网络设置：

vi /etc/sysconfig/network-scripts/ifcfg-ens33



重启网络：

service network restart

ping [www.baidu.com](http://www.baidu.com)

2.解决yum命令

大概意思是fastestmirror不能使用，fastestmirror是yum的一个加速插件

处理办法就是禁用这个插件

方法两种

第一种

vi /etc/yum/pluginconf.d/fastestmirror.conf

把enabled=1改为0

vi /etc/yum.conf

把plugins=1改为0

3. yum -y install net-tools

yum -y install wget

yum -y install lrzsz

安装jdk

资料地址：<https://blog.csdn.net/weixin_42266606/article/details/80863781>

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

cd /usr/local

mkdir java

tar -zxvf jdk-8u211-linux-x64.tar.gz

cd jdk1.8.0\_211/

pwd

/usr/local/java/jdk1.8.0\_211

vim /etc/profile在最后面增加：

JAVA\_HOME=/usr/local/java/jdk1.8.0\_211

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

PATH=$JAVA\_HOME/bin:$PATH

export JAVA\_HOME CLASSPATH PATH

resource /etc/profile

测试：

java –version

## 考虑引入saltstack

## Linux如果最小安装的话，先把系统工具安装上：

yum -y install gcc automake autoconf libtool make net-tools openssh-clients wget openssh-server binutils glibc-kernheaders glibc-common glibc-devel

## redis安装：

### 安装依赖工具

wget http://downloads.sourceforge.net/tcl/tcl8.6.1-src.tar.gz

tar -xzvf tcl8.6.1-src.tar.gz

cd  /usr/local/tcl8.6.1/unix/

./configure

make && make install

### 安装redis

cd /usr/local

mkdir redis

cd redis/

wget <http://download.redis.io/releases/redis-5.0.5.tar.gz>

tar -zxvf redis-5.0.5.tar.gz

rm -rf redis-5.0.5.tar.gz

cd redis-5.0.5/

make && make test && make install

启动redis:

