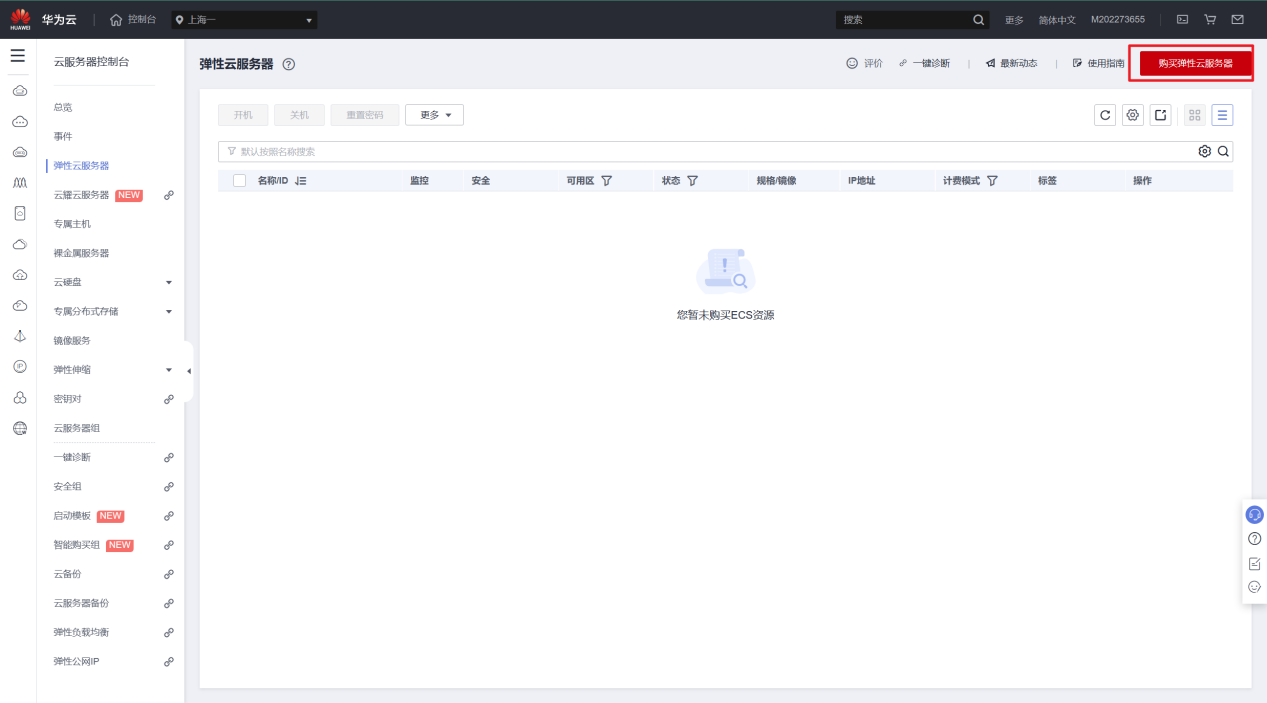
# 实验环境部署手册

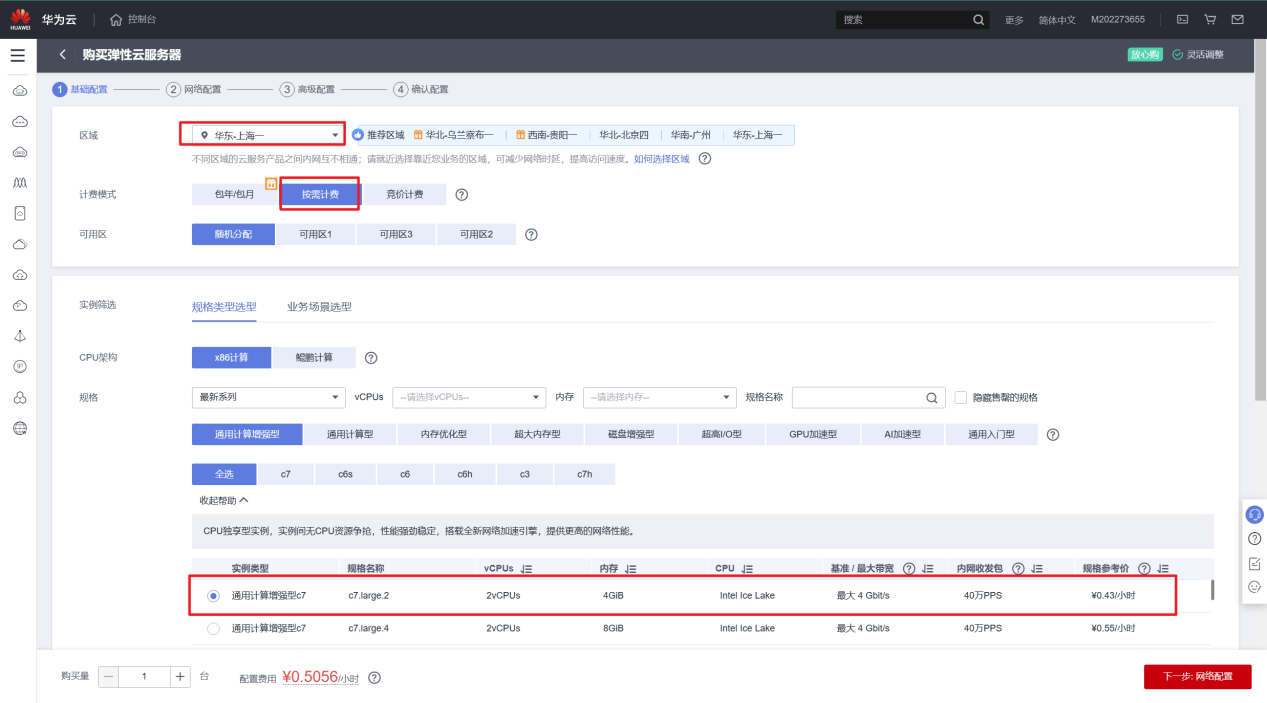
## 华为云Linux服务器环境搭建

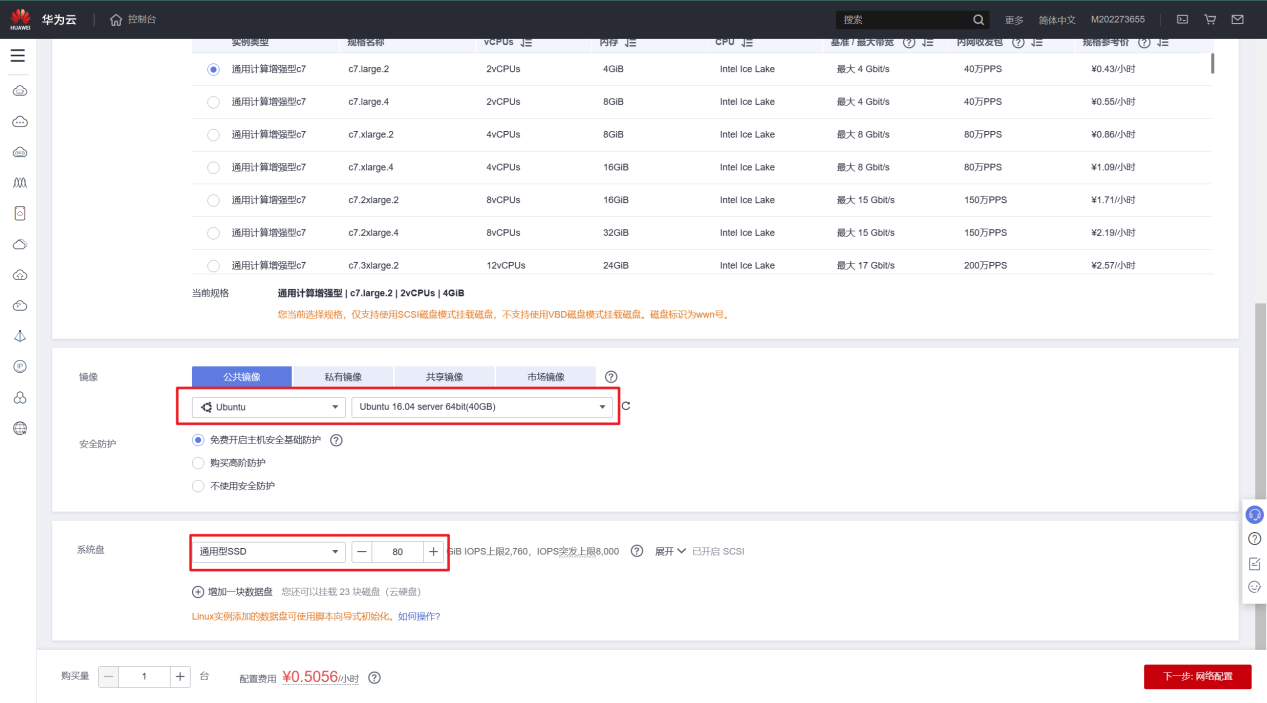
### 购买云服务器

1. 登录华为云官网购买弹性云服务器ECS。

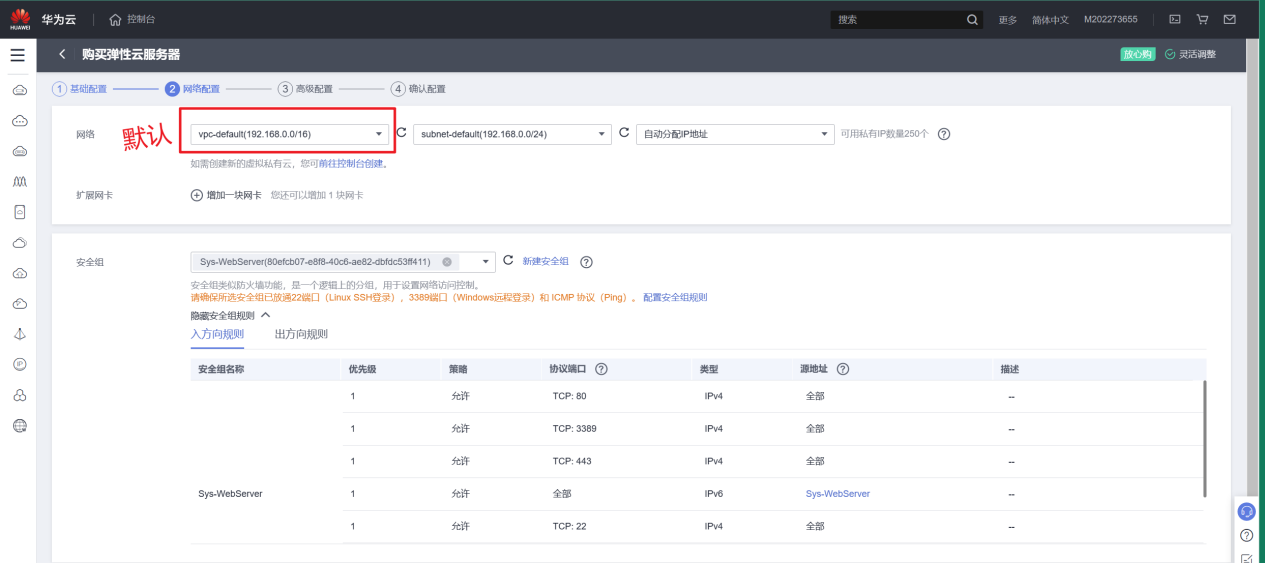


1. 计费模式选择【按需计费】，区域选择【华东-上海一】(可选其它区域，选择其它区域可能会在下数据集的时候慢，建议选相同区域)，选择Linux系统(Ubuntu16.04)镜像（配置最低为2核，内存4GB，硬盘80GB），详细设置如下所示。



