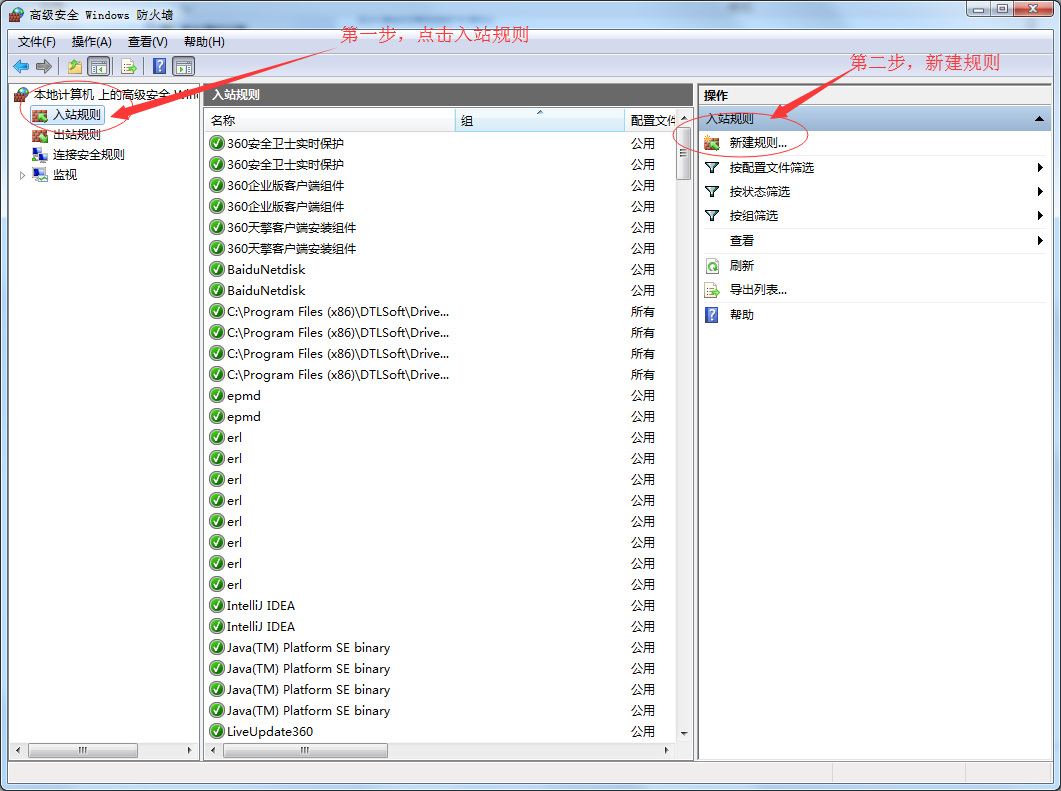
然后打开电脑的控制面板🡪系统和安全🡪Windows防火墙，或者直接复制

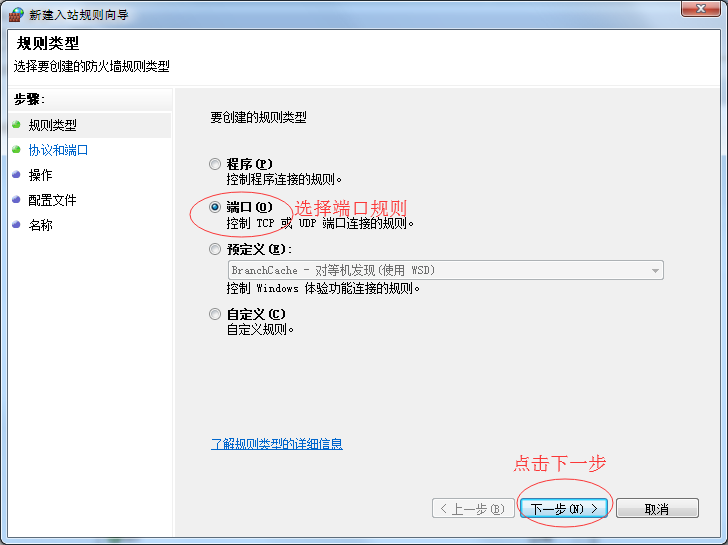
控制面板\系统和安全\Windows 防火墙，点击enter

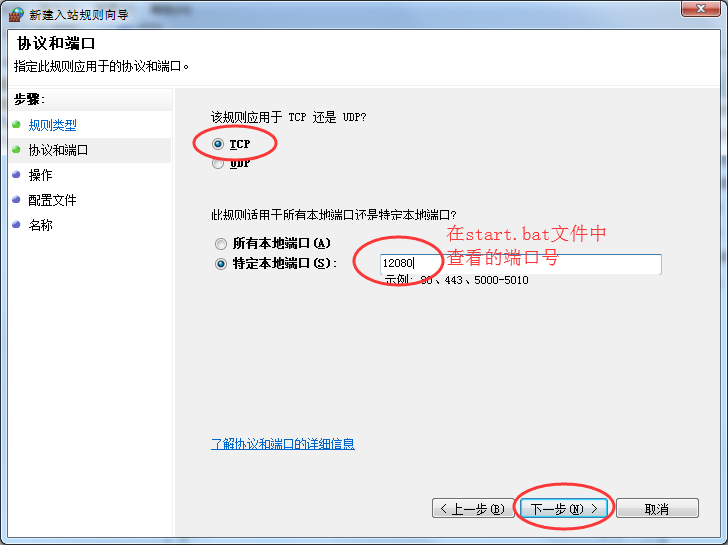


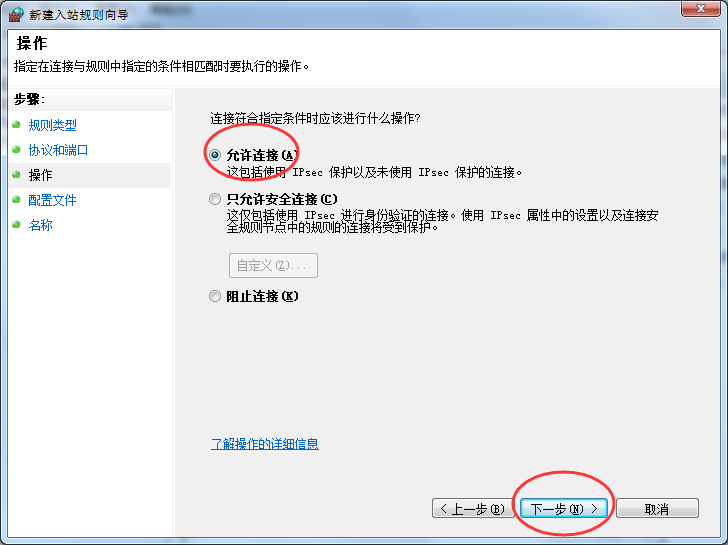


点击高级设置，首先设置入站规则：

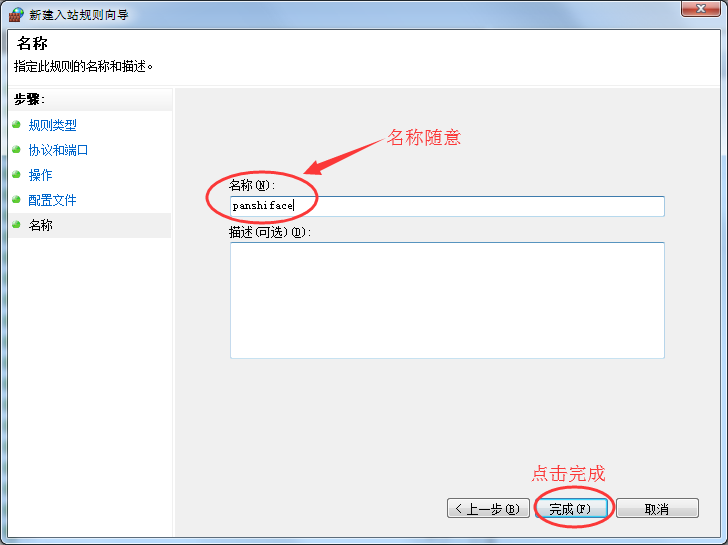




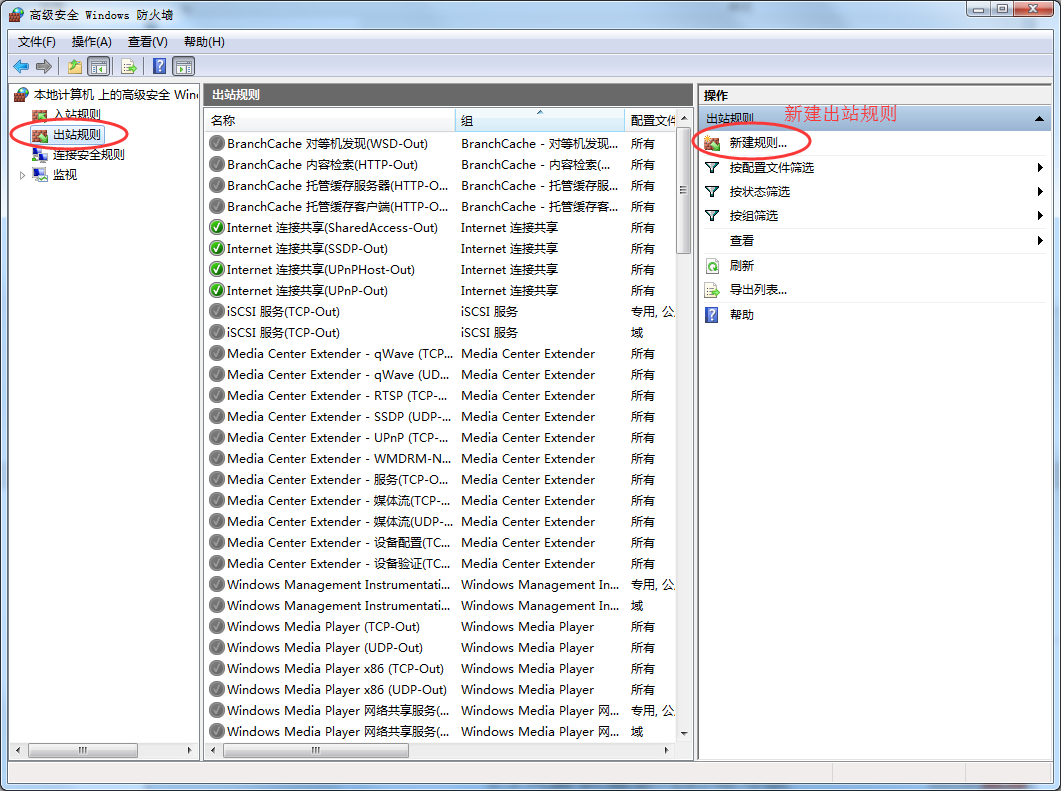




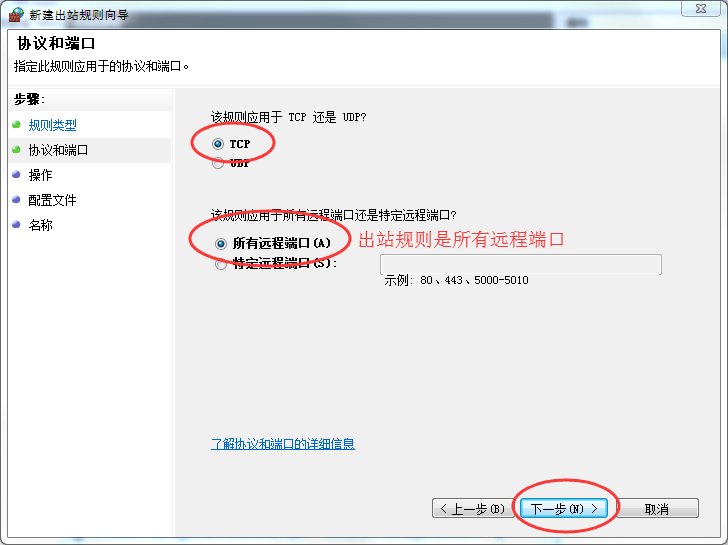


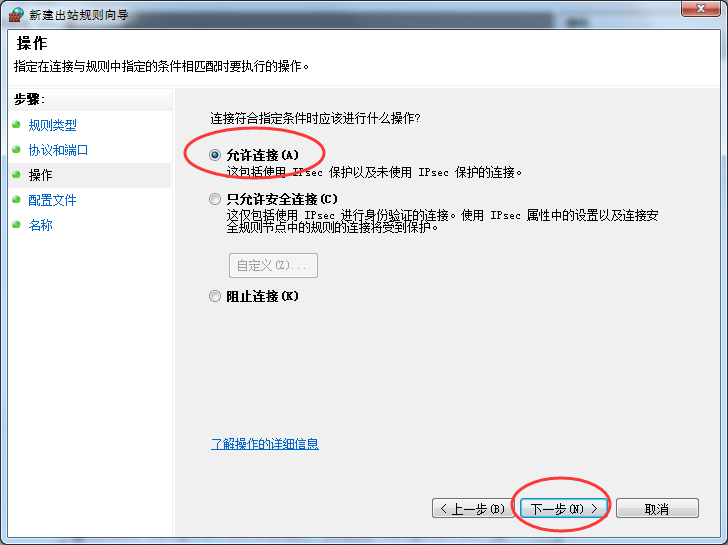


设置出站规则：











# 验证图片是否存储在文件服务器中

我们使用了2台机器用来做分布式文件服务器，所以文件存储的位置是在这两台机器中的其中一台。