cd /usr/local/redis/redis-5.0.5/src/

./redis-server /usr/local/redis/redis-5.0.5/redis.conf

测试是否启动成功：

./redis-cli -h 127.0.0.1 -p 6379

ping 若出现PONG 则启动成功

配置开机启动

1. 将redis utils目录下的，redis\_init\_script脚本拷贝到linux的/etc/init.d目录中

cd /usr/local/redis/redis-5.0.5/utils/

cp ./redis\_init\_script /etc/init.d/

1. 将redis\_init\_script重命名为redis\_6379，6379是我们希望这个redis实例监听的端口号

cd /etc/init.d/

mv /etc/init.d/redis\_init\_script /etc/init.d/redis\_6379

修改redis\_6379脚本的第6行的REDISPORT，设置为相同的端口号（默认就是6379）

cd /etc/init.d/

vim redis\_6379

1. 创建两个目录：/etc/redis（存放redis的配置文件），/var/redis/6379（存放redis的持久化文件）

cd /etc/

mkdir /etc/redis/

mkdir /var/redis/

mkdir /var/redis/6379

1. redis配置文件（默认在根目录下，redis.conf），拷贝到/etc/redis目录中，修改名称为6379.conf

cd /usr/local/redis/redis-5.0.5/

cp ./redis.conf /etc/redis/

cd /etc/redis/

mv redis.conf 6379.conf

5.修改启动项

进入/etc/init.d

vim ./redis\_6379

改成自己的安装路径

EXEC=/usr/local/redis/redis-5.0.5/src/redis-server

CLIEXEC=/usr/local/redis/redis-5.0.5/src/redis-cli

6.设置权限

chmod 777 redis\_6379

7．启动redis

./redis\_6379 start

./redis\_6379 stop

8.测试是否启动成功

./redis-cli -h 127.0.0.1 -p 6379

ping 若出现PONG 则启动成功

9.配置开机启动

vim /etc/init.d/redis\_6379

在最上面加上下面语句

# chkconfig: 2345 90 10

# description: Redis is a persistent key-value database

chkconfig redis\_6379 on

### Redis常规配置

常规配置：

修改6379.conf中的部分配置为生产环境

# bind 127.0.0.1 绑定ip

protected-mode no 设置保护模式

daemonize yes 让redis以daemon进程运行

pidfile /var/run/redis\_6379.pid 设置redis的pid文件位置

port 6379 设置redis的监听端口号

dir /var/redis/6379 设置持久化文件的存储位置

redis持久化配置：

持久化有两种方式:rdb 和aof

rdb:

redis默认开启

可配置

save 900 1

save 300 10

save 60 10000

/var/redis/6379/dump.rdb

Aof:

appendonly yes 开启aof持久化

appendfsync 1.no 2. always 3.everysec

生产一般默认 使用everysec

日志文件名

appendfilename "appendonly.aof"

AOF rewrite设置：

auto-aof-rewrite-percentage 100

auto-aof-rewrite-min-size 64mb

**数据备份方案 ：**

（1）写crontab定时调度脚本去做数据备份

（2）每小时都copy一份rdb的备份，到一个目录中去，仅仅保留最近48小时的备份

（3）每天都保留一份当日的rdb的备份，到一个目录中去，仅仅保留最近1个月的份

（4）每次copy备份的时候，都把太旧的备份给删了

（5）每天晚上将当前服务器上所有的数据备份，发送一份到远程的云服务上去

1.判断linux系统是否安装crontab服务

rpm -qa | grep crontab

2.在目录/usr/local/redis创建文件件copy，并进出入copy目录，创建两个文件

redis\_rdb\_copy\_hourly.sh、redis\_rdb\_copy\_daily.sh

mkdir daybackup

mkdir hourbackup

mkdir copy

cd copy

touch redis\_rdb\_copy\_hourly.sh

touch redis\_rdb\_copy\_daily.sh

vim redis\_rdb\_copy\_hourly.sh

把以下内容复制进去

#!/bin/sh

cur\_date=`date +%Y%m%d%k`

echo ‘开始删除’

rm -rf /usr/local/redis/hourbackup/$cur\_date

mkdir /usr/local/redis/hourbackup/$cur\_date

cp /var/redis/6379/dump.rdb /usr/local/redis/ hourbackup /$cur\_date

del\_date=`date -d -48hour +%Y%m%d%k`

rm -rf /usr/local/redis/hourbackup/$del\_date

vim redis\_rdb\_copy\_daily.sh

把以下绿色部分内容复制进去

#!/bin/sh

cur\_date=`date +%Y%m%d`

rm -rf /usr/local/redis/daybackup/$cur\_date

mkdir /usr/local/redis/daybackup/$cur\_date

cp /var/redis/6379/dump.rdb /usr/local/redis/daybackup/$cur\_date

del\_date=`date -d -1month +%Y%m%d`

rm -rf /usr/local/redis/daybackup/$del\_date

3. crontab -e 进入定时任务编辑

0 \* \* \* \* sh /usr/local/redis/copy/redis\_rdb\_copy\_hourly.sh

0 0 \* \* \* sh /usr/local/redis/copy/redis\_rdb\_copy\_daily.sh

**数据恢复方案**

1. 如果是redis进程挂了 ，直接重启redis应用即可，直接恢复aof日志中数据，最多丢失1秒数据
2. 如果是redis进程所在机器挂掉，那么重启机器后，尝试重启redis进程，尝试直接基于AOF日志文件进行，前提是AOF没有破损，AOF append-only，顺序写入，如果AOF文件破损，那么用redis-check-aof fix修复。
3. 如果redis当前最新的AOF和RDB文件出现了丢失/损坏，那么可以尝试基于该机器上当前的某个最新的RDB数据副本进行数据恢复，/usr/local/redis/snapshotting

参考资料：<https://blog.csdn.net/qq_29461259/article/details/84059387>