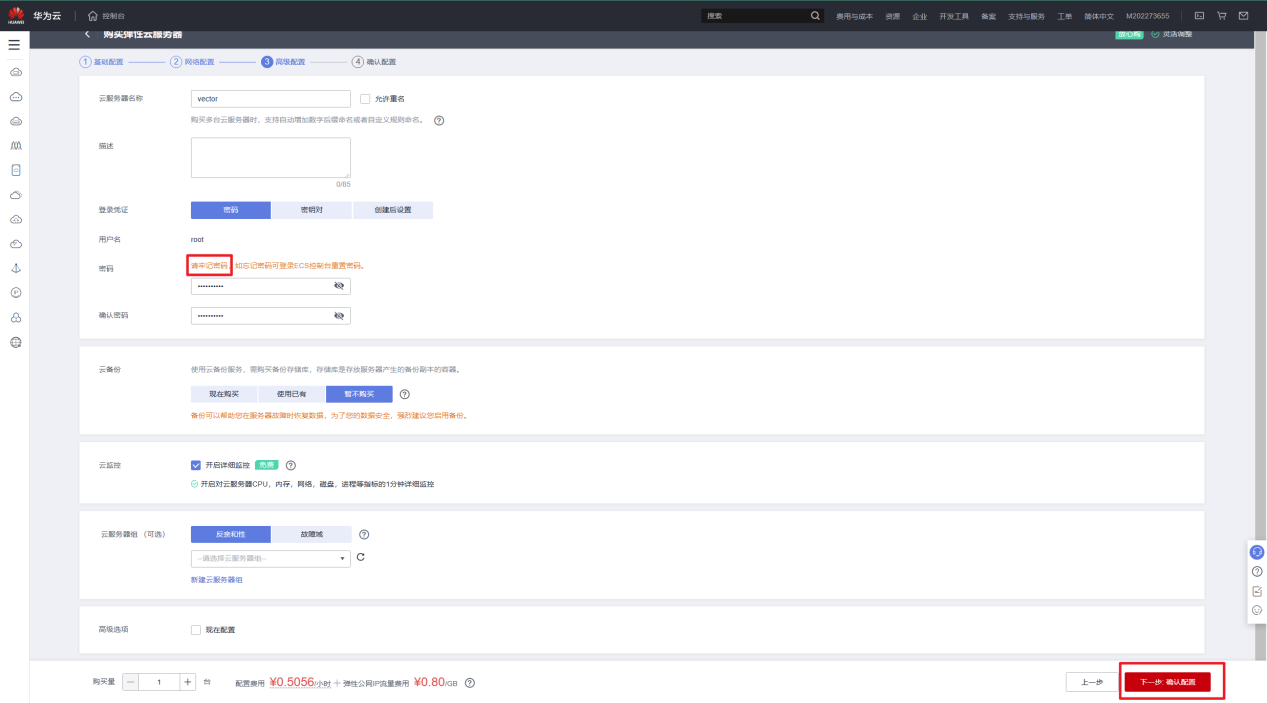


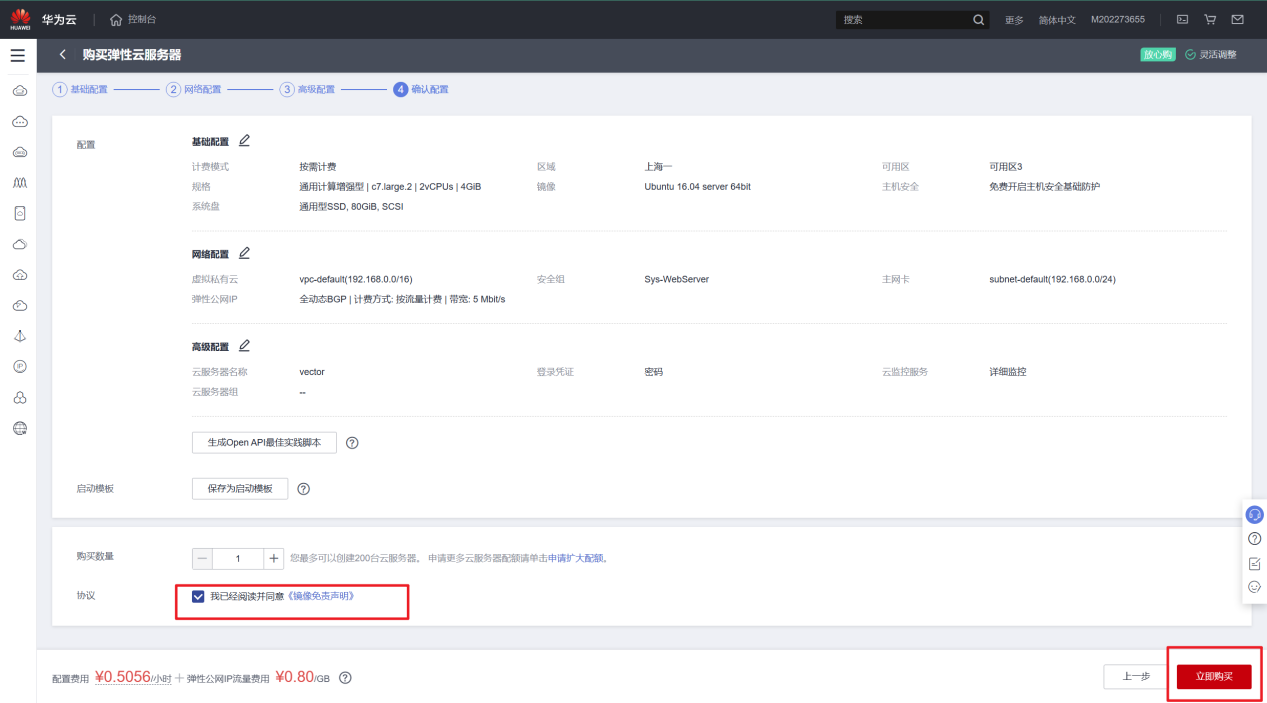
进入网络配置



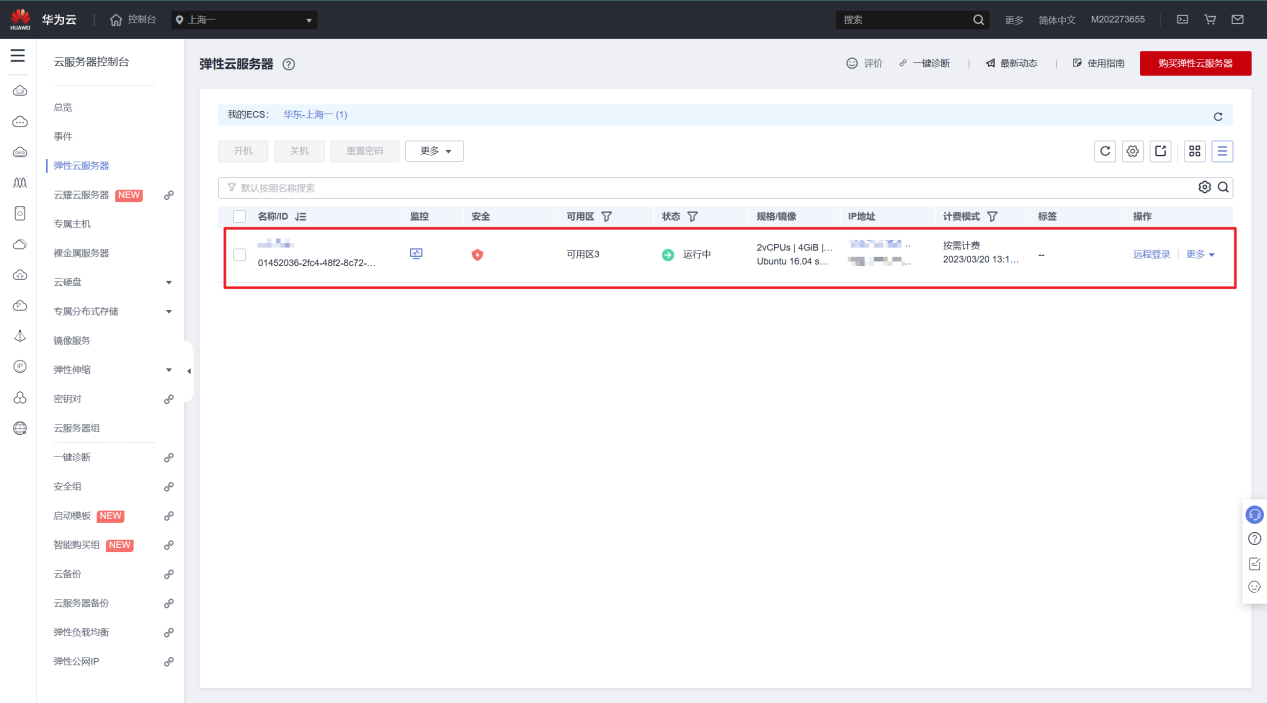


可以自己设定服务器名称，设置密码，密码要记住。

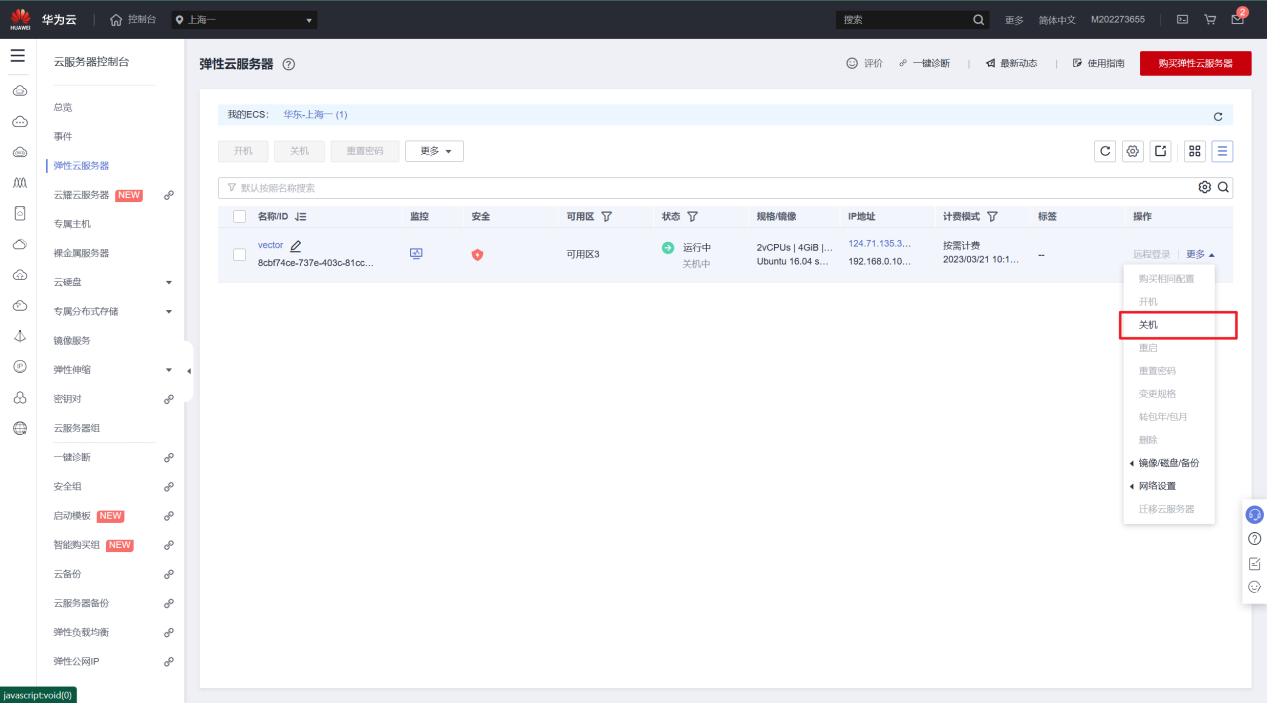




1. 支付完成后，进入华为云控制台，找到自己的服务器，其中弹性公网IP地址需要记下来，之后连接会使用。点击“名称”可进入更详细配置。



所选为按需计费，华为云支持即开即停，秒级计费。如上图是启动状态，不使用的服务器，一定关机，关机能减少大量计费。普通实例关机后，基础资源(包括vCPU、内存、镜像)不计费。其他绑定资源（云硬盘、弹性公网IP）正常计费，详细费用说明见[官网](https://support.huaweicloud.com/productdesc-ecs/ecs_01_0065.html)。

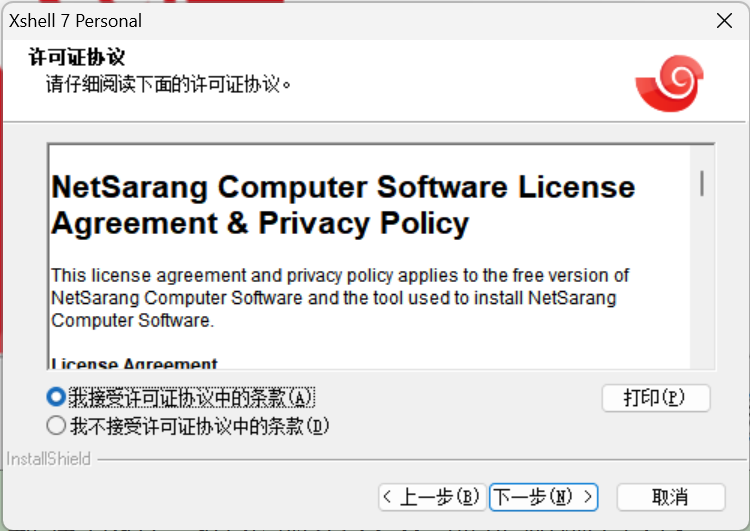


### 安装Xshell

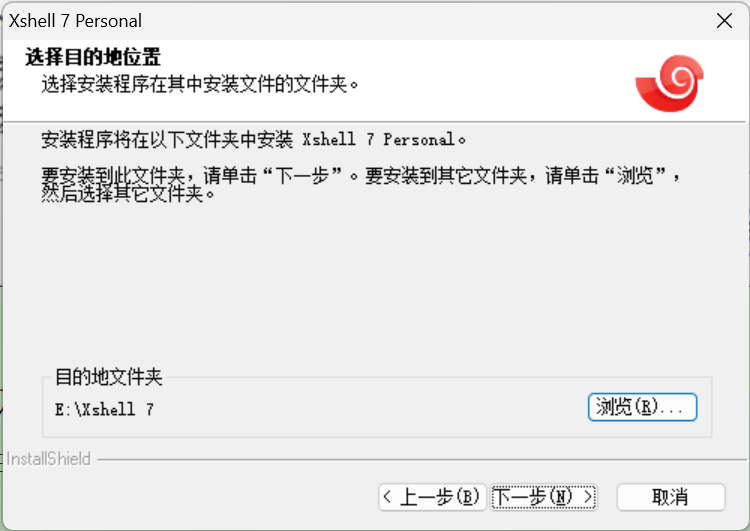
Windows用户在本地机器上[下载安装Xshell](https://cdn.netsarang.net/df59d221/Xshell-7.0.0122p.exe)，实现远程登录服务器。