两台机器的ip

192.168.0.232

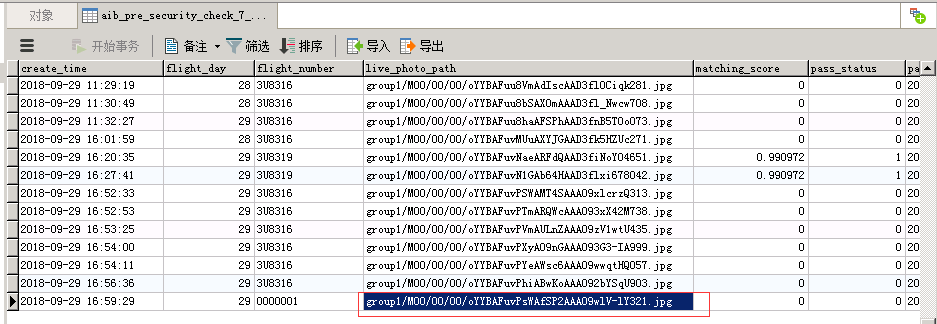
192.168.0.233

完成预安检流程后，系统会将图片上传至文件服务器，先去验证数据库中是否有该条记录，

表aib\_pre\_passenger\_check 是预安检旅客流水表

表中 certificate\_photo\_path 字段和live\_photo\_path字段分别表示证件照存放的路径以及现场照存放的路径

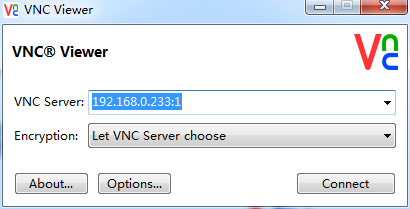
如图：



以红圈中的路径为例

使用VNC软件分别登录到两台服务器上

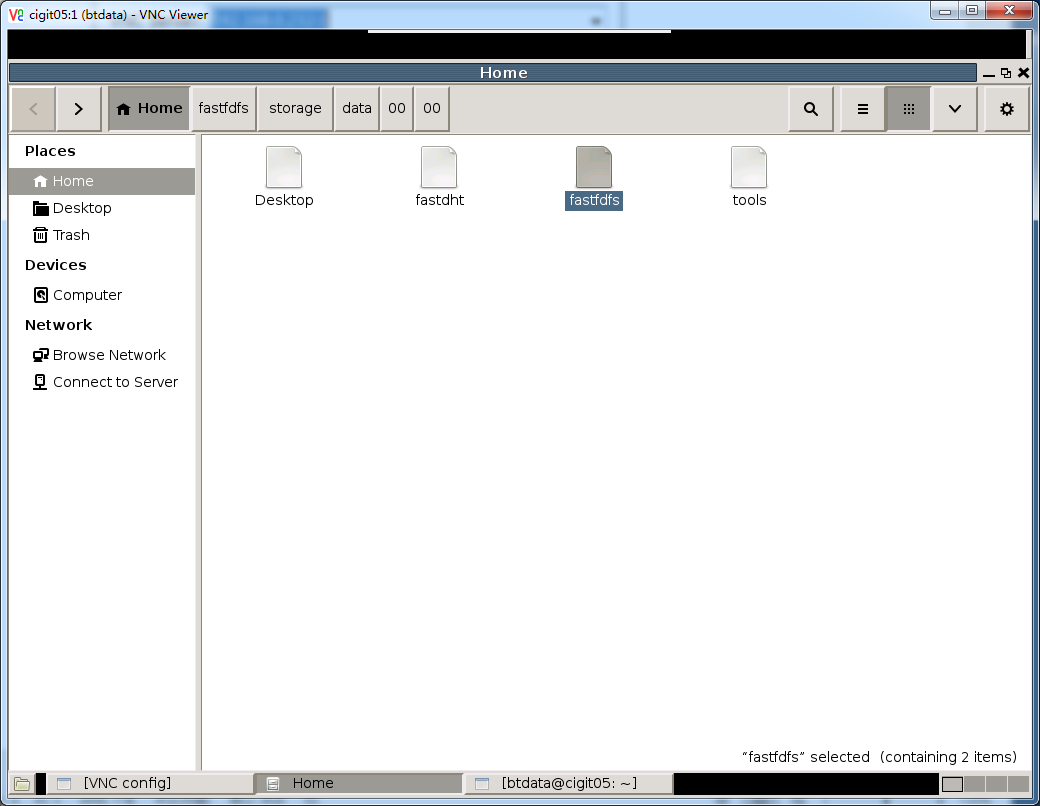
先查看192.18.0.233



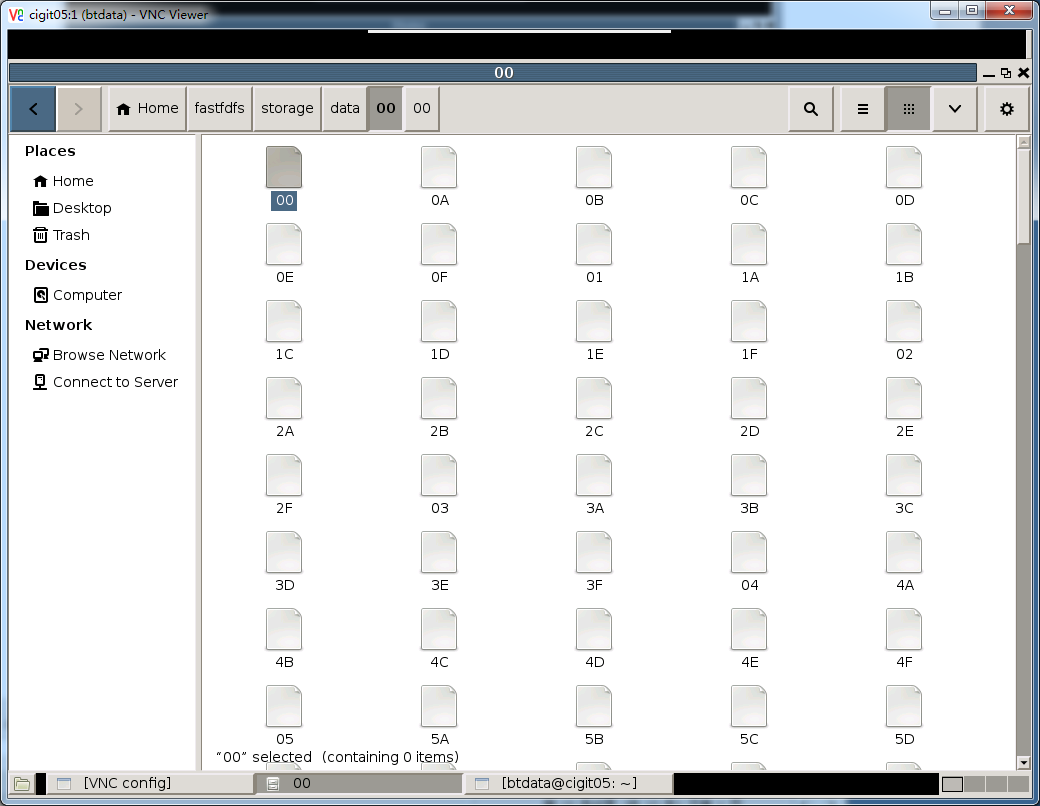
输入密码123456



之后出现

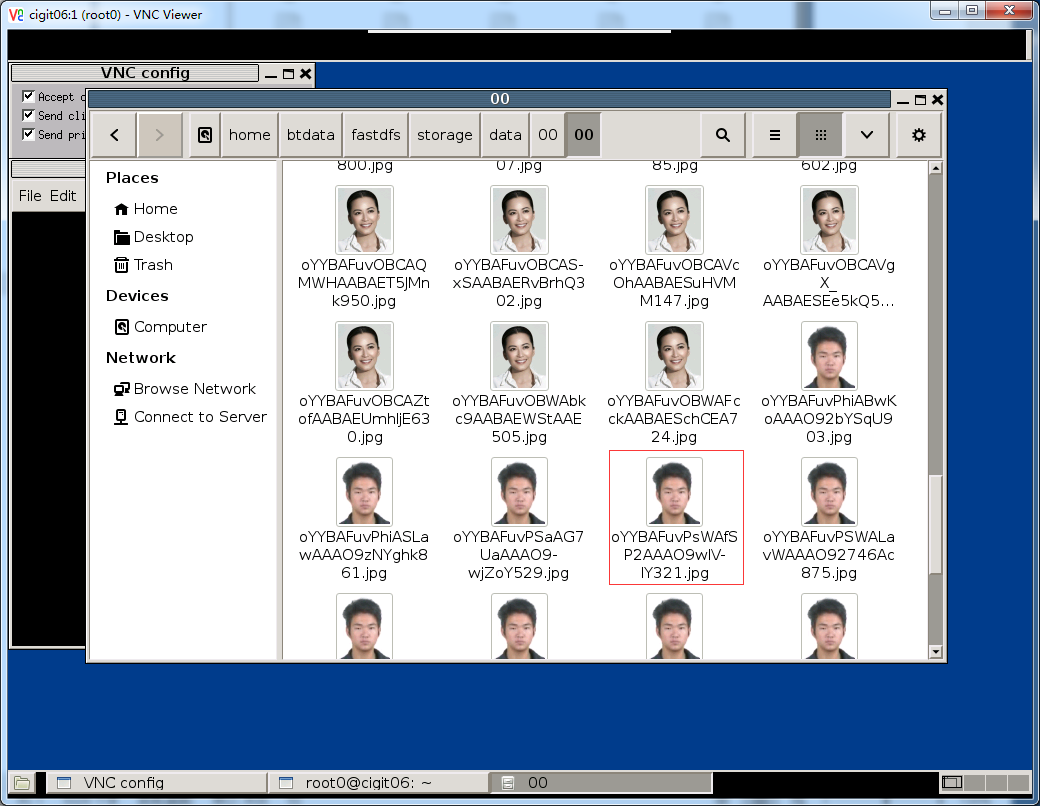


进入fastfdfs/storage/data目录，如图



然后按照在数据库中看到的路径依次进入到 /00/00目录下

如图



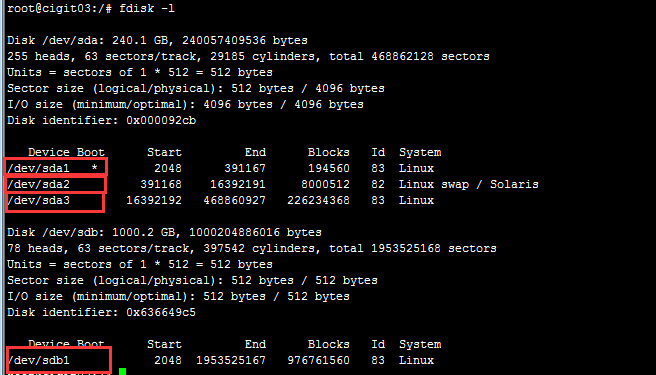
可以看到文件成功被上传到服务器中

如果在这台服务器中没有找到文件，则应切换至另外一台文件服务器上查看。

# 硬盘挂载

1. 查看硬盘分区

sudo fdisk -l



可以看到一共有这几个分区。

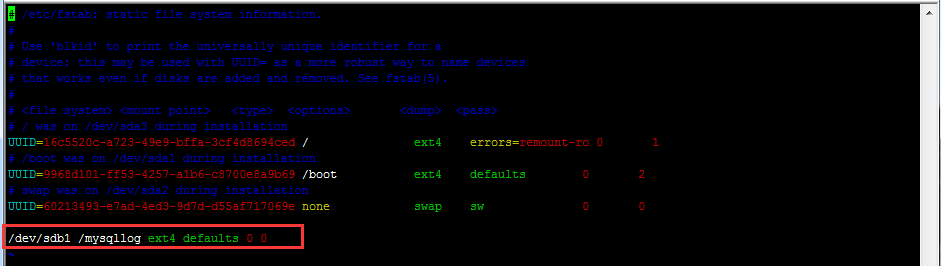
挂载分区

现在假设 /dev/sdb1 这个分区是我们想要挂载的分区。而/mysqllog是我们要映射的路径

1. 修改配置文件 /etc/fstab

sudo vim /etc/fstab

如图



在文档末尾这样添加即可

第一个参数是硬盘分区

第二个参数是系统目录

第三个参数是硬盘分区的文件系统类型

后面的参数不变

保存并退出

使用 mount -a 使刚刚的配置生效

这种配置方式是永久生效的。

还可以使用mount 命令来进行挂载，这种方式重启之后就会失效。

mount -t fstype fspath syspath

fstype: 硬盘分区的文件系统类型

fspath: 硬盘分区

syspath: 系统目录

# redis和redis-sentinel注册成windows服务

redis自己提供了相应的命令来管理服务

删除服务 ，默认的redis

redis-server --service-uninstall

删除指定servername服务

redis-server --service-uninstall –service-name servername

注册redis为windows服务

redis-server --server-install redis.windows.service.conf --loglevel verbose

注册其他名称的服务并指定端口

redis-server --service-install --service-name servername --port 10001

将Sentinel安装为Windows服务

redis-server --service-install sentinel.conf --sentinel --service-name RedisSentinel --port 26379