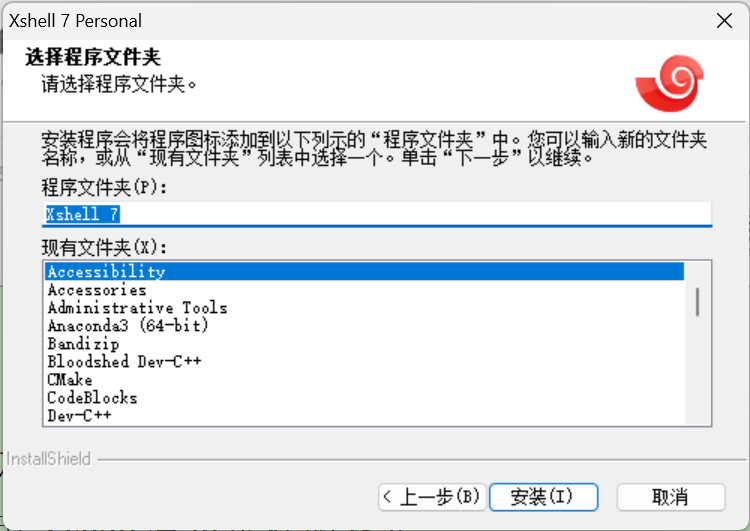
若本地机器操作系统为Linux，见3.2节。



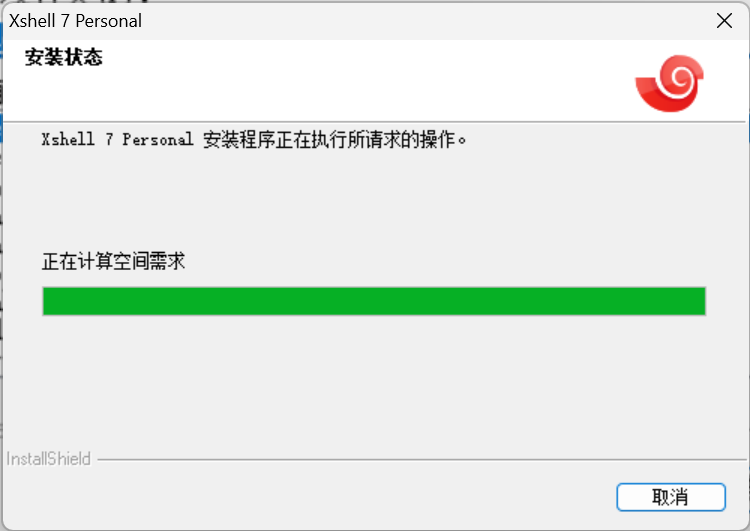


选择安装路径





点击安装，等待安装完成

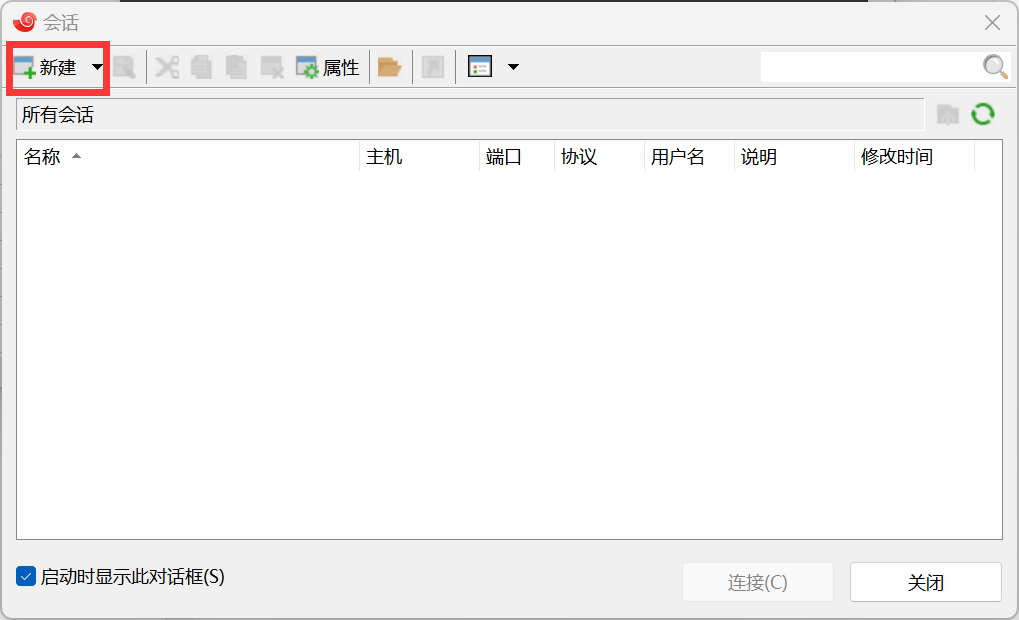




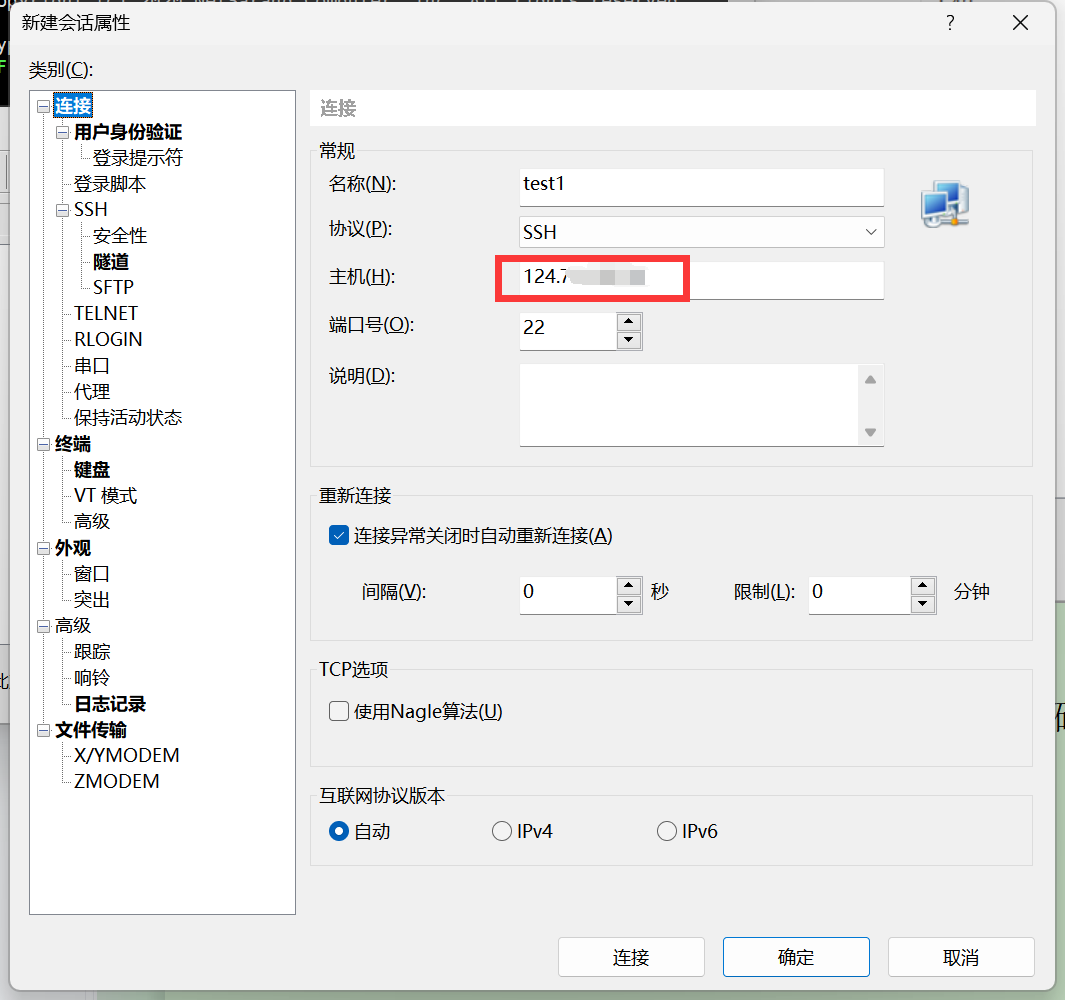
### 建立连接

### 3.1 使用Xshell建立连接

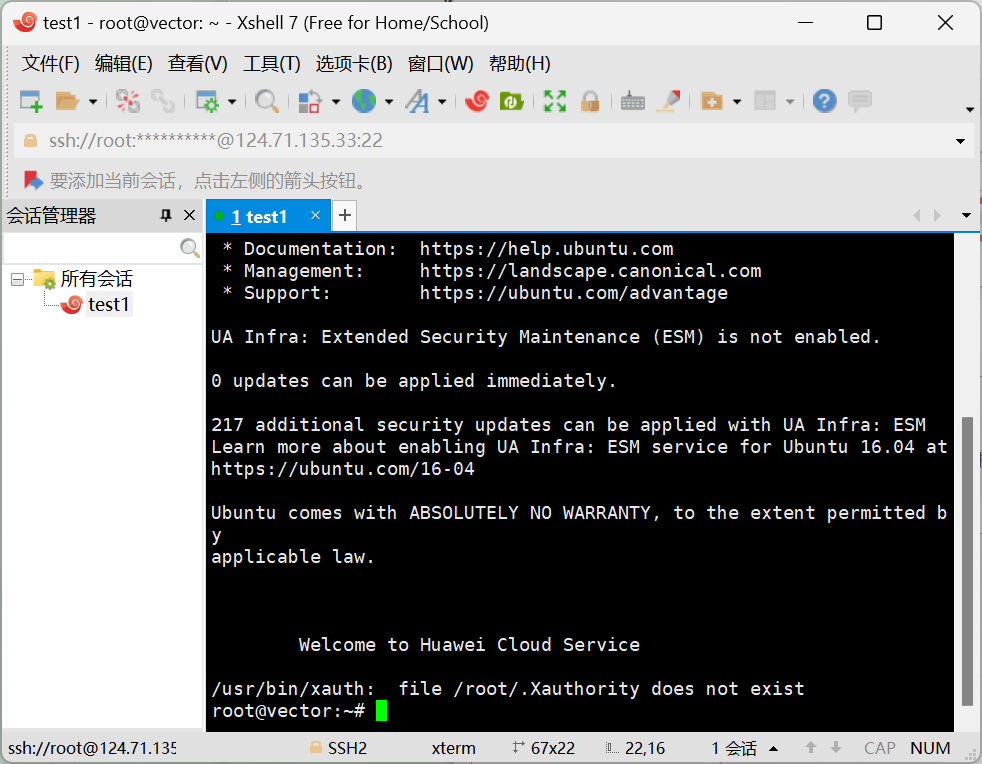
新建连接，指定服务器IP地址和端口号：



IP地址为1.3步中的IP地址（弹性公网IP地址）。



点击用户身份验证，输入用户名（默认为root）和设置的密码（服务器端可重置），最后点连接，使用Xshell登录远程服务器：



### 3.2 在Linux下使用ssh建立连接

Linux下建立连接使用ssh即可，首先需要安装ssh服务，以ubuntu24.04为例，首先执行以下命令：

sudo apt update

sudo apt install openssh-server

不同的发行版使用的包管理工具不同，需要将’apt’替换为自己Linux发行版所使用的包管理工具。

首先要启动ssh服务，在终端执行以下命令：

sudo systemctl start ssh

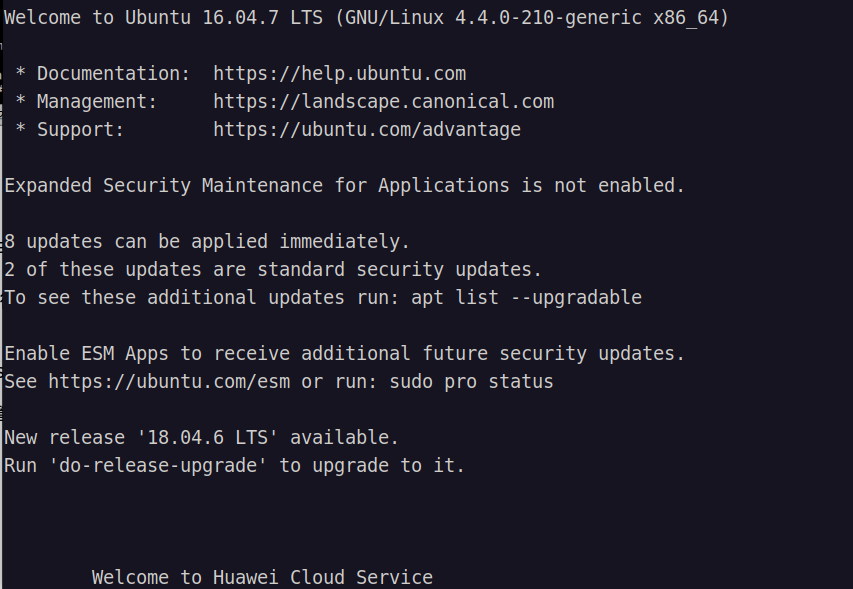
然后执行下面的命令：

ssh <用户名>@<此处是弹性公网IP>

比如以我的部署为例，我的用户名是root，应当在终端输入下面的命令：

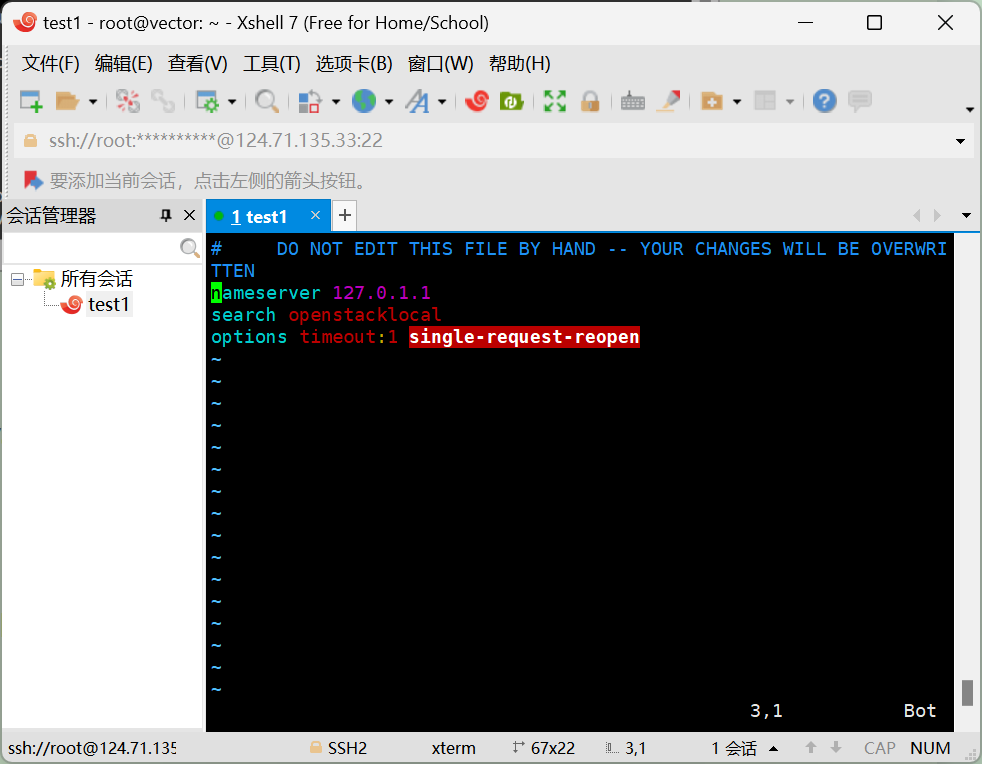
ssh [root@xxx.xx.xxx.xx](mailto:root@<123.xx.xxx.xx>)

执行后出现下面的输出即为执行成功：

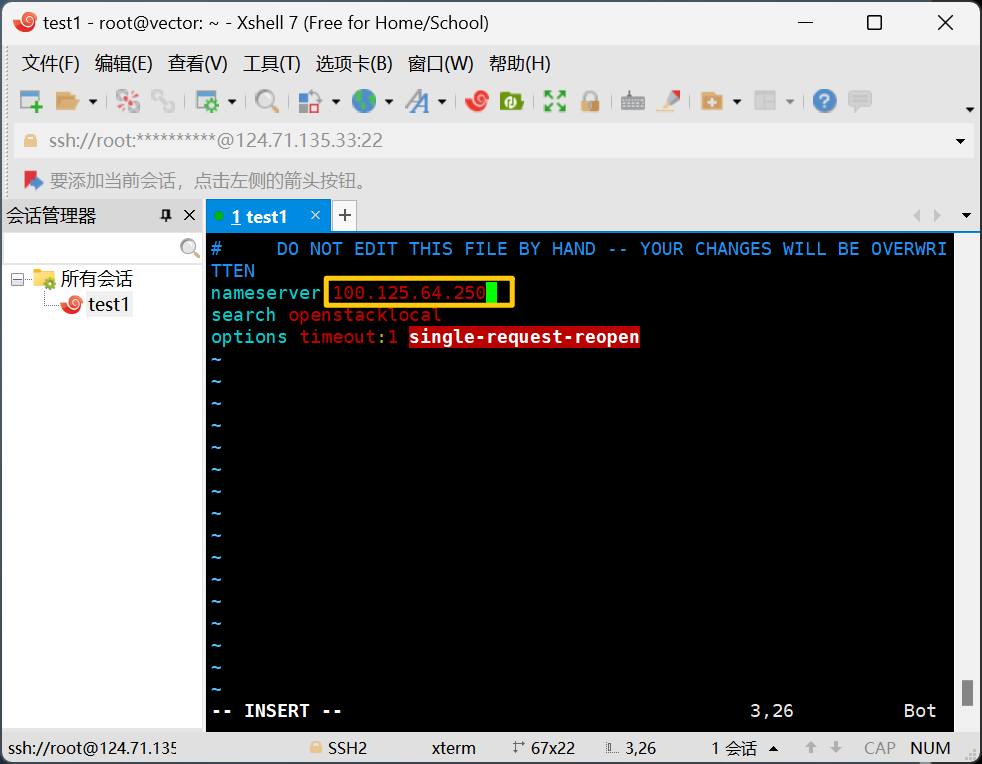


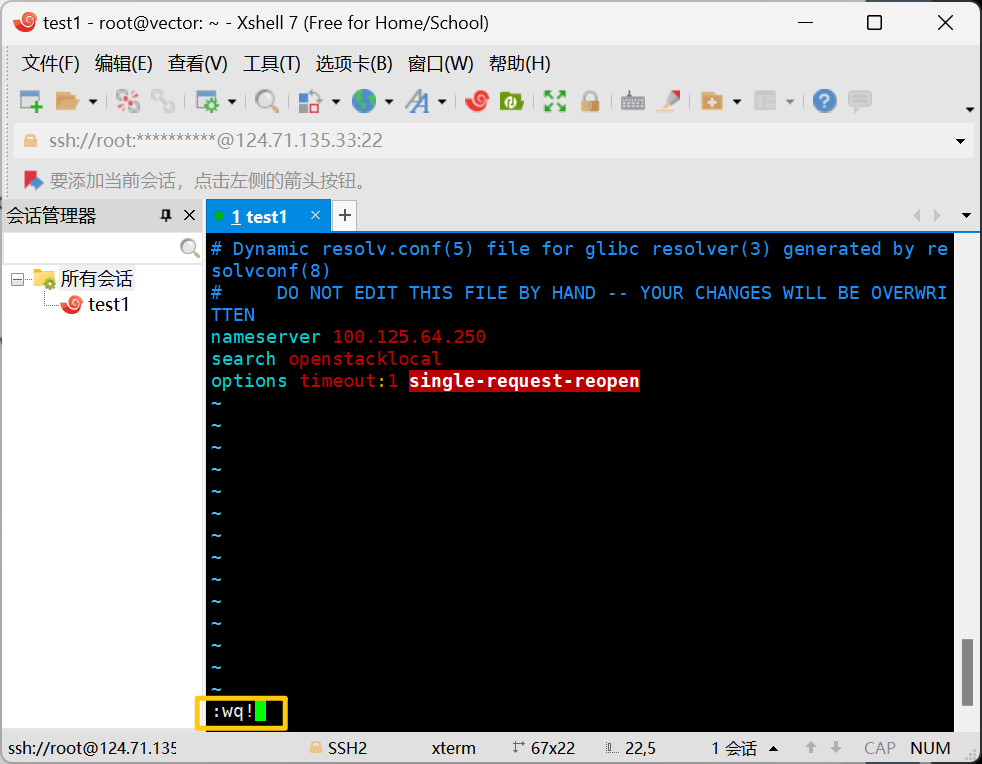
### 4. 增加内网配置

命令行输入vim /etc/resolv.conf，服务购买的是上海1的增加ip地址为100.125.64.250，选择其它区域服务器的参看[官网](https://support.huaweicloud.com/dns_faq/dns_faq_002.html)。



先按i键，转换为插入模式，然后增加内网地址，按esc后输入**:wq!**回车保存并退出。

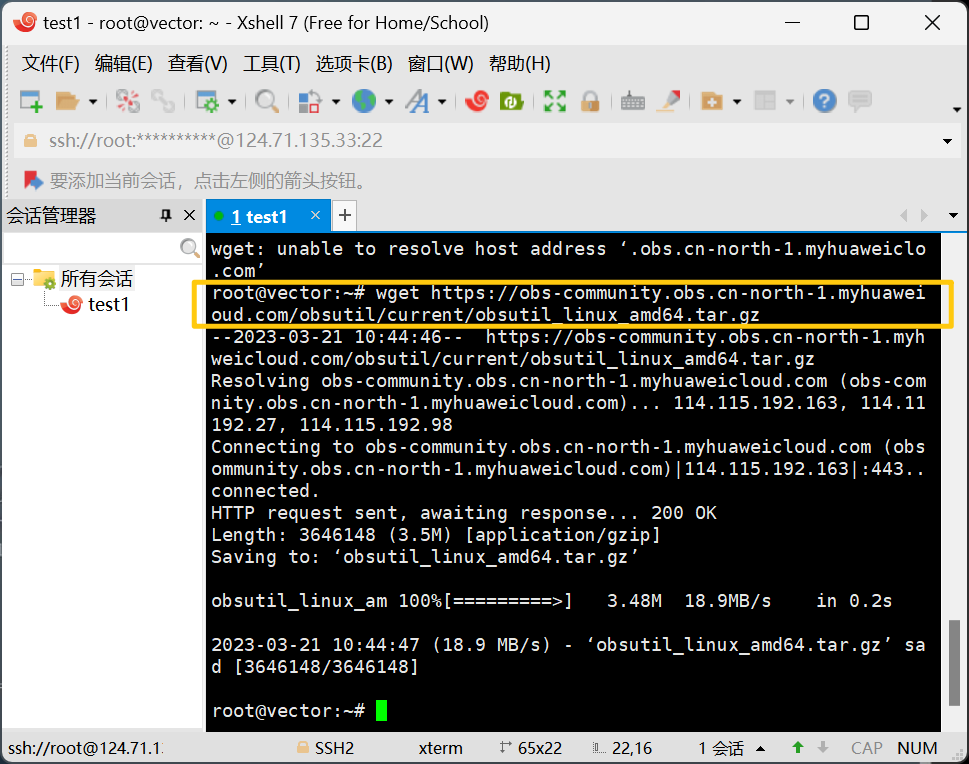




### 5. 安装obsutil以快速下载yelp数据集

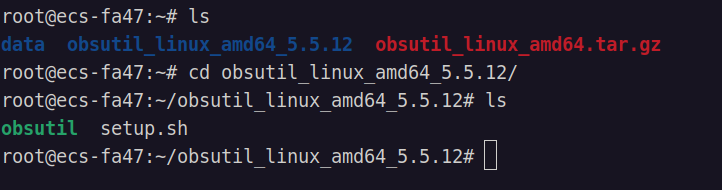
1. 输入下载命令

wget https://obs-community.obs.cn-north-1.myhuaweicloud.com/obsutil/current/obsutil\_linux\_amd64.tar.gz



1. 解压缩

tar -zvxf obsutil\_linux\_amd64.tar.gz



解压缩后可以用ls命令查看解压目录下的文件变化

1. 进入该目录配置obsutil

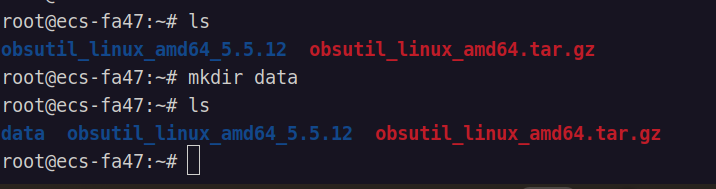
需要OBS终端节点地址(Endpoint)和访问密钥（AK和SK），选择上海1区域服务器的输入命令：

./obsutil config -i=12HHX8MOGUI9C4BBYDA6 -k=HsN22sShsa5S3bu0izyF0tHqTxYRhR6wPIXGAKeC -e=obs.cn-east-3.myhuaweicloud.com



1. 如何下载数据集

下面是如何下载obs中的数据集，在obsutil的同级目录下创建data文件夹用于存放下载的yelp数据集。



下载数据集命令为：

./obsutil cp 源地址 目标地址

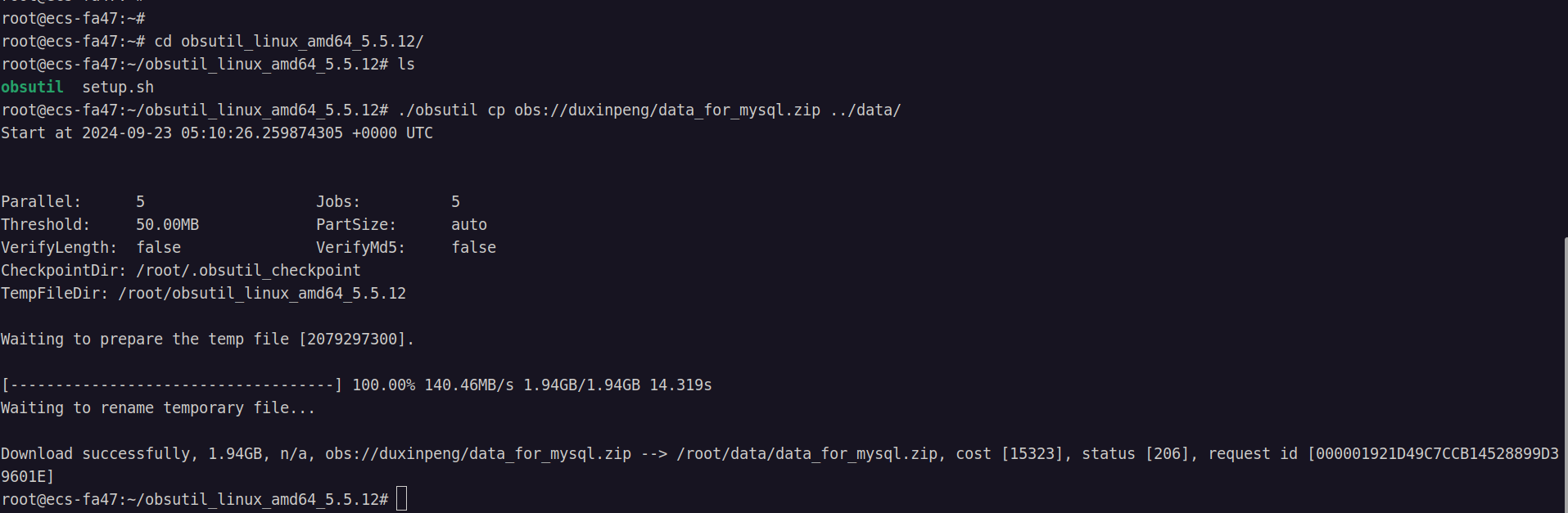
三个数据库数据集的源地址分别为：

obs://duxinpeng/data\_for\_mysql.zip

obs://duxinpeng/data\_for\_mongo.zip

obs://duxinpeng/data\_for\_neo4j.zip

下面是下载data\_for\_mysql.zip数据集的示例：



下载完后cd到data文件夹查看目标地址：



## 安装MongoDB

### 准备,更新软件源并安装mongodb

**(1)**准备安装MongoDB4.4版本，官网查找软件源公钥并导入：

wget -qO - https://www.mongodb.org/static/pgp/server-4.4.asc | sudo apt-key add -

该命令响应OK即为执行成功。

**(2)**为mongodb创建软件源list文件:

echo "deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu xenial/mongodb-org/4.4 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-4.4.list