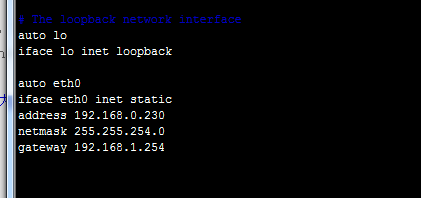
# Ubuntu 网络配置

修改/etc/network/interfaces 文件

sudo vim /etc/network/interfaces

静态分配ip

内容如下



将address和gateway改为自己网络的配置。

  使用命令 /etc/init.d/networking restart   将网卡重启，使配置生效

# 数据平台服务器数据写入

数据平台没有提供服务器信息的写入操作，所以服务器信息需要我们手动写入





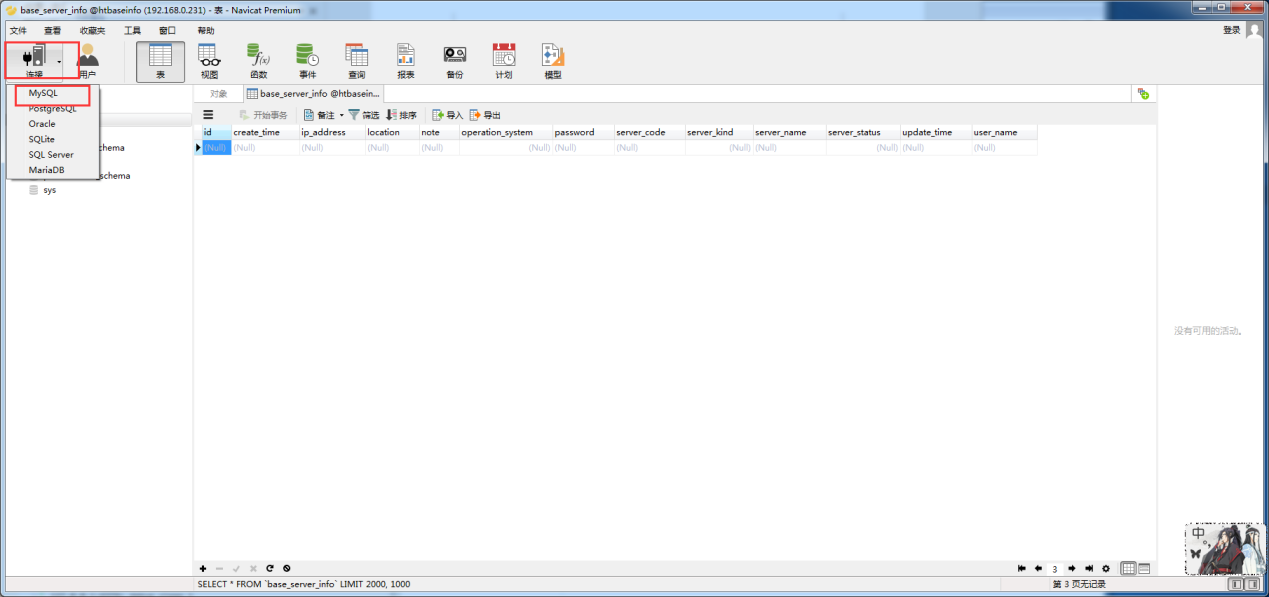
根据服务器信息表以及字典数据定义，我们可以知道要写入的值。

以一台windows系统，ip地址为192.168.0.234，用户名为Administrator，密码为11111111的人脸识别服务器为例。

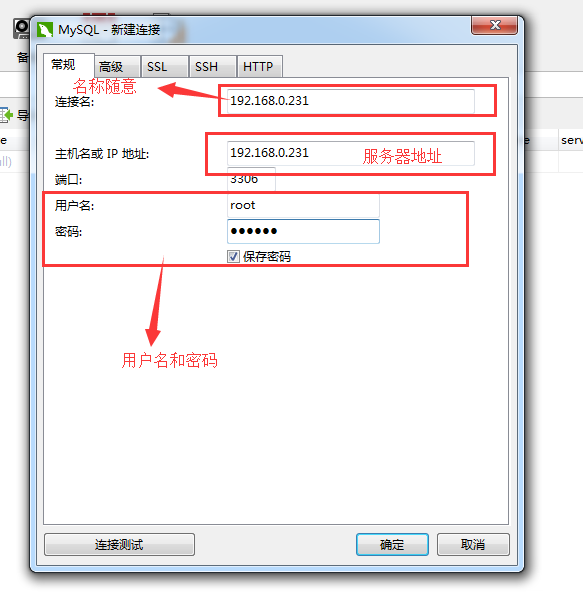
1. 首先下载Navicat Premium 安装包。双击打开安装包，全部按照默认设置进行安装。
2. 连接数据库服务器