**(3)**然后执行update操作刷新镜像源：

sudo apt update

**(4)**然后执行下面的命令安装即可：

sudo apt install mongodb-org

### 启动、重启和关闭服务命令

启动服务：sudo systemctl start mongod

重启服务：sudo systemctl restart mongod

关闭服务：sudo systemctl stop mongod

1. **使用mongdb**

首先启动mongod服务：

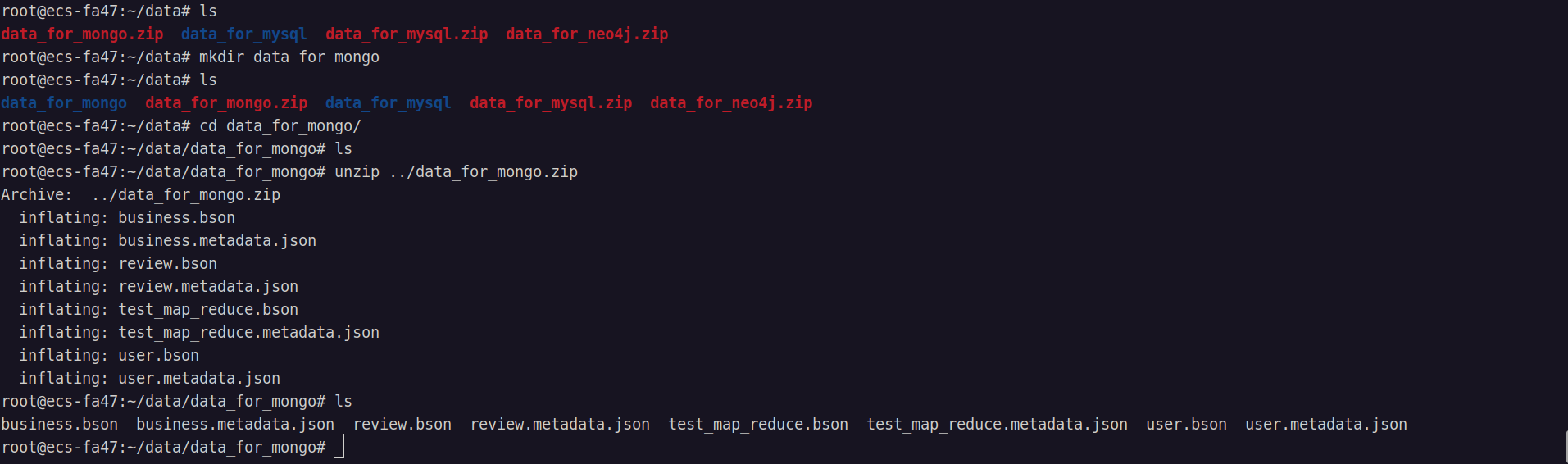
sudo systemctl start mongod

然后执行mongo即可启动mongodb



### 4. 导入数据

在data文件夹之下创建data\_for\_mongo文件夹，并将mongo的zip文件解压到data\_for\_mongo文件夹之下：

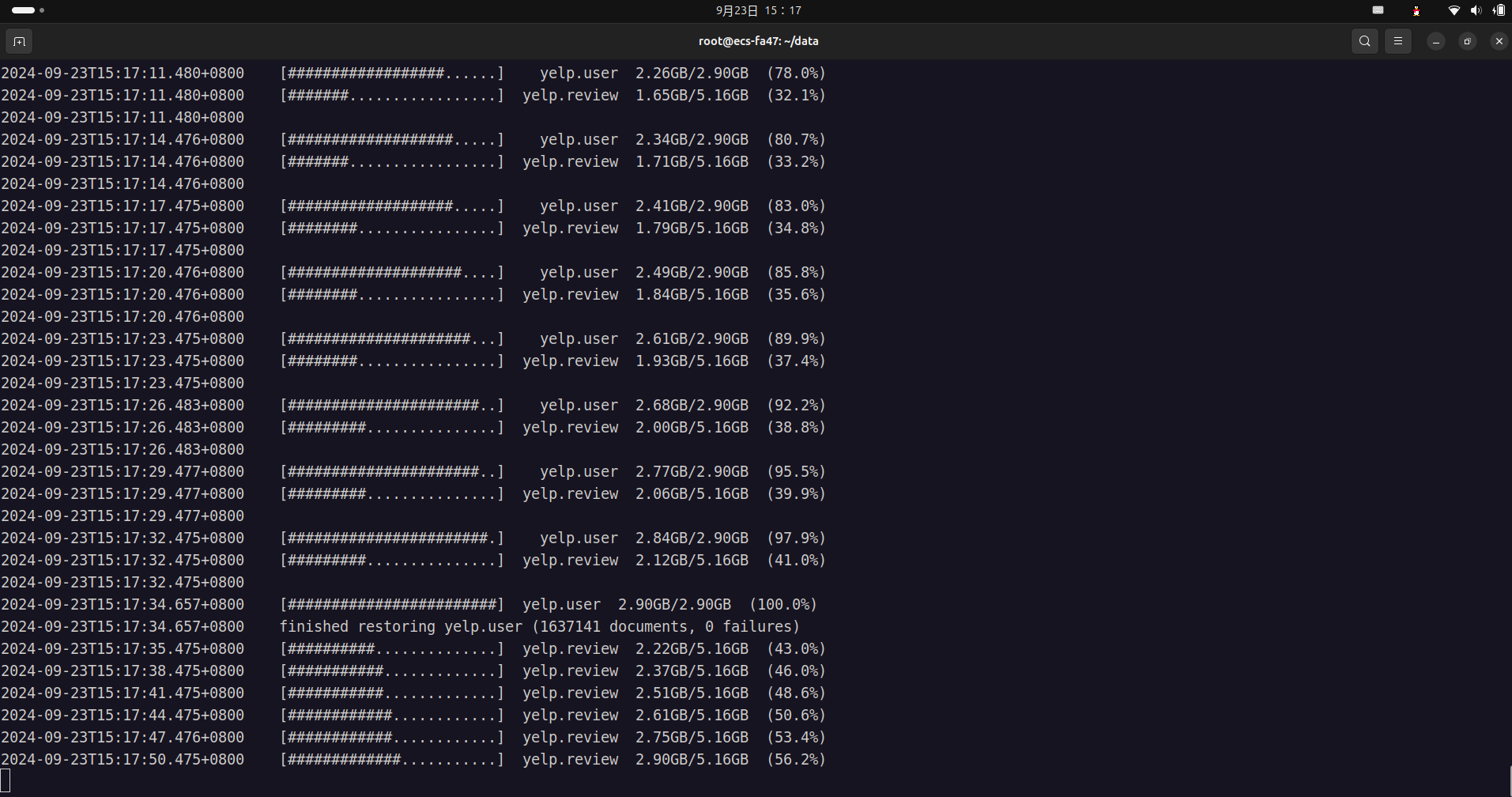


然后将文件目录切换到data下

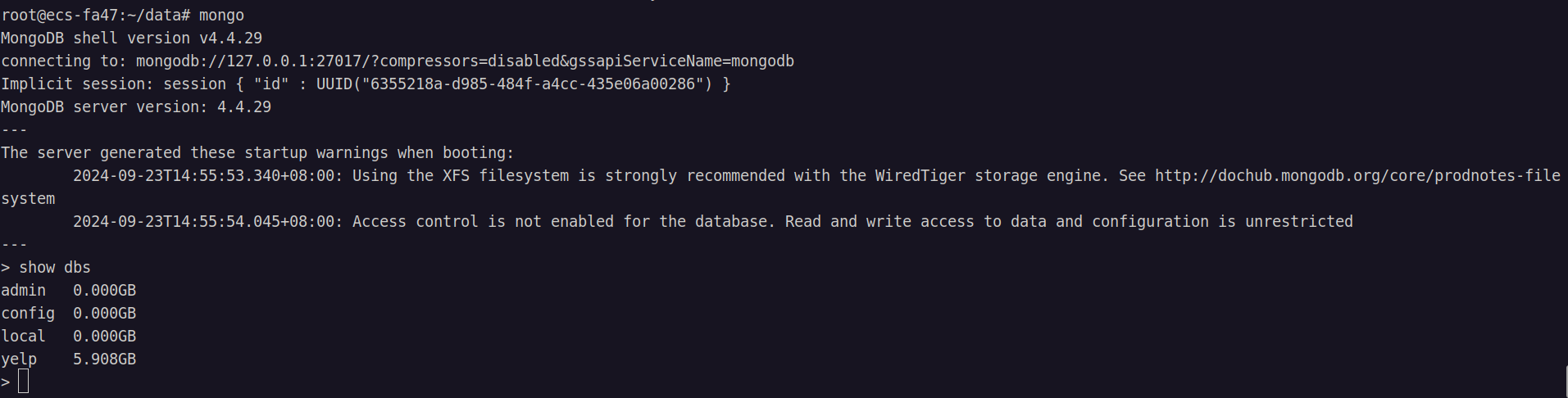
执行下面的命令：

mongorestore -h localhost -d yelp --dir data\_for\_mongo

可以看到正在导入：



此时启动mongo，查看数据库：



可以看到导入成功。

### 5. 为MongoDB设置密码

默认安装MongoDB之后没有密码，这样的话服务器的MongoDB端口如果没有关闭很可能受到攻击（有同学遇到过），所以强烈建议为MongoDB设置密码，流程如下所示。

（1）use admin //切换到admin

db.createUser({

user: 'admin', // 用户名（自定义）

pwd: 'xxx', // 密码（自定义）

roles:[{

role: 'root', // 使用超级用户角色

db: 'admin' // 指定数据库

}]

})

db.createUser({

user: 'qitu', *// 用户名（自定义）*

pwd: 'QWer2004', *// 密码（自定义）*

roles: [{

role: 'root', *// 使用超级用户角色*

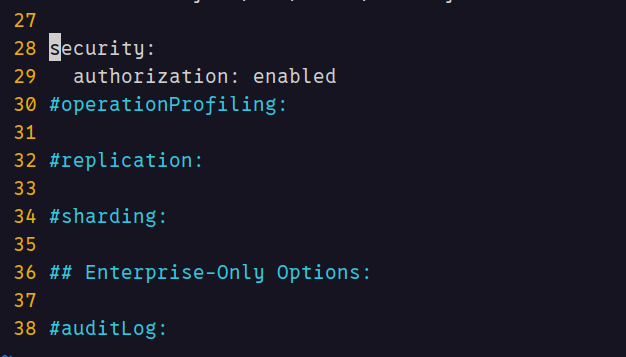
db: 'admin' *// 指定数据库*

}]})

1. 修改配置文件

使用vim /etc/mongod.conf进入配置文件

将security反注释，然后在下面添加如下的内容：（注意缩进是两个空格，否则可能配置失败）



1. 登录命令

mongo admin -u admin -p xxx(这里是你的密码)

mongo admin -u qitu -p QWer2004

## 安装Neo4J

### 安装配置java11环境

1. 打开终端，添加PPA存储库

sudo add-apt-repository ppa:openjdk-r/ppa

1. 更新apt-get包管理器的软件包索引

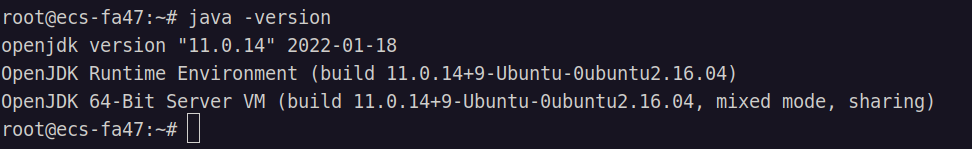
sudo apt-get update

1. 安装OpenJDK 11

sudo apt-get install openjdk-11-jdk

1. 检查Java版本是否正确安装

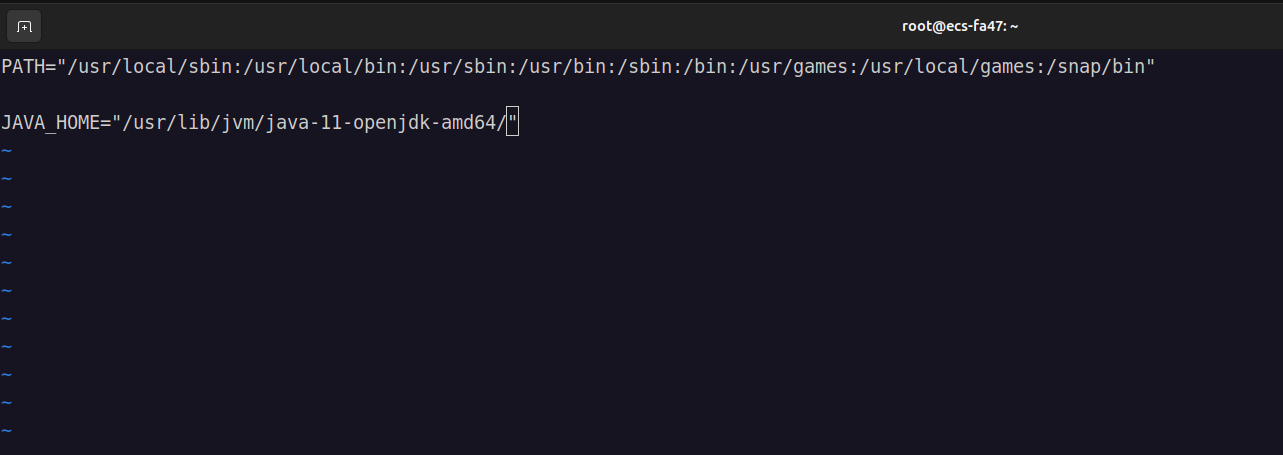
java -version



1. 设置JAVA\_HOME环境变量

使用vim编辑/etc/environment文件，添加以下内容：

JAVA\_HOME="/usr/lib/jvm/java-11-openjdk-amd64/"



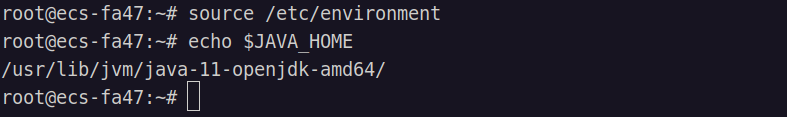
按Esc，然后输入:wq保存并退出。

1. 更新系统环境变量

source /etc/environment

1. 检查JAVA\_HOME环境变量是否设置正确

echo $JAVA\_HOME



### 安装Neo4J

neo4j-community-4.0.9-unix.tar.gz已经上传到了我创建的桶中，可以使用obs快速下载。

首先进入obsutil\_linux\_amd64\_5.5.12目录之下，然后执行下面的命令：

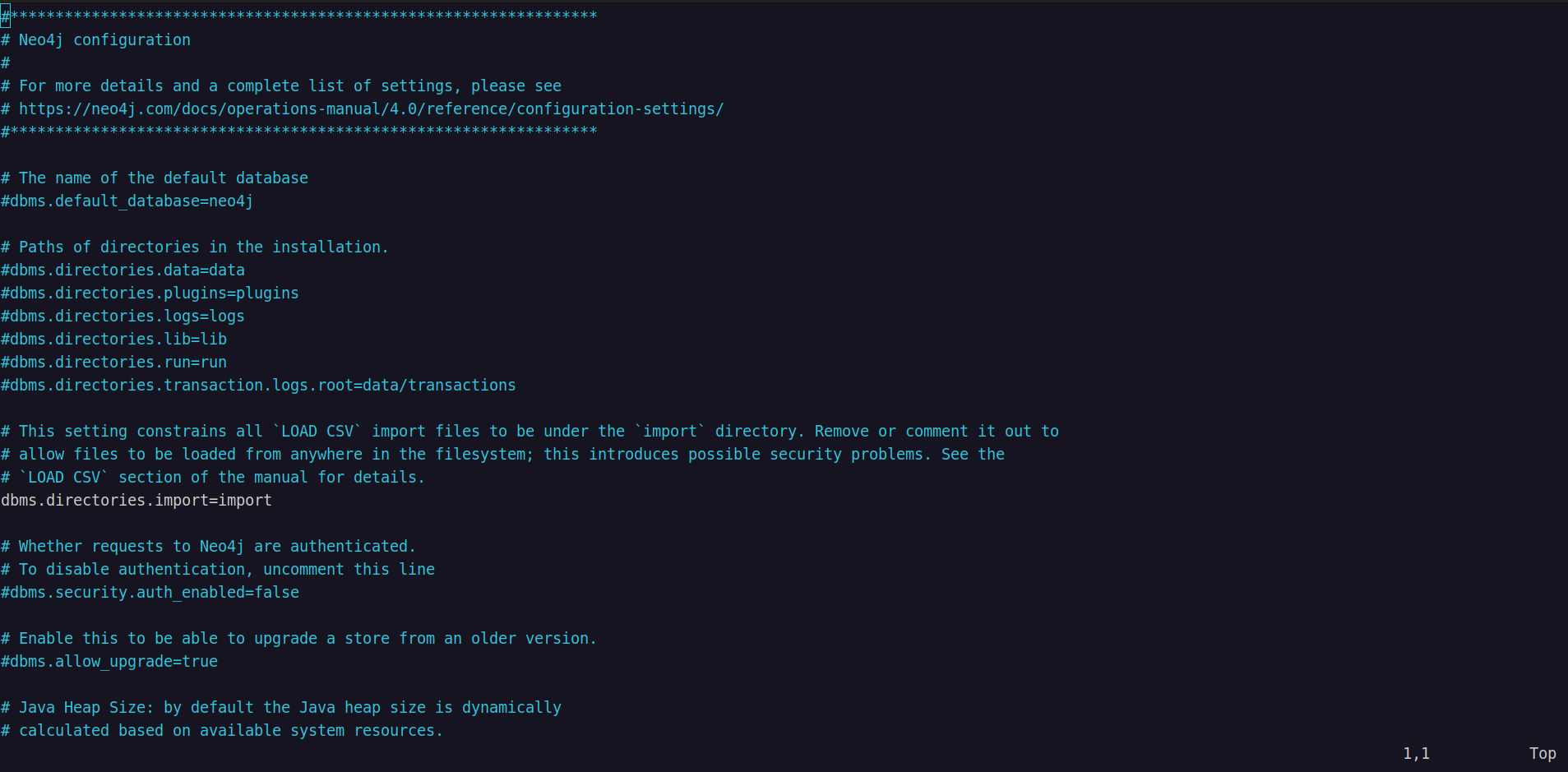
./obsutil cp obs://duxinpeng/neo4j-community-4.0.9-unix.tar.gz /root/



进入root目录下，使用如下命令解压：

tar -zxvf neo4j-community-4.0.9-unix.tar.gz

进入目录解压后文件夹中的conf文件夹中，然后使用vim打开neo4j.conf文件



打开之后输入 “:set nu ”打开行号显示

找到第80行:

#dbms.connector.http.listen\_address=:7474

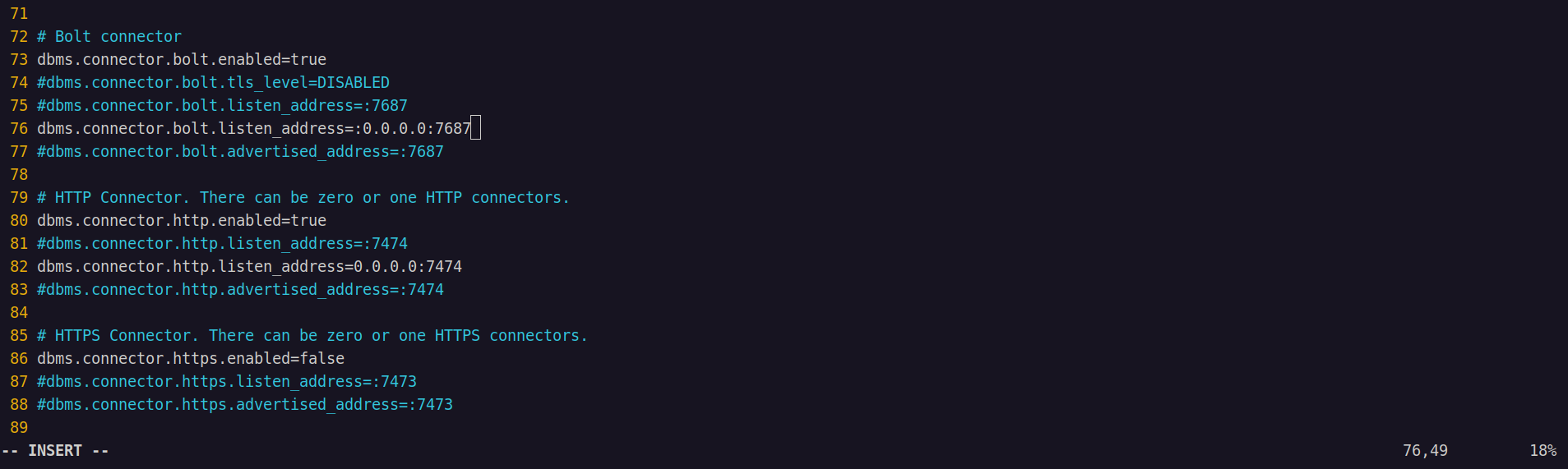
在它下面添加一行：

dbms.connector.http.listen\_address=0.0.0.0:7474

找到第75行：

#dbms.connector.bolt.listen\_address=:7687

在它下一行添加dbms.connector.bolt.listen\_address=0.0.0.0:7687



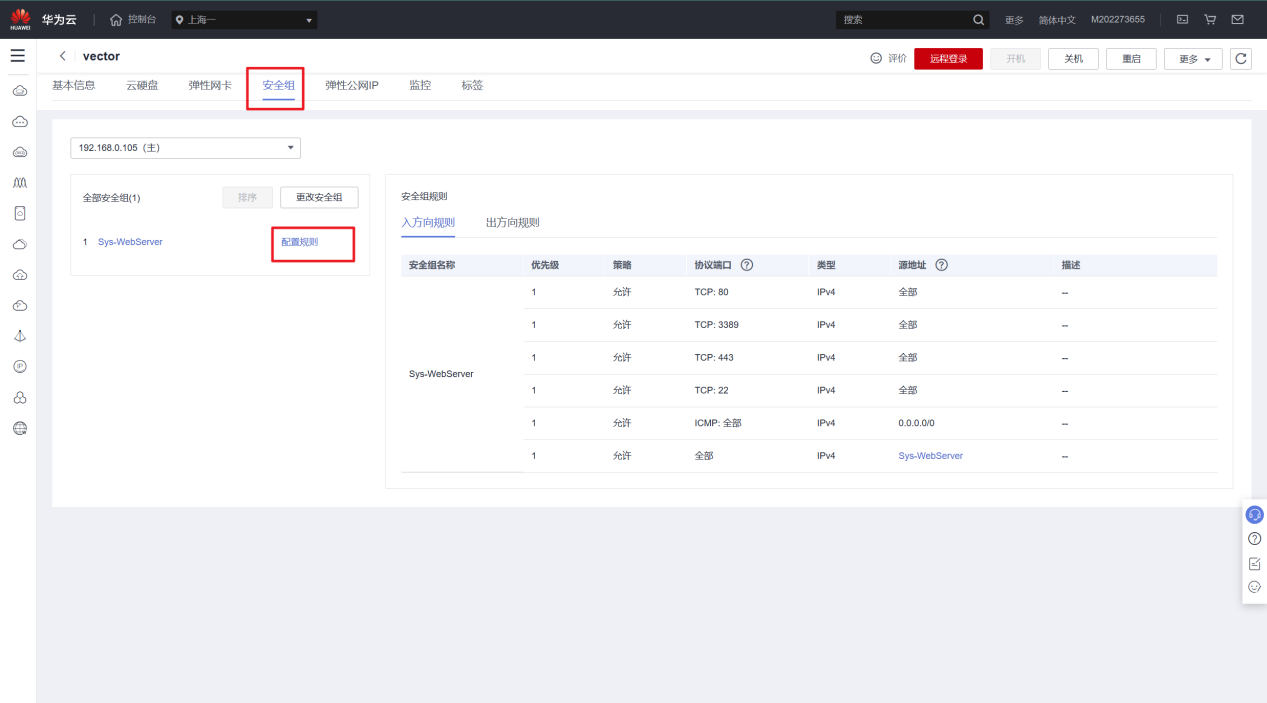
修改完毕，按Esc，然后输入“:wq”保存并退出即可。

### 华为云配置安全组规则

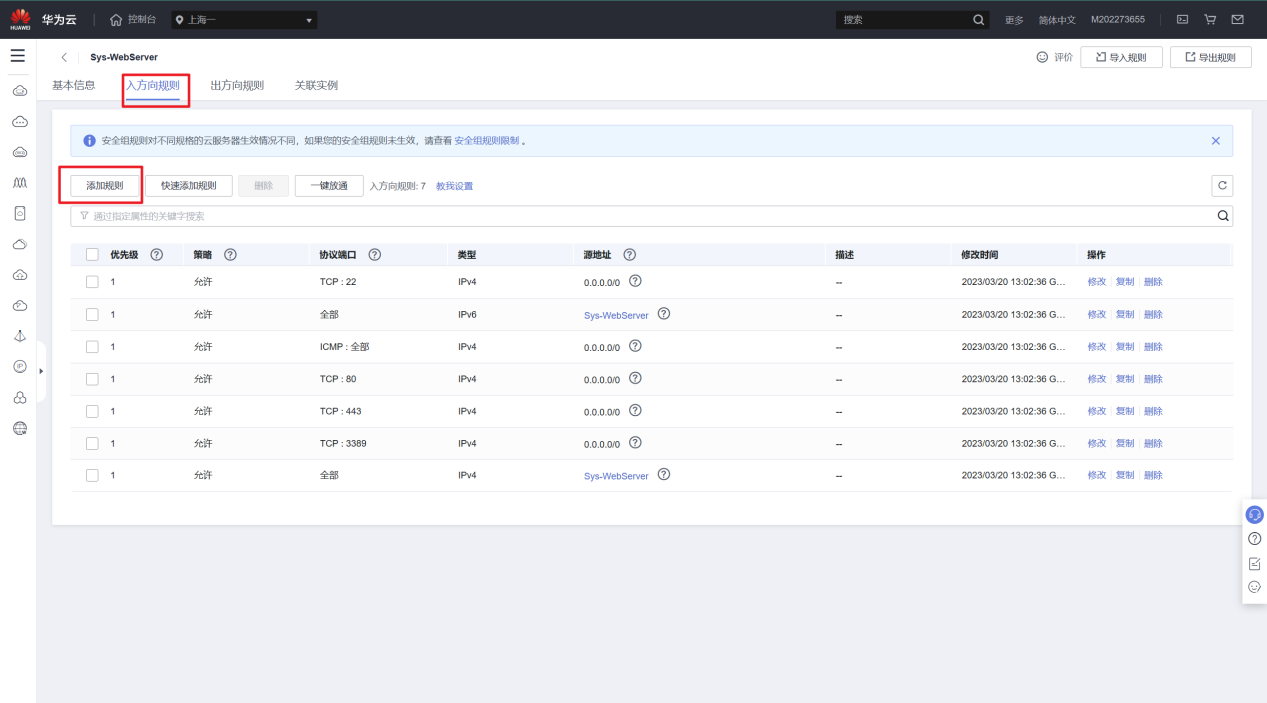
点击华为云控制台的服务器:



选择安全组，配置规则；



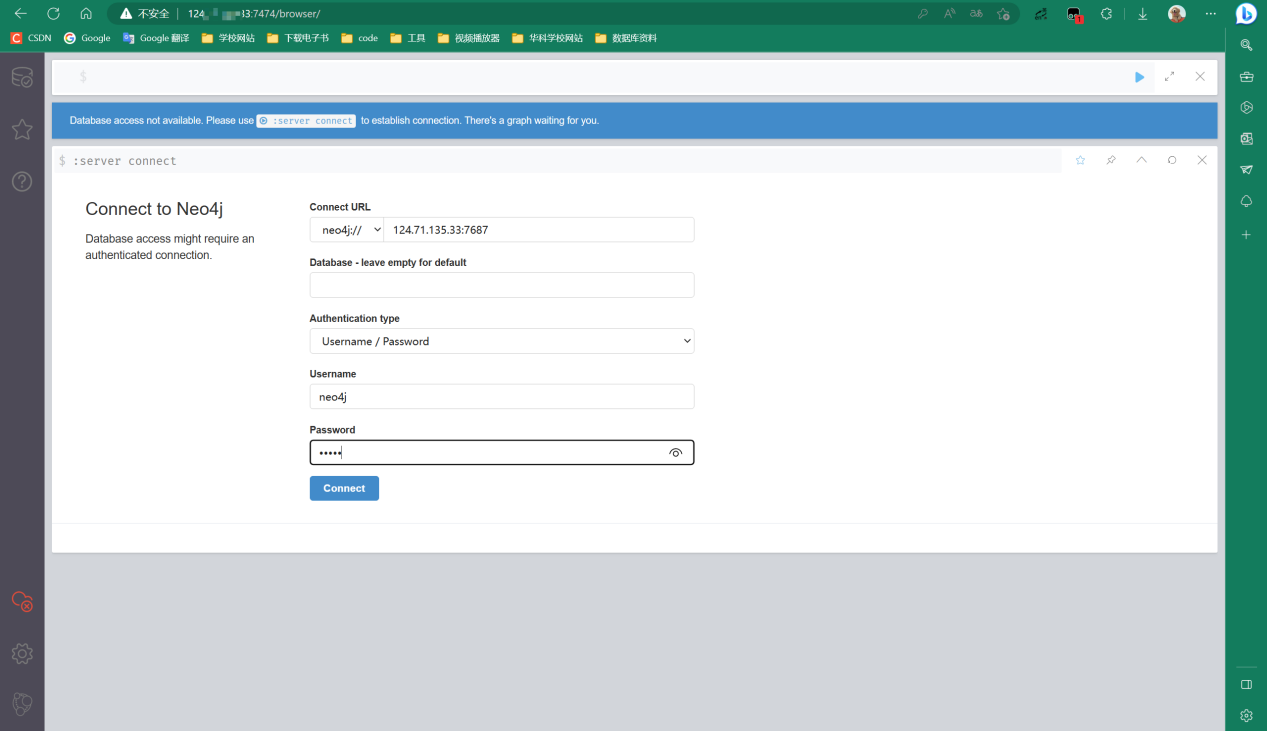
点击添加入方向规则，开放7474端口和7687端口:





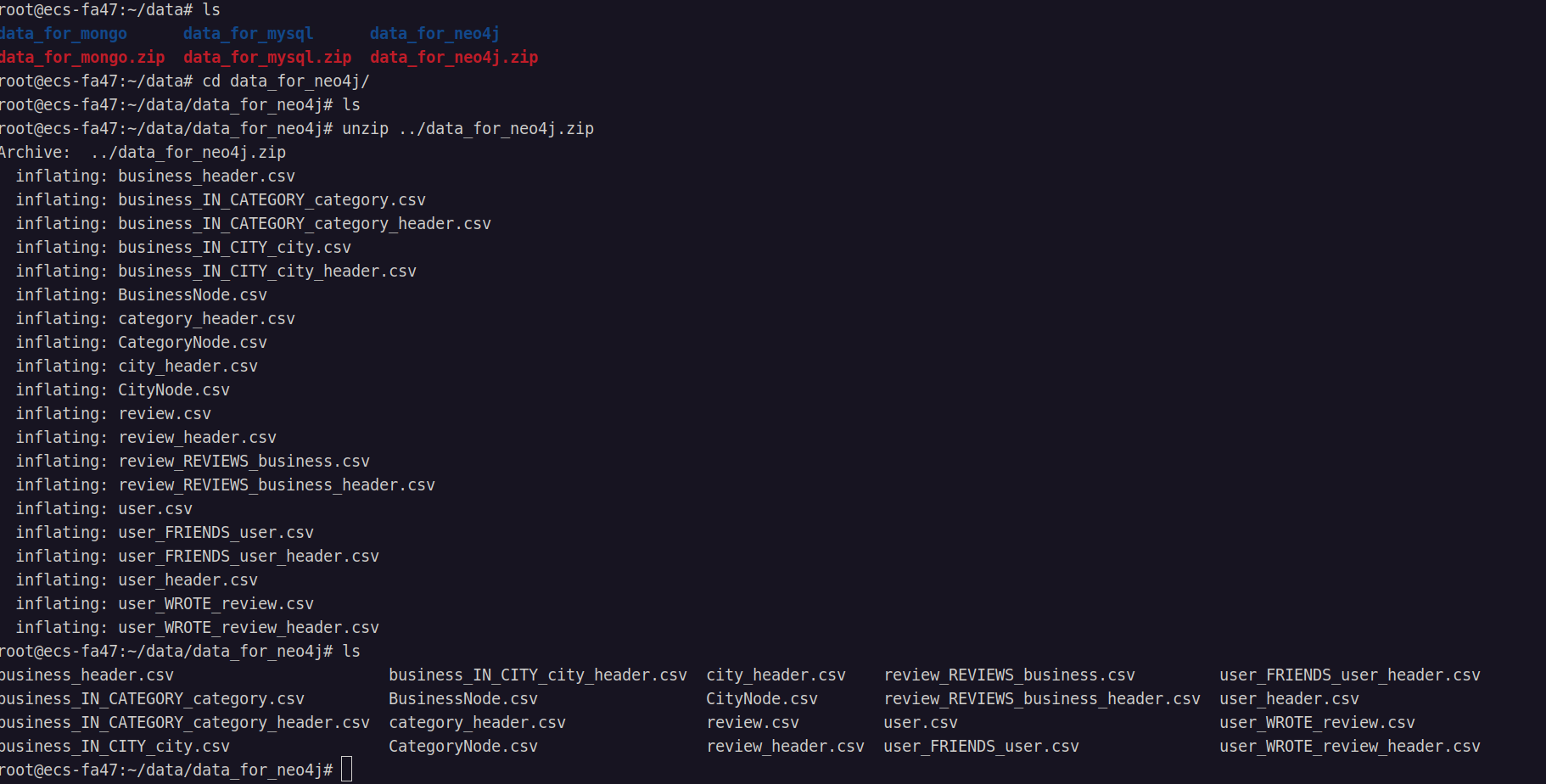
### 启动Neo4J

进入解压后的文件夹的bin目录下，使用./neo4j console或./neo4j start命令即可启动数据库，然后在浏览器中输入http://<服务器IP地址>:7474/ 即可访问，初始时用户名和密码均为neo4j。