假设数据库服务器的地址是192.168.0.231 数据库用户名是root 密码为123456

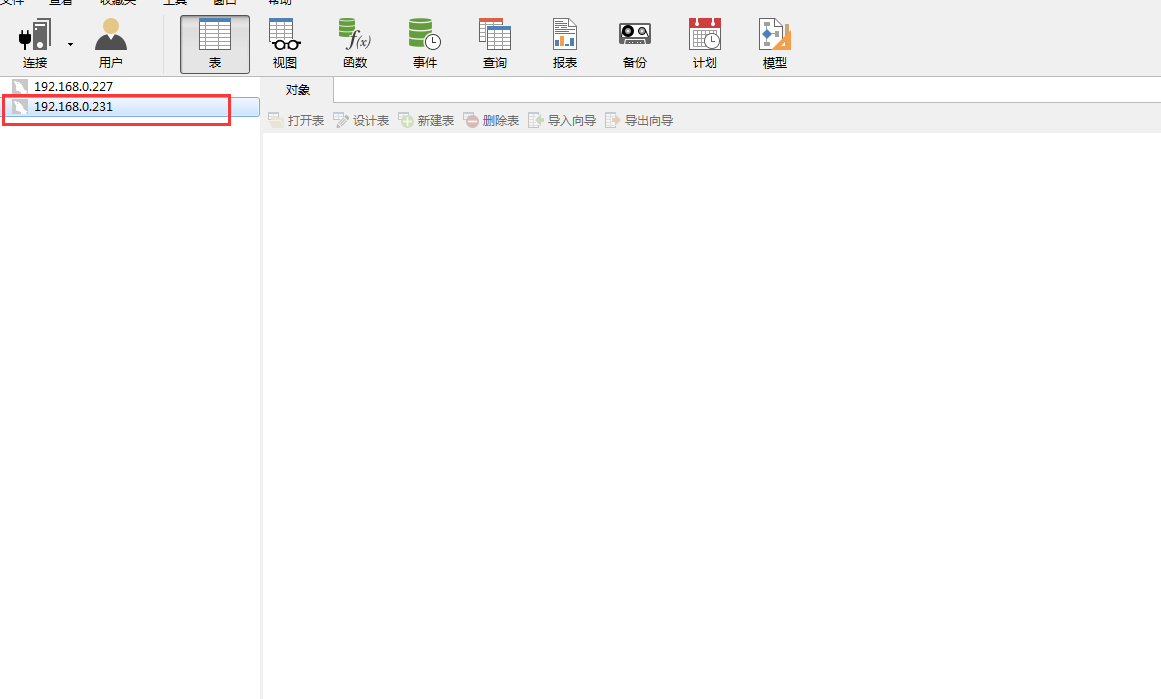
打开Navicat Premium，点击连接，并选择mysql



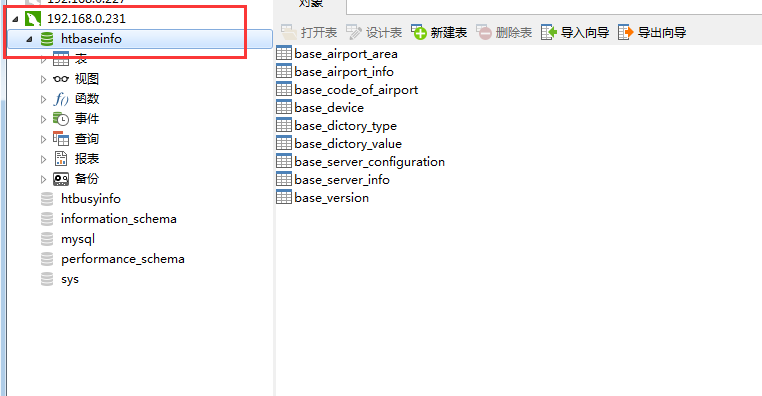
输入服务器相关信息后，点击确定。



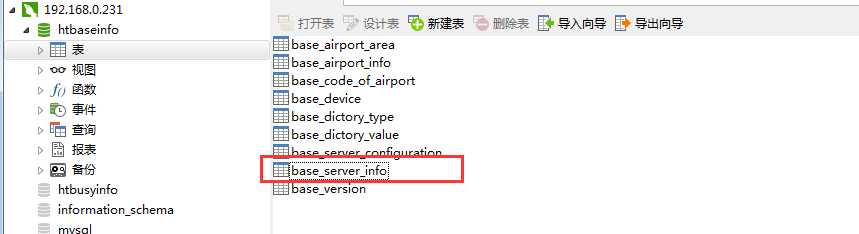
然后左侧会多出一个连接

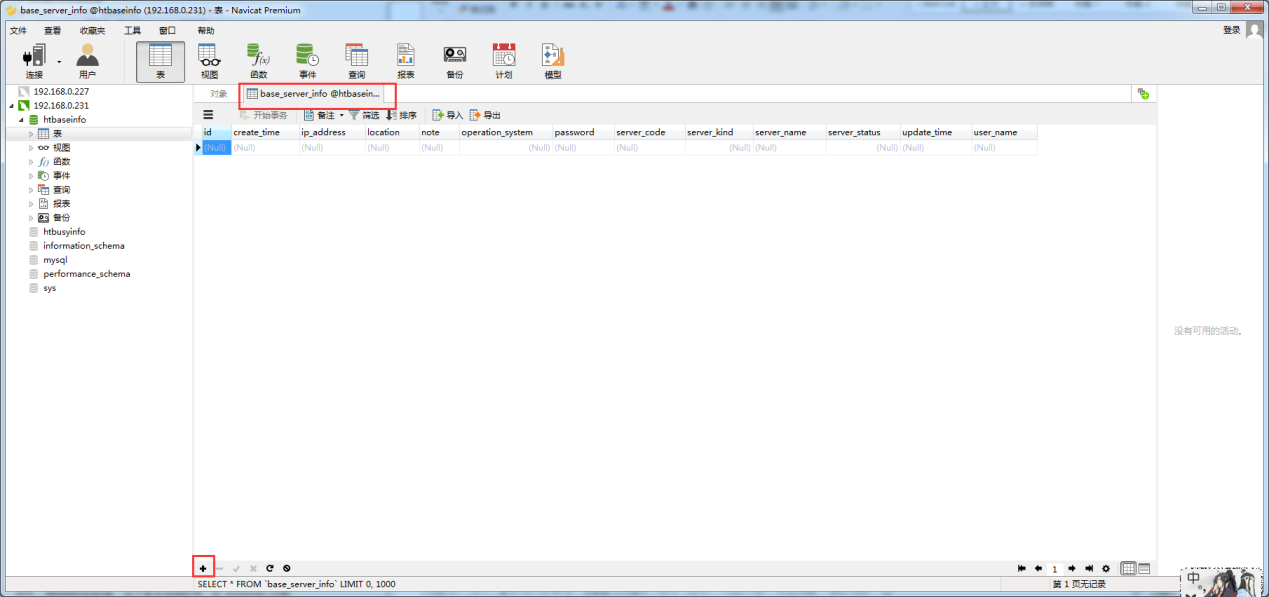


双击它，打开连接，并双击选择btbaseinfo数据库。

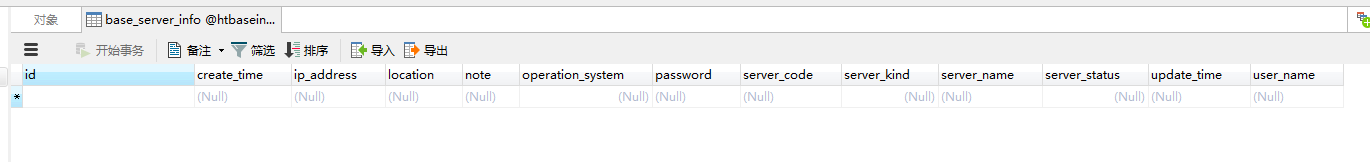


双击打开base\_server\_info表





点击下方的+号，添加数据



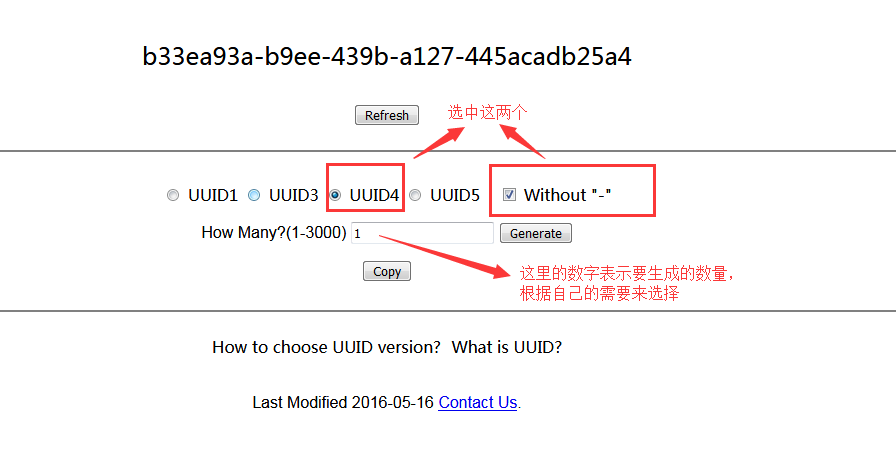
根据数据库设计，id需要使用uuid，用户名和密码需要使用base64加密。

uuid可以在<http://www.uuid.online/> 生成

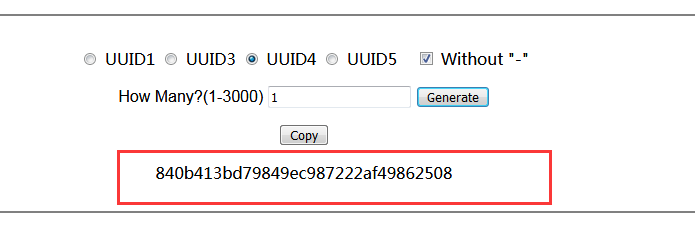
用户名和密码加密可以在<http://tool.chinaz.com/Tools/Base64.aspx> 进行加密