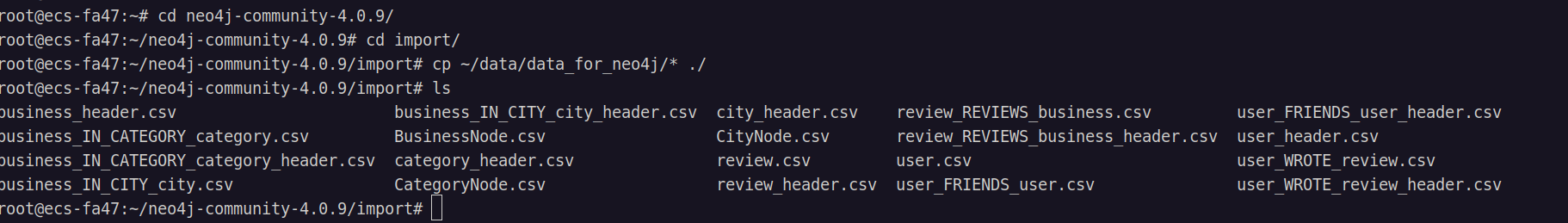
### 数据导入

在data目录下创建文件夹data\_for\_neo4j，并将data\_for\_neo4j.zip解压到该文件夹下：



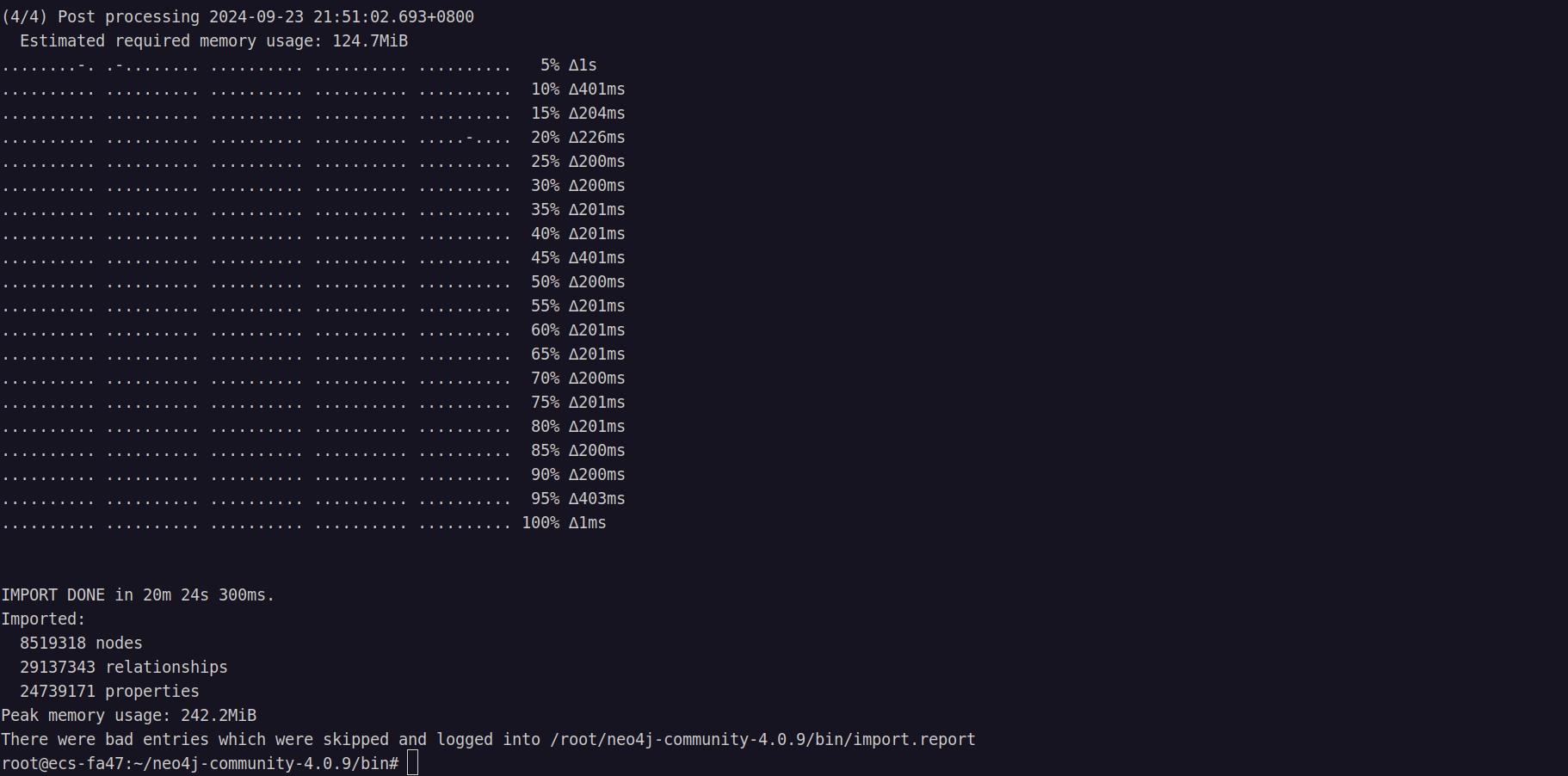
在使用之前，我们需要导入实验需要用到的数据。

1. 将data\_for\_neo4j文件夹下的数据全部拷贝到neo4j安装目录下的import文件夹下

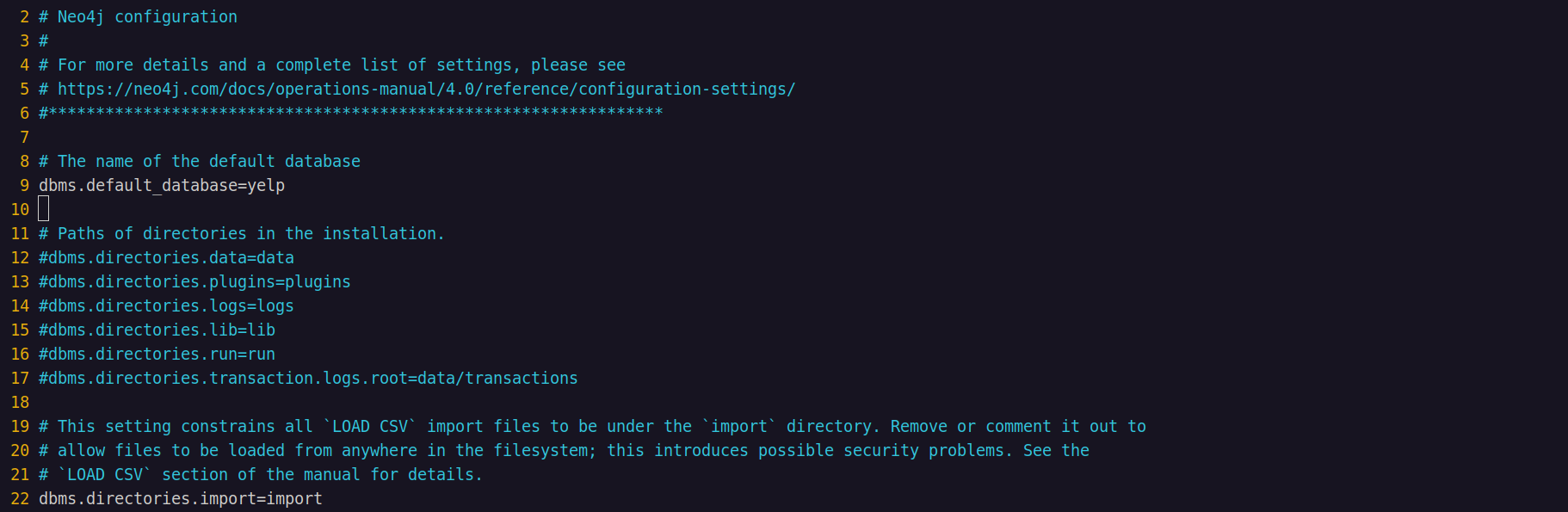


1. 进入neo4j的bin目录，执行以下命令：

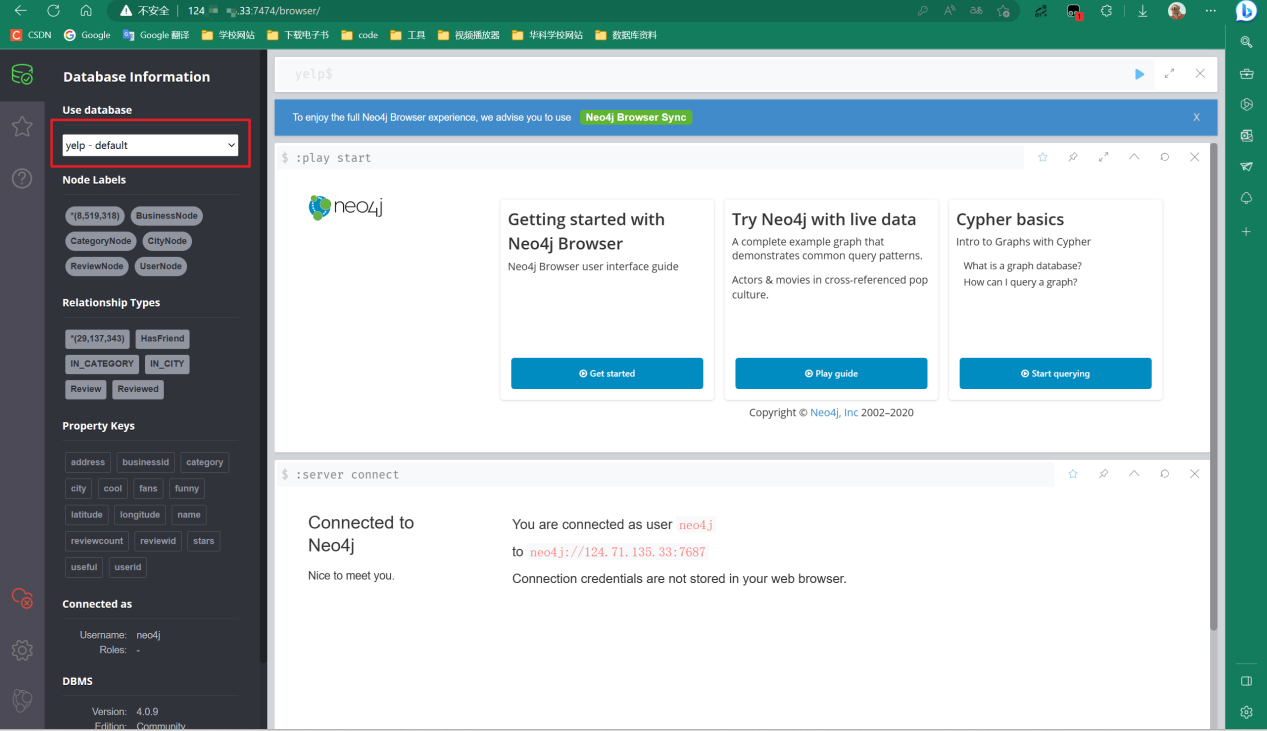
./neo4j-admin import --id-type=STRING --database=yelp --nodes=UserNode="../import/user\_header.csv,../import/user.csv" --nodes=ReviewNode="../import/review\_header.csv,../import/review.csv" --nodes=BusinessNode="../import/business\_header.csv,../import/BusinessNode.csv" --nodes=CityNode="../import/city\_header.csv,../import/CityNode.csv" --nodes=CategoryNode="../import/category\_header.csv,../import/CategoryNode.csv" --relationships=HasFriend="../import/user\_FRIENDS\_user\_header.csv,../import/user\_FRIENDS\_user.csv" --relationships=Review="../import/user\_WROTE\_review\_header.csv,../import/user\_WROTE\_review.csv" --relationships=Reviewed="../import/review\_REVIEWS\_business\_header.csv,../import/review\_REVIEWS\_business.csv" --relationships=IN\_CITY="../import/business\_IN\_CITY\_city\_header.csv,../import/business\_IN\_CITY\_city.csv" --relationships=IN\_CATEGORY="../import/business\_IN\_CATEGORY\_category\_header.csv,../import/business\_IN\_CATEGORY\_category.csv" --multiline-fields=true --skip-bad-relationships



数据导入成功后，修改neo4j.conf的默认database为yelp：（这里需要注意，Neo4j默认只能开启一个数据库，如果需要切换数据库，需要将这里的默认database修改，然后再重启登录即可）



执行./neo4j restart命令重启数据库，然后./neo4j console启动服务，在浏览器中重新登录数据库：

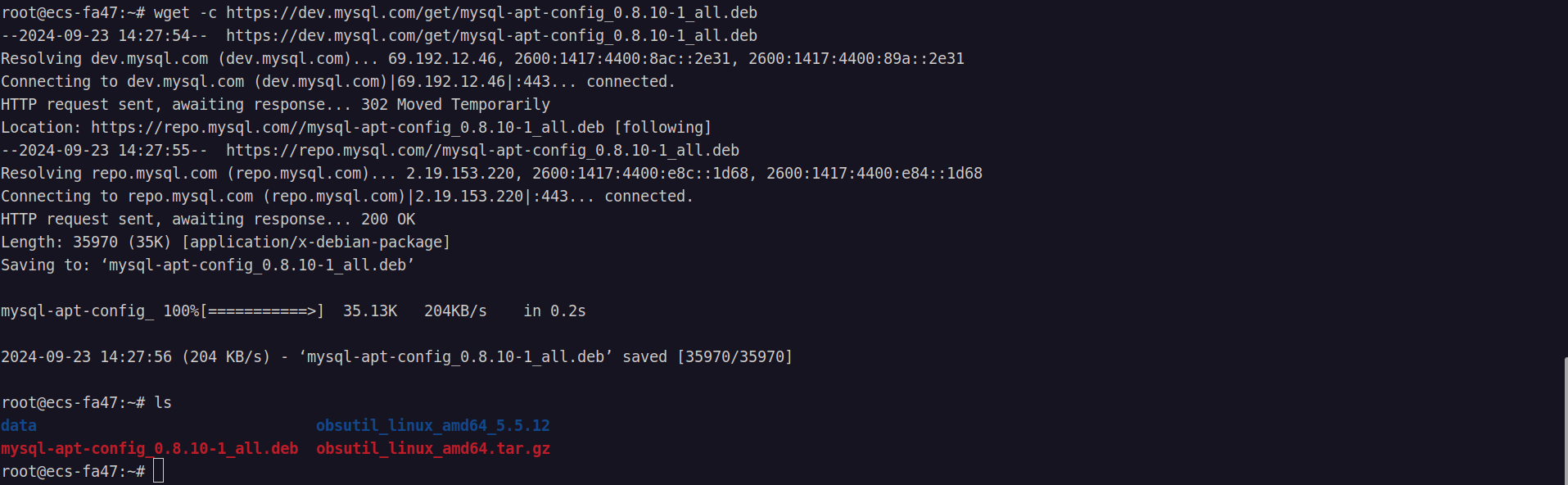


## 安装MySQL

使用以下命令即可进行mysql8.0版本安装：

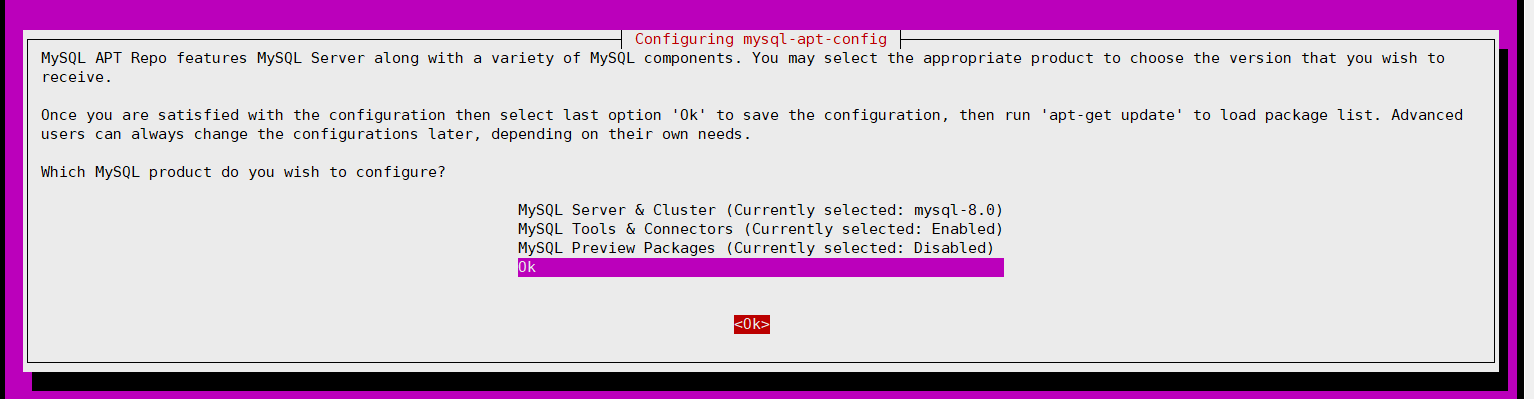
先下载.deb文件

wget -c <https://dev.mysql.com/get/mysql-apt-config_0.8.10-1_all.deb>



找到下载好的mysql-apt-config\_0.8.10-1\_all.deb文件

使用sudo dpkg -i mysql-apt-config\_0.8.10-1\_all.deb之后出现设置界面





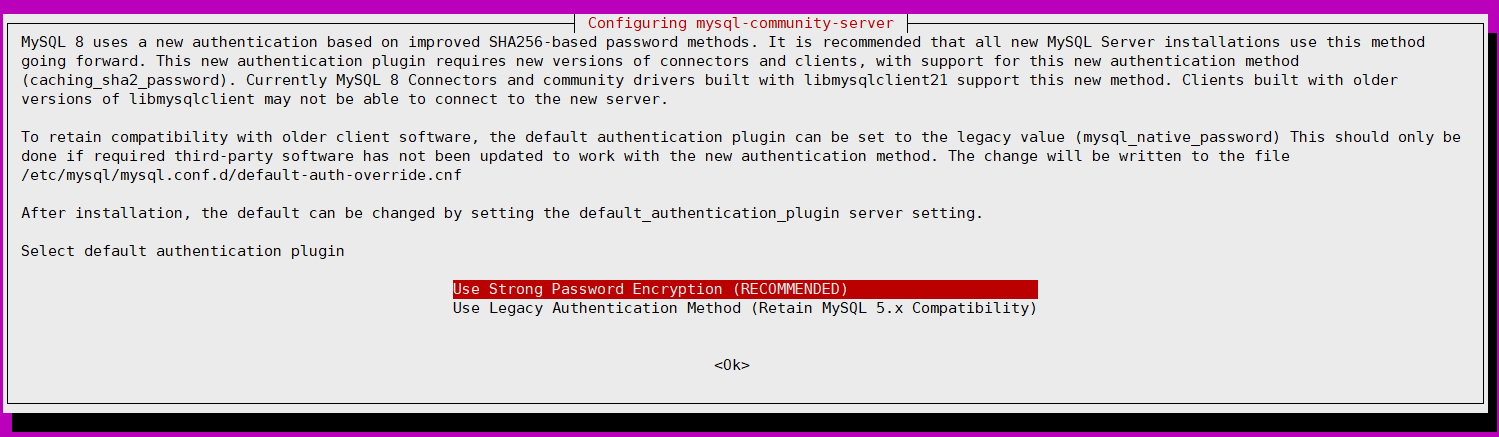
选择ok之后回车退出 ,再使用sudo apt-get update #更新软件源

sudo apt-get install mysql-server=8.0.\* #安装mysql8.0

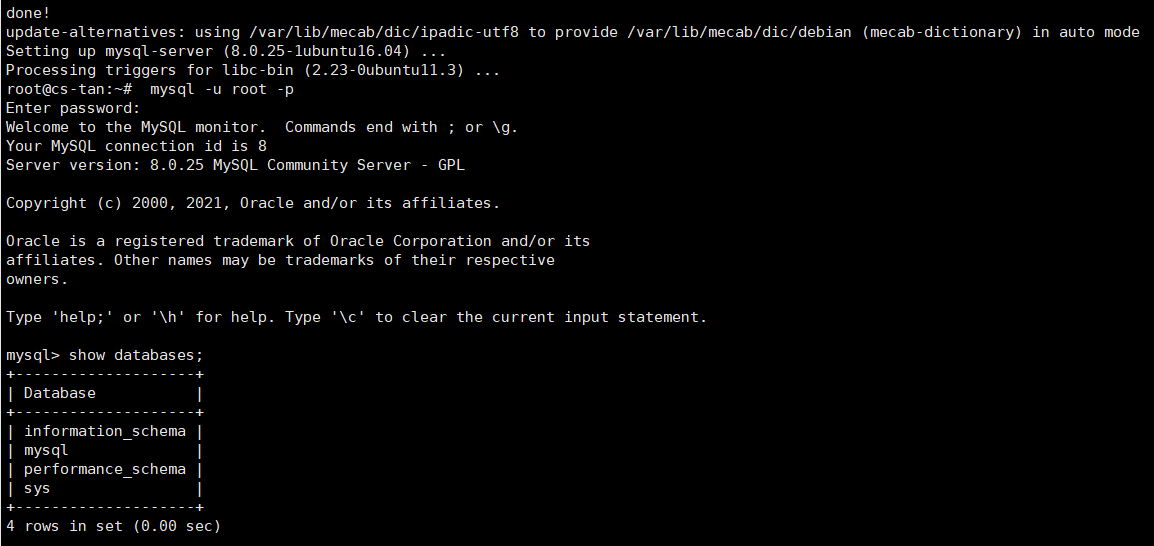
会出现两次选项都选择y 然后进入安装过程

安装过程中会出现设置MySQL密码界面和确认密码界面,输入自己的密码即可

然后出现以下界面,选择默认的推荐选项ok即可完成安装



使用mysql -u root -p 再键入密码即可进入mysql,安装完成.