如图

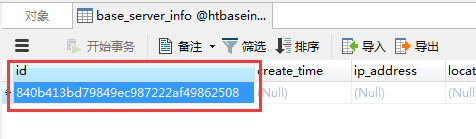
uuid



点击Generate，会在下方显示生成的uuid,如图



将此id复制下来，拷贝至id字段中。



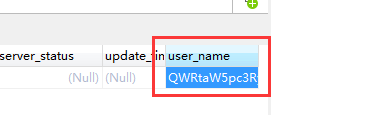
加密用户名，打开网站<http://tool.chinaz.com/Tools/Base64.aspx>



这里我们输入的是用户名Admintrator，点击进行加密。



将加密后的数据复制粘贴至user\_name字段



同理，密码也是同样的操作。

然后将我们已知的数据写入，

目前已知的数据有

操作系统：windows

服务器类型：人脸识别服务器

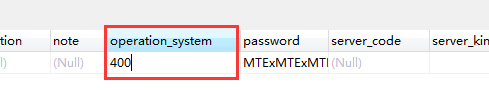
IP地址：192.168.0.234

操作系统和服务器类型对应的字段分别是operating\_system,server\_kind

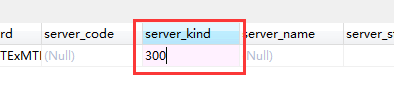
但是我们不能直接这样写入，而是应该对应上面的字典表的值来进行写入

如，windows系统对应的操作系统值是400

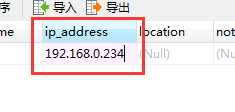
所以我们应该将400写入



同理服务器类型的值应该是300。所以



IP地址



现在还剩下location ，server\_code，server\_name，server\_status，note

note字段可以不填

location是安装地址，根据实际情况来填写，这里假设是重庆绿色智能技术研究院

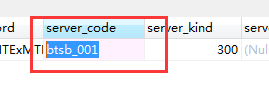
server\_code是服务器编号，以机场拼音首字母为开头，功能拼音的首字母为中间字符，下划线做分隔，后面跟数字编号，如btsb\_001 意为 白塔机场，功能为识别的第1台服务器

server\_name是服务器名称，根据实际情况命名，这里假设为 人脸识别服务器

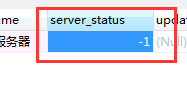
server\_status是服务器状态，有两个值，0表示在线 ，-1表示离线，默认填写-1