如下命令是启动和关闭mysql服务器：

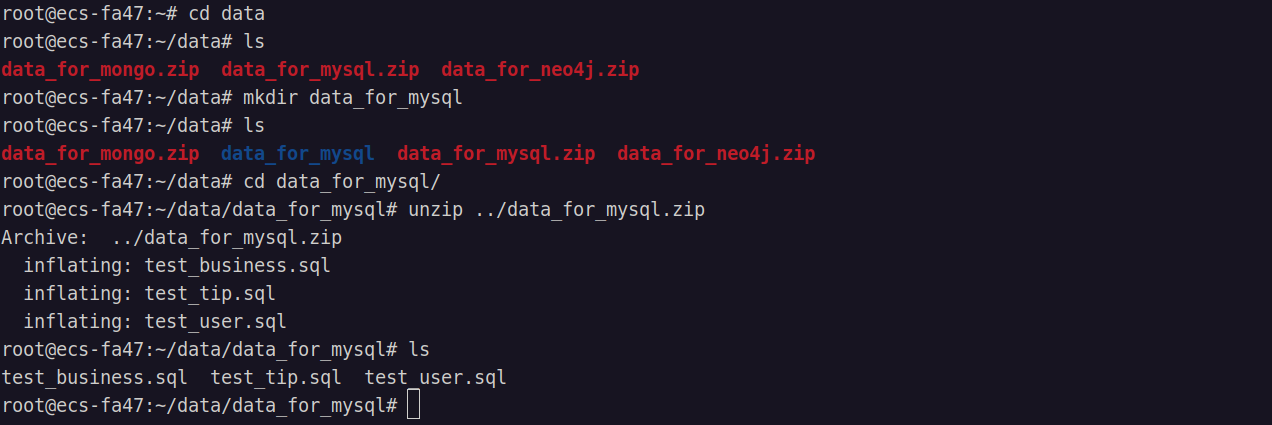
sudo service mysql start

sudo service mysql stop

启动服务器后使用mysql -u root -p命令，后跟密码即可使用mysql数据库。

**MySQL数据导入:**

将data\_for\_mysql.zip解压到data\_for\_mysql文件夹中



输入mysql -u root -p 然后键入你的密码进入mysql

使用 create database test; 和 use test; 切换到test数据库

依次输入: source /root/data/data\_for\_mysql/test\_business.sql

source /root/data/data\_for\_mysql/test\_user.sql

source /root/data/data\_for\_mysql/test\_tip.sql

完成三张表数据的导入,最后三张表中数据行数如下



## MongoDB创建分片集群

需要在华为云购置三台服务器，可在不同区域。以我的三台服务器为例，三台服务器的弹性公网IP地址如下：

server1：123.60.129.44

server2：124.71.195.208

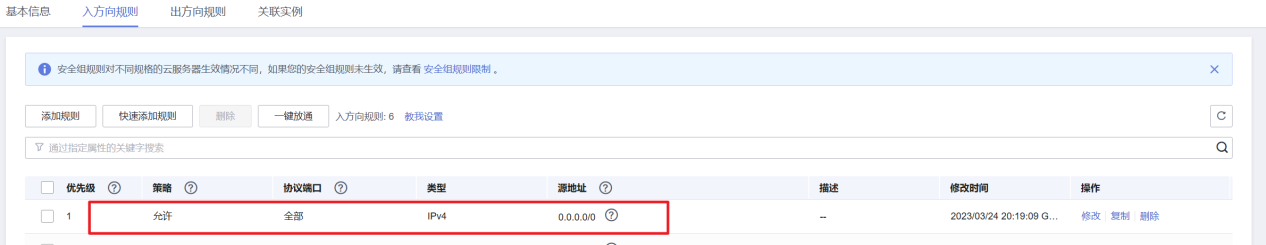
server3：124.71.199.181

(下面涉及这三个ip的地方请替换成你自己的ip)

三台服务器上都应该安装MonGoDB，安装步骤参见上文。

### 准备

在华为云为三台服务器都添加入方向规则和出方向规则，否则后面会出现解析IP错误，添加方法同本手册第三部分第3部分。之后可在本机ping服务器判断是否添加规则成功。



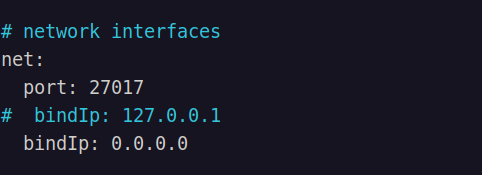


如图所示，则添加规则成功。

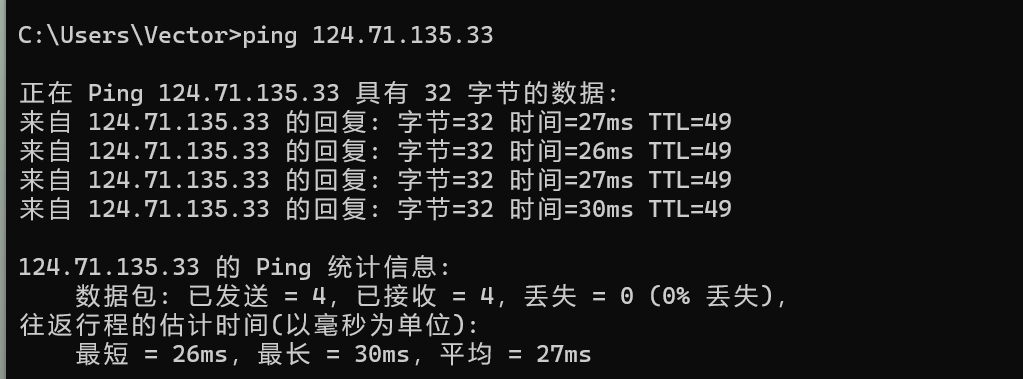
在三台服务的安全组中添加端口为27017和27018的入方向规则：



修改三个服务器的/etc/mongod.conf文件，将其bindIp修改为0.0.0.0这样可以允许除了本机之外的远程连接。（注意缩进，bindIp的缩进与port对其，不然会配置错误）（另外需要注意，在进行下面的配置之前，一定要关闭mongod服务，执行sudo systemctl stop mongod，否则可能导致配置不成功）



ping命令检测服务器状态：



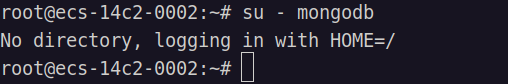
三台服务器创建数据日志存放目录：

(提示:下面的su – mongodb执行失败的话先执行su root就行)

Server1:

su - mongodb

正常应当如下所示：



mkdir /usr/local/mongodb

cd /usr/local/mongodb

mkdir -p data/shard11

mkdir -p data/shard21

mkdir -p data/config

touch data/shard11.log

touch data/shard21.log

Server2:

su - mongodb

mkdir /usr/local/mongodb

cd /usr/local/mongodb

mkdir -p data/shard12

mkdir -p data/shard22

mkdir -p data/config

touch data/shard12.log

touch data/shard22.log

Server3:

su - mongodb

mkdir /usr/local/mongodb

cd /usr/local/mongodb

mkdir -p data/shard13

mkdir -p data/shard23

mkdir -p data/config

touch data/shard13.log

touch data/shard23.log

### 配置Shard1的replica sets

Sever1:

mongod --shardsvr --replSet shard1 --port 27017 --dbpath /usr/local/mongodb/data/shard11 --oplogSize 2048 --logpath /usr/local/mongodb/data/shard11.log --logappend --bind\_ip=0.0.0.0 --fork

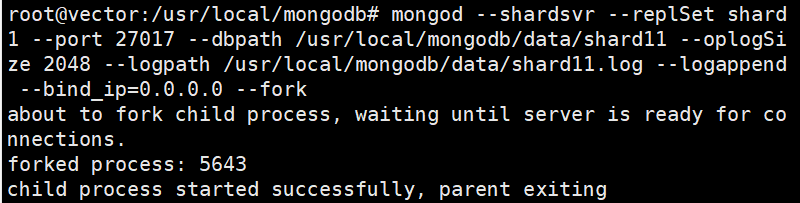
Server2:

mongod --shardsvr --replSet shard1 --port 27017 --dbpath /usr/local/mongodb/data/shard12 --oplogSize 2048 --logpath /usr/local/mongodb/data/shard12.log --logappend --bind\_ip=0.0.0.0 --fork

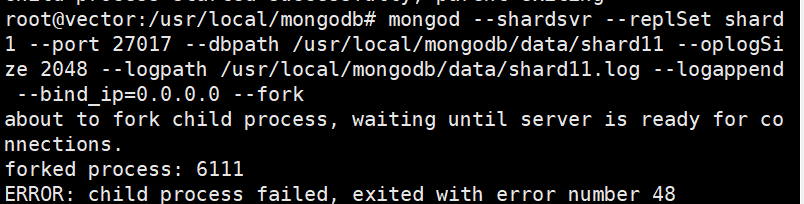
Server3:

mongod --shardsvr --replSet shard1 --port 27017 --dbpath /usr/local/mongodb/data/shard13 --oplogSize 2048 --logpath /usr/local/mongodb/data/shard13.log --logappend --bind\_ip=0.0.0.0 --fork

Server1执行示例如下：



之后如果遇到这种出错：



需要关闭mongod进程：

查看mongo相关进程：ps aux | grep mongo

关闭mongodb全部进程：killall mongod

如果还是报错，尝试退出终端命令行并重启服务器，再重新使用该命令。

### 初始化replica set

可在本机，也可在服务器用mongo连接其中一个mongod，执行:

（这里的IP替换为自己的ip）

mongo 123.60.129.44:27017

config = {

\_id: 'shard1',

members: [

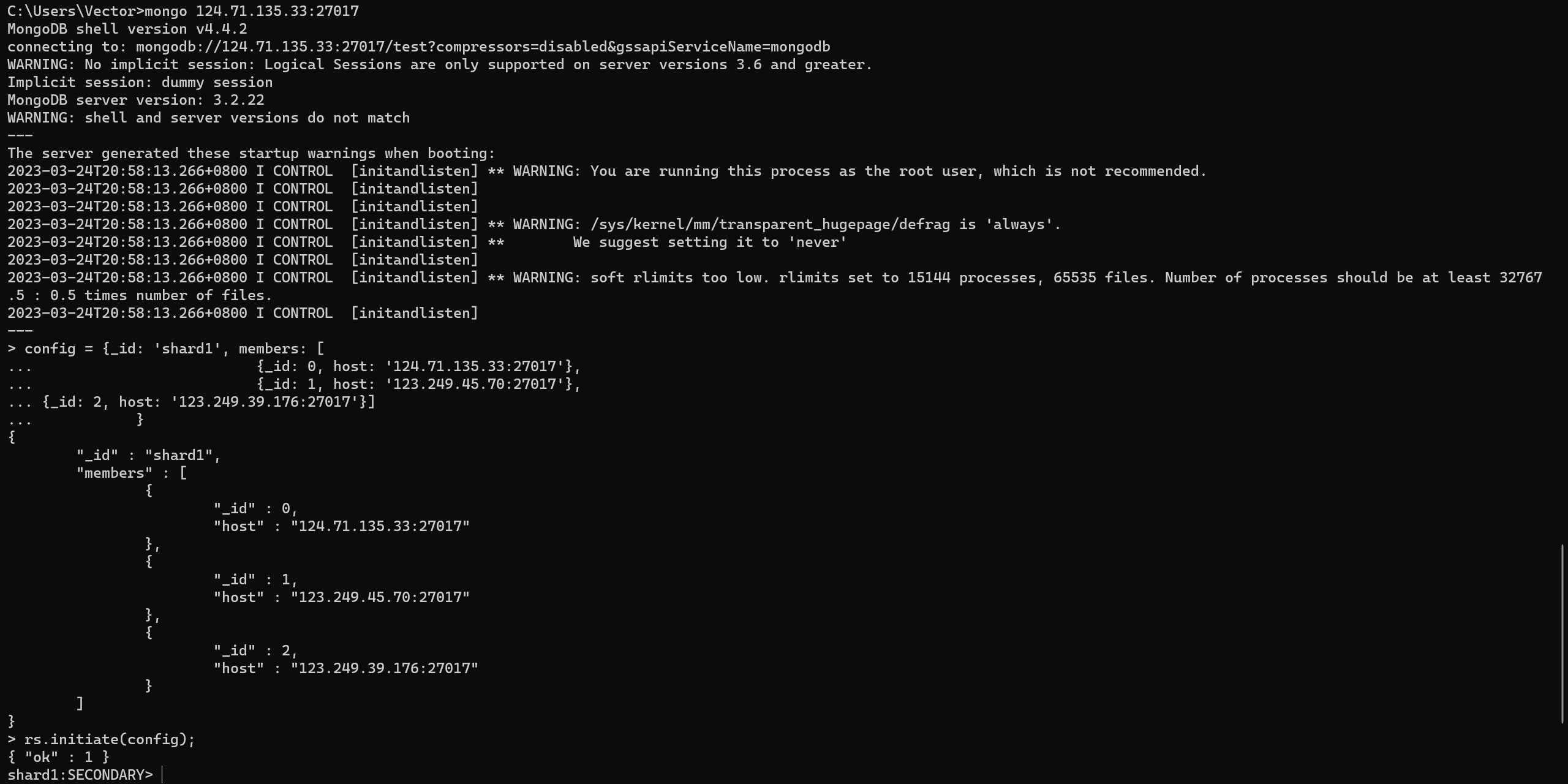
{\_id: 0, host: '123.60.129.44:27017'},

{\_id: 1, host: '124.71.195.208:27017'},

{\_id: 2, host: '124.71.199.181:27017'}]

}

rs.initiate(config);



### 配置shard2的replica sets

Server1:

mongod --shardsvr --replSet shard2 --port 27018 --dbpath /usr/local/mongodb/data/shard21 --oplogSize 2048 --logpath /usr/local/mongodb/data/shard21.log --logappend --bind\_ip=0.0.0.0 --fork

Server2:

mongod --shardsvr --replSet shard2 --port 27018 --dbpath /usr/local/mongodb/data/shard22 --oplogSize 2048 --logpath /usr/local/mongodb/data/shard22.log --logappend --bind\_ip=0.0.0.0 --fork

Server3:

mongod --shardsvr --replSet shard2 --port 27018 --dbpath /usr/local/mongodb/data/shard23 --oplogSize 2048 --logpath /usr/local/mongodb/data/shard23.log --logappend --bind\_ip=0.0.0.0 --fork

### 初始化replica set

可在本机，也可在服务器用mongo连接其中一个mongod，执行:

（这里的IP替换成自己的IP）

mongo 123.249.45.70:27018

config = {

\_id: 'shard2',

members: [

{\_id: 0, host: '123.60.129.44:27018'},

{\_id: 1, host: '124.71.195.208:27018'},

{\_id: 2, host: '124.71.199.181:27018'}]

}

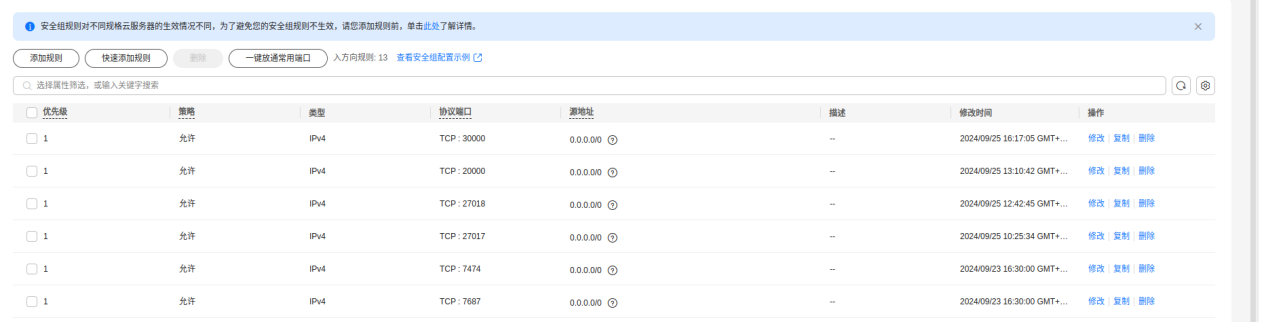
rs.initiate(config);

### 配置config server

Server1、2、3分别执行:

mongod --configsvr --replSet config --dbpath /usr/local/mongodb/data/config --port 20000 --logpath /usr/local/mongodb/data/config.log --logappend --bind\_ip=0.0.0.0 --fork

然后在三个服务器的安全组规则的入方向中都添加端口为20000的规则和30000的规则，配置规则如下：



在本地计算机用mongo连接其中一个mongod，执行:

（这里的ip替换为自己的ip）

mongo 123.60.129.44:20000

config = {

\_id: 'config',

members: [

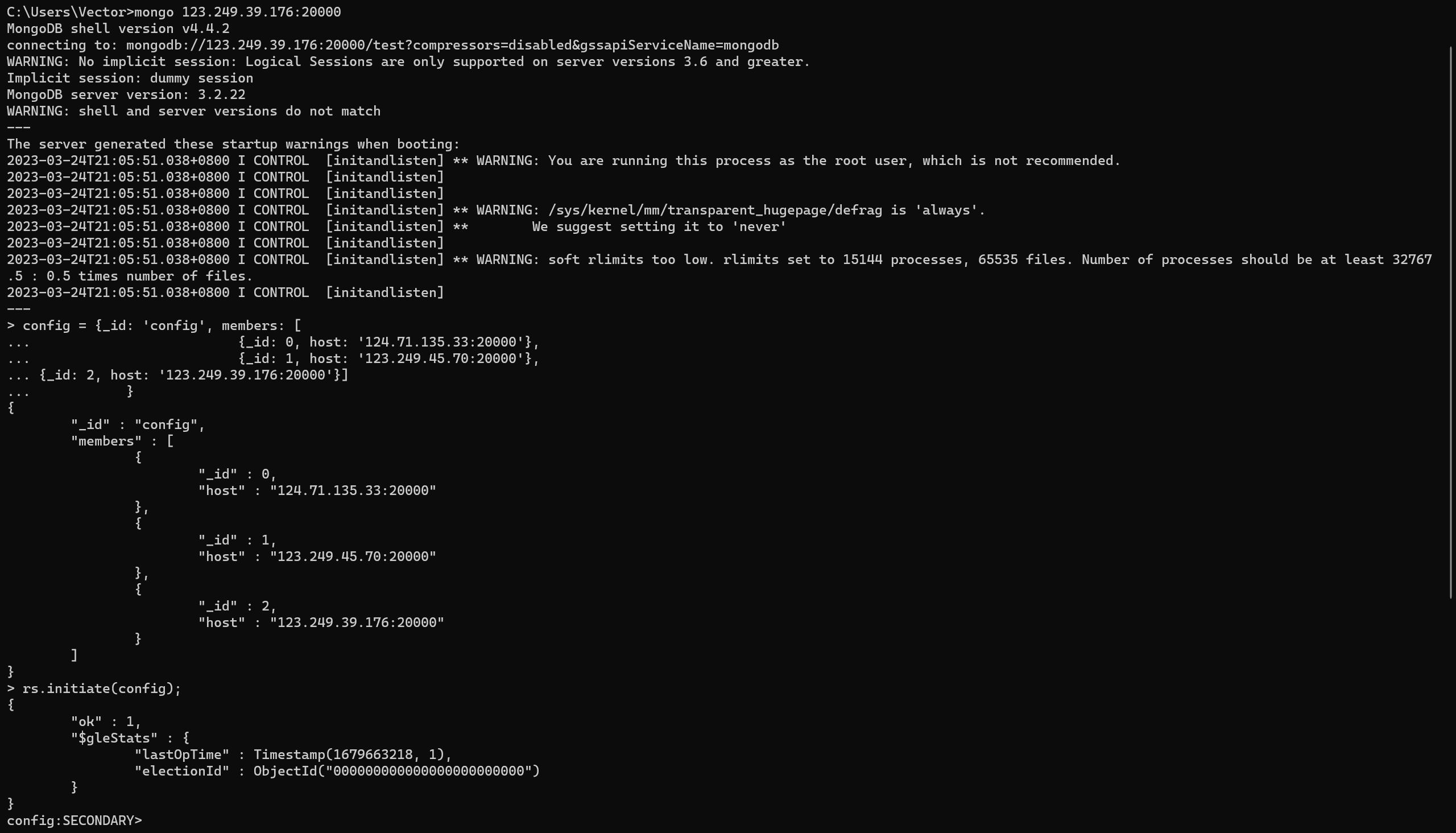
{\_id: 0, host: '123.60.129.44:20000'},

{\_id: 1, host: '124.71.195.208:20000'},

{\_id: 2, host: '124.71.199.181:20000'}]

}

rs.initiate(config);



### 配置mongs

Server1、2、3分别执行:

（这里的IP替换为自己的ip）

mongos --configdb config/123.60.129.44:20000,124.71.195.208:20000,124.71.199.181:20000 --port 30000 --logpath /usr/local/mongodb/data/mongos.log --logappend --bind\_ip=0.0.0.0 --fork

### 使用mongos

(1)首先由要以shard1实例启动mongod，见本章第2小节。

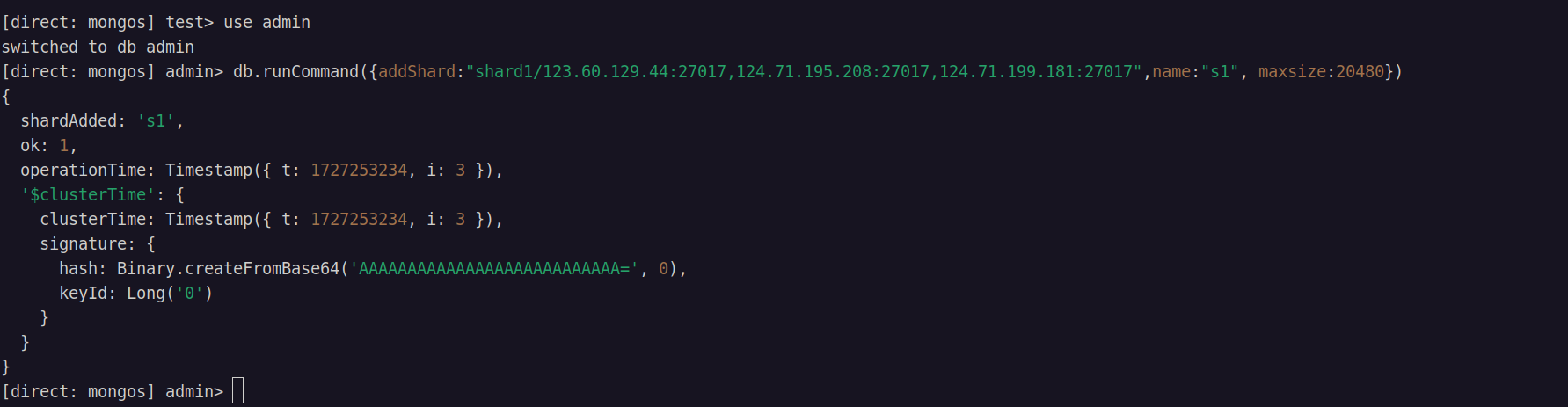
(2)可在本机，也可在服务器用mongo连接其中一个mongod，执行:

mongo 123.60.129.44:30000

切换到admin，添加分片：

use admin;

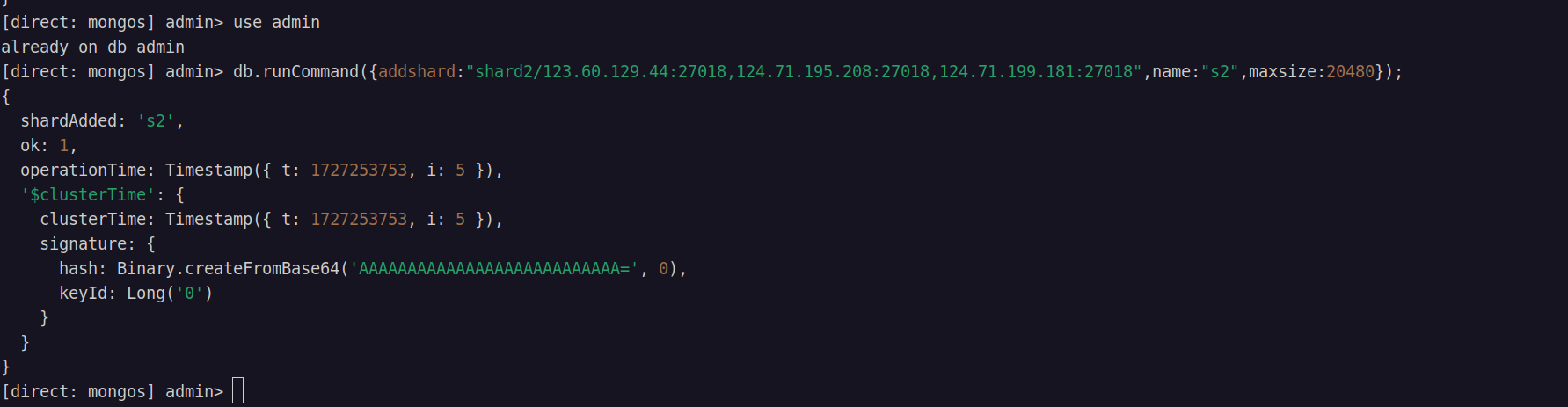
db.runCommand({addshard:"shard1/123.60.129.44:27017,124.71.195.208:27017,124.71.199.181:27017",name:"s2",maxsize:20480});



1. 以shard2实例启动mongod，见本章第4小节。

use admin

db.runCommand({addshard:"shard2/123.60.129.44:27018,124.71.195.208:27018,124.71.199.181:27018",name:"s2",maxsize:20480});



### 激活数据库分片

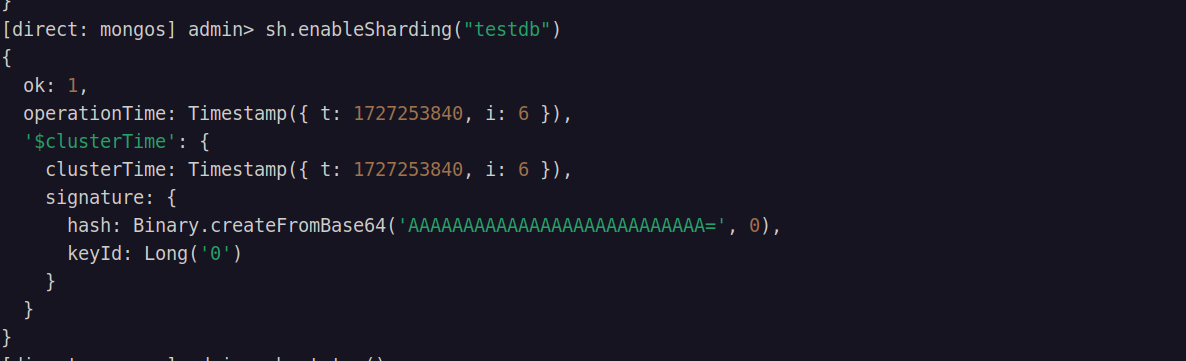
创建数据库，激活数据库分片，如创建testdb数据库，则使用以下命令激活分片：

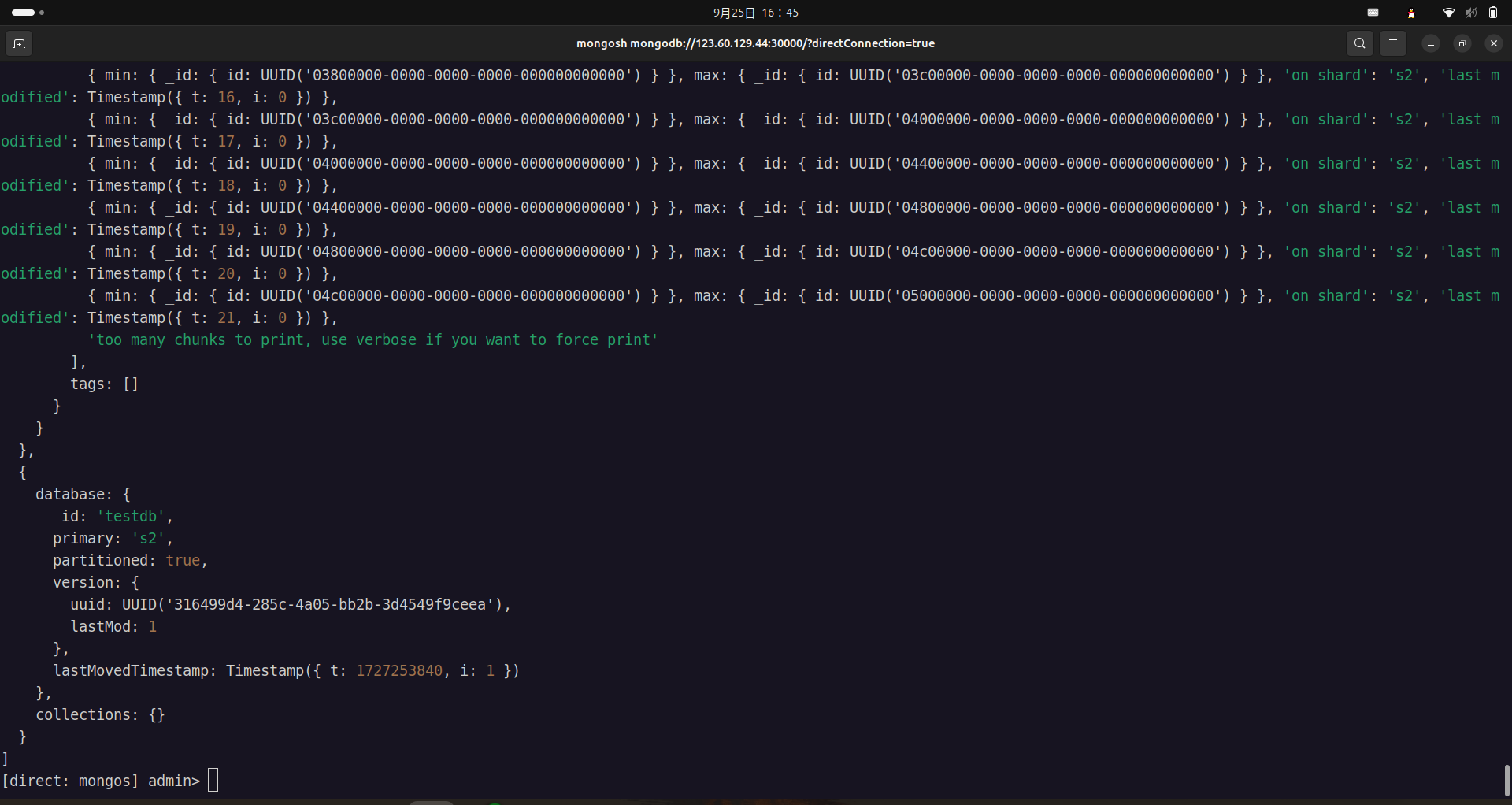
sh.enableSharding("testdb")

使用sh.status()查看数据库当前情况，其中

primary：数据库主分片位置

partitioned：false表示分片未开启/true表示分片已开启。





切换到testdb：

use testdb

创建一个集合并指定分片键。分片键必须是索引的字段：

db.createCollection("myCollection")

sh.shardCollection("testdb.myCollection", { myShardKey: 0})

//////// myShardKey为n表示小于n的key分到shard1，大于等于n的key分到shard2

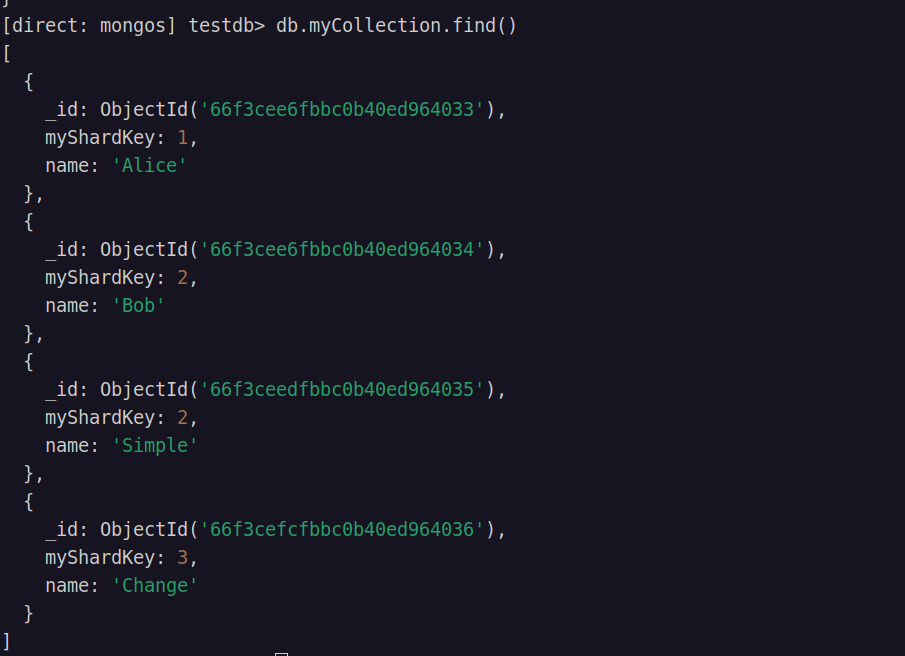
向testdb中插入几行数据：

db.myCollection.insert({ myShardKey: 1, name: "Alice" })

//可以多插入几行



查询所有数据如下所示：



单个分片登录方法：本地打开终端，使用执行以下命令：

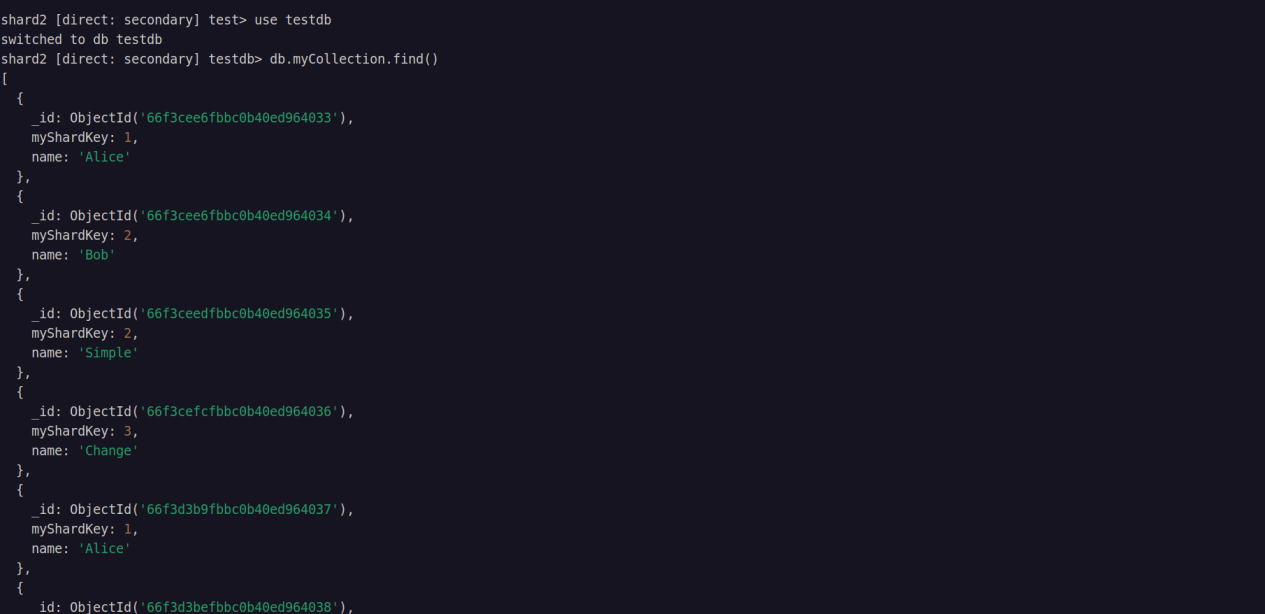
mongo 123.60.129.44：27017

该端口表示是shard1，查询数据如下：



mongo 123.60.129.44：27018

该端口表示是shard2，查询数据如下：



重新启动分片集群服务的流程如下：

1. 启动三台服务器
2. 本地ssh连接到三台服务器
3. 按照本章第6小节启动服务器
4. 按照本章第7小节打开端口30000

（若遇到执行错误，错误代码为48，表明端口暂时被占用，第6小节的命令和第7小节的命令可以隔一段时间再执行，或者在执行第6小节的命令之后，使用lsof -i:30000查看占用30000端口的进程，并使用kill <PID>的方式杀死该进程，然后执行第7小节的命令即可）

1. 本地启动mongo服务，使用mongo 123.60.129.44:30000连接到端口30000，这时所连接的是集群服务器。
2. 如果要启动分片来查看，需要打开对应的端口，参见本章第2小节和第4小节（shard1对应的端口为27017，shard2对应的端口为27018）。比如可以从IP地址为123.60.129.44的服务器打开27017和27018端口，这时候在本地新的命令行终端执行分别执行以下命令：

mongo 123.60.129.44:27017 // 连接到shard1

mongo 123.60.129.44:27018 // 连接到shard2