如图

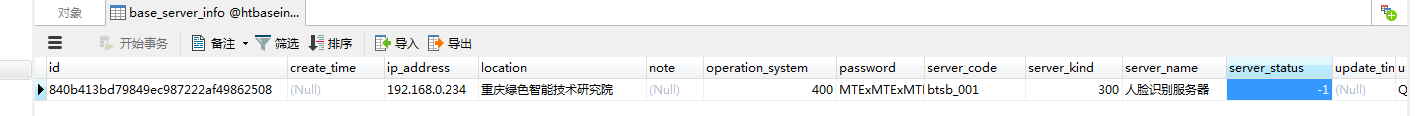








然后按Ctrl+s 保存



# 数据平台网关设置

由于数据平台的请求返回的数据量加大，数据平台的网关跟全流程的网关分开部署。

数据平台网关的请求耗时长，zuul服务器里面加入新的配置

#读取数据时间

ribbon.ReadTimeout=50000

#链接超时时间

ribbon.ConnectTimeout=5000

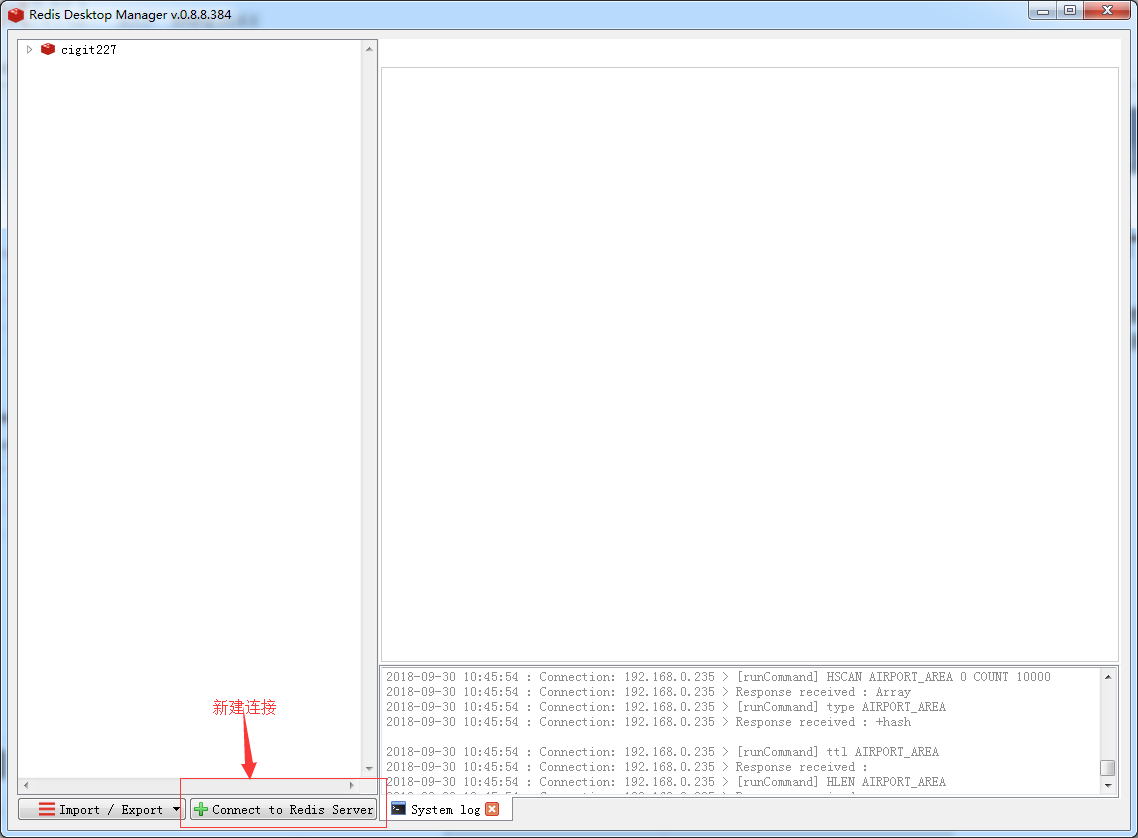
#断路器超时时间，必须大于读取时间+响应时间

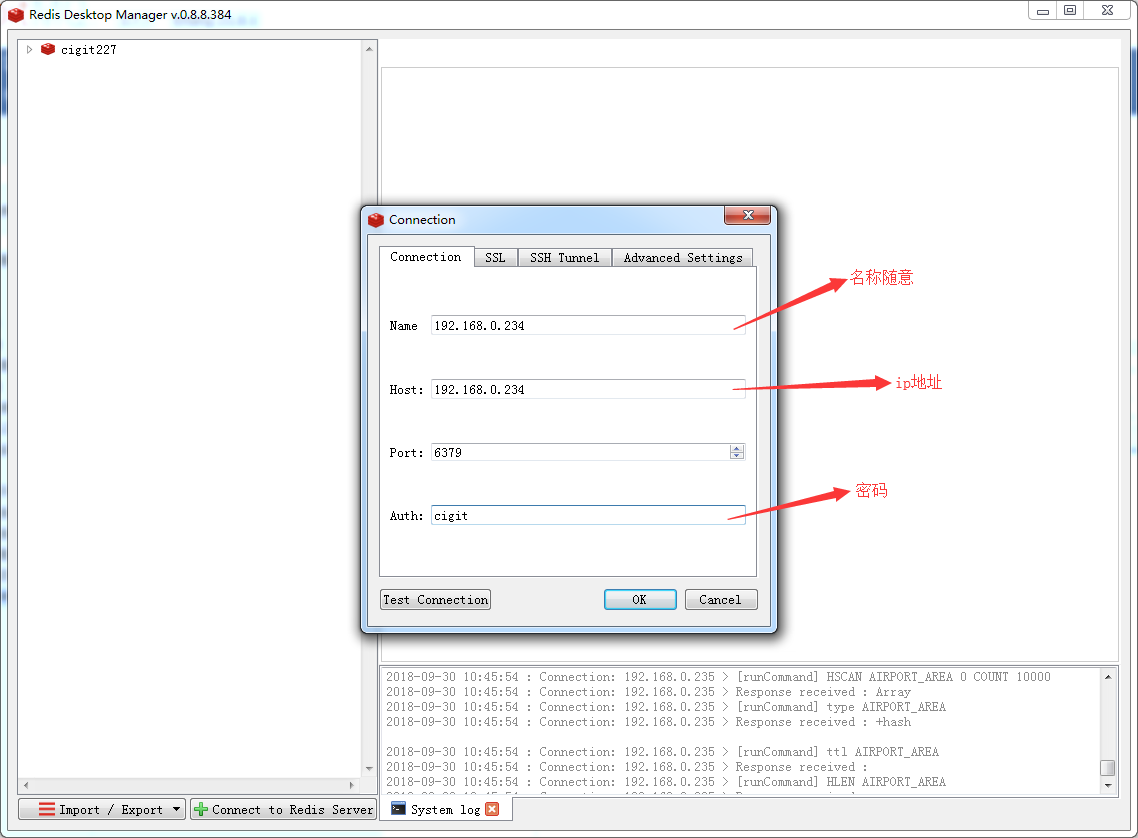
hystrix.command.default.execution.isolation.thread.timeoutInMilliseconds=60000

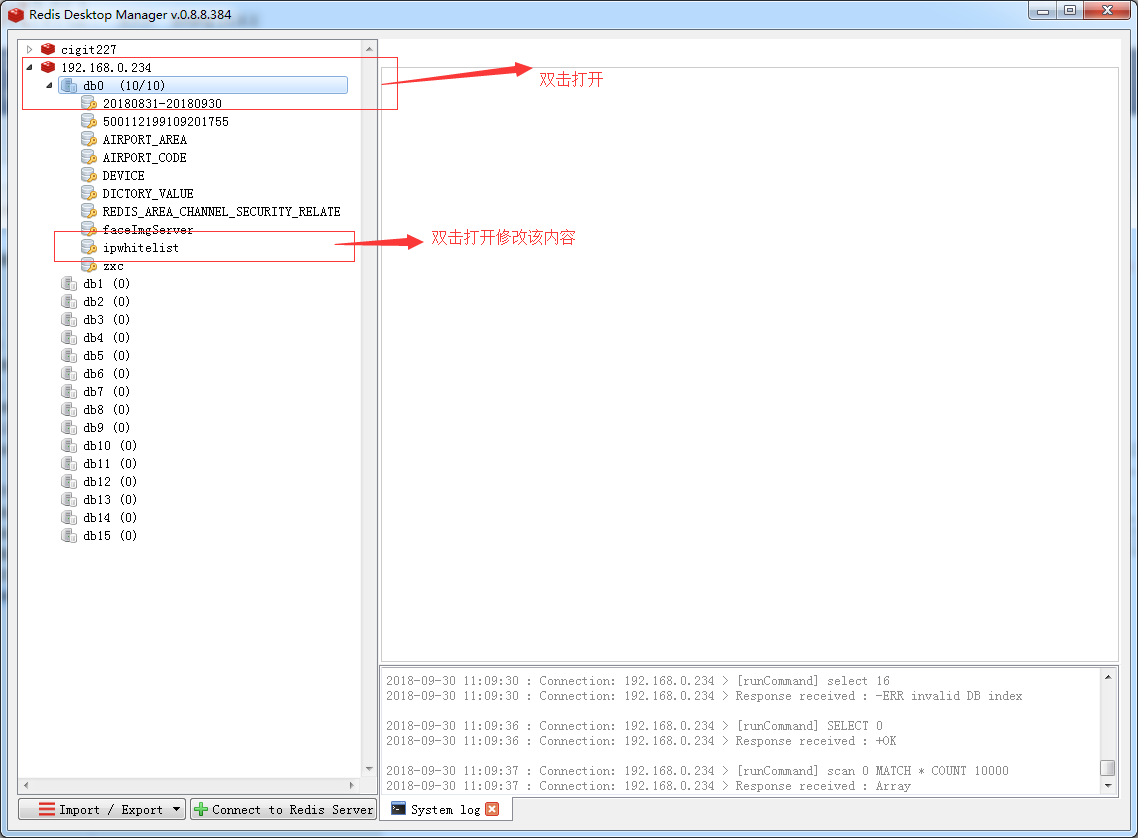
# 网关白名单设置

下载软件 Redis Desktop Manager 并安装

登录redis，如图：









修改后，点击 Reload Value ，如果发现值仍未改变，说明当前ip的redis服务不是主节点。按照上述说明，登录到另外一台ip上的redis进行相同的操作即可。

1. [↑](#endnote-ref-0)