# Linux

## 什么是Linux

* 一种免费，开源的操作系统
* 常作为服务器的操作系统使用
* 在目前市面上的服务器操作系统一般使用linux,极少数使用Windows Server(微软公司为服务器提供的应用操作系统,包含更加全面的权限管理系统以及防火墙机制).

## 常用发行版本

* RedHat(红帽子)
* CentOS
* Ubuntu(有一套完整的界面解决方案,可以作为个人用户使用的操作系统)

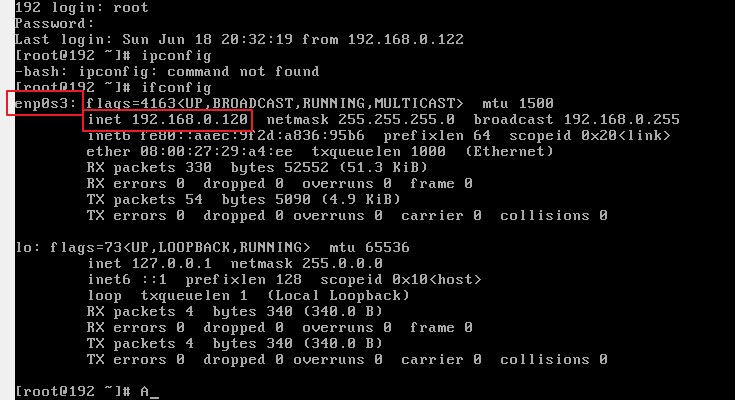
## 常用命令

### 退出命令行

ctrl+c

### 查看IP

ifconfig : 查看当前电脑IP地址及网卡信息



### 查看当前目录文件

* 显示当前目录下的文件以及文件信息
  + ll
* 显示当前目录文件名称
  + ls
* 进入目录
* cd 目录名
* 回到上层目录
* cd ..
* 显示当前的目录的绝对路径
* pwd
* 命令历史

通过↑/↓查看命令历史

### 编辑文件

* vim : 编辑文件命令 vim text.txt
  + i: 进入编辑模式
  + esc按键 : 退出编辑状态
  + :q : 退出vim编辑器
  + :q! : 强制退出vim,不保存修改内容
  + :w : 保存但停留在vim界面
  + :wq : 保存并退出文件

### 操作目录

* 创建目录
  + mkdir 文件夹名称
* 删除目录/文件
  + rm [-r] [-f] 文件/目录
* 拷贝目录/文件
  + cp [-r] 源文件/源目录 目标文件/目标目录
* 移动目录/文件
  + mv 源文件/源目录 目标文件/目标目录

### 查找文件

* 查找文件内容
  + grep 需要查找的字符串 文件名
* 查找文件
  + find / -name \*文件名\*

### 压缩包及解压包

* .tar
  + 解包：tar xvf FileName.tar
  + 打包：tar cvf FileName.tar DirName
* .gz
  + 解包：gzip -d FileName.gz
  + 打包：gzip FileName
* .tar.gz
  + 解包：tar zxvf FileName.tar.gz
  + 打包：tar zcvf FileName.tar.gz DirName
* .zip
  + 解包：unzip FileName.zip
  + 打包：zip FileName.zip FileName

### 开放服务器端口

Linux默认只开放了80和22端口.其他端口需要用户自行授权开放

* + 对外开放端口(临时)：
    - /sbin/iptables -I INPUT -p tcp --dport 8080 -j ACCEPT
  + 端口加入防火墙白名单(永久)
    - firewall-cmd --permanent --zone=public --add-port=8080/tcp
    - firewall-cmd --reload

### 用户添加

* 添加用户
  + useradd 用户账户名
* 修改用户密码
  + passwd 用户账户名
* 切换用户
  + su [用户名]
    - 当用户名不填时,表示切换到系统用户root
* 给用户添加权限
  + chmod u+w 指定的文件（/etc/sudoers）
  + 添加完权限之后,需要使用sudo 来执行命令

## 软件安装

### 安装JDK

使用ftp工具将jdk的rpm安装包上传至某一个目录中.

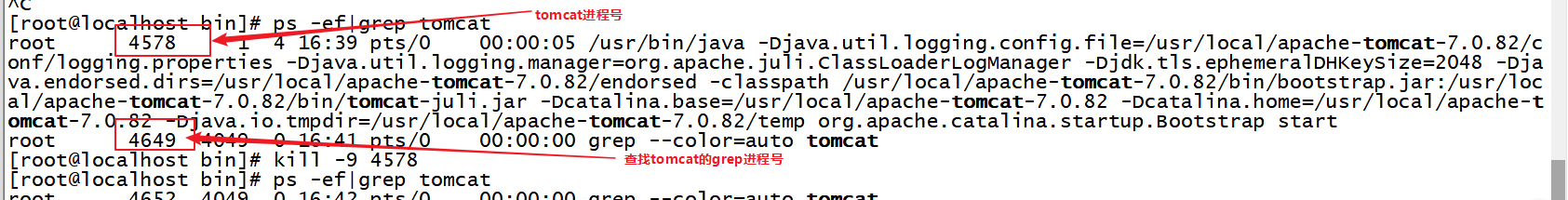
安装命令:

rpm -ivh jdk的rpm安装包

验证安装:

Java -version

### 安装tomcat

* 解压tomcat的压缩包
  + tar zxvf apache-tomcat-7.0.82.tar.gz
* 将解压缩的文件移动到/usr/local目录
  + mv apache-tomcat-7.0.82 /usr/local/
* 启动tomcat
  + **./**startup.sh
* **输出tomcat日志**
  + tail -f ../logs/catalina.out
* 停止tomcat
  + ./shutdown.sh
* 查看tomcat进程
  + ps -ef|grep tomcat
* 强制终止tomcat
  + 
  + Kill -9 端口号
* 开放tomcat端口
  + 对外开放端口(临时)：
    - /sbin/iptables -I INPUT -p tcp --dport 8080 -j ACCEPT
  + 端口加入防火墙白名单(永久)
    - firewall-cmd --permanent --zone=public --add-port=8080/tcp
    - firewall-cmd –reload

### redis安装

* 解压redis的压缩包
  + tar zxvf redis-3.2.9.tar.gz
* 将redis的解压缩文件夹移到/usr/local
  + mv redis-3.2.9 /usr/local/
* cd到redis的源码目录
  + cd /usr/local/redis-3.2.9/src
* 编译redis源码
  + make
  + make test
* 修改redis.conf
  + vim redis.conf
  + /xxx 查找搜索文档 n下一个
  + bind 绑定端口号注释
  + requirepass 打开注释 修改密码
* 端口号加入白名单
  + /sbin/iptables -I INPUT -p tcp --dport 6379 -j ACCEPT
* 启动redis服务
  + ./redis-server ../redis.conf & (后台运行)
* 启动redis客户端
  + ./redis-cli -h 127.0.0.1 -p 6379
  + auth 密码
* 停止redis服务
  + Kill -9 进程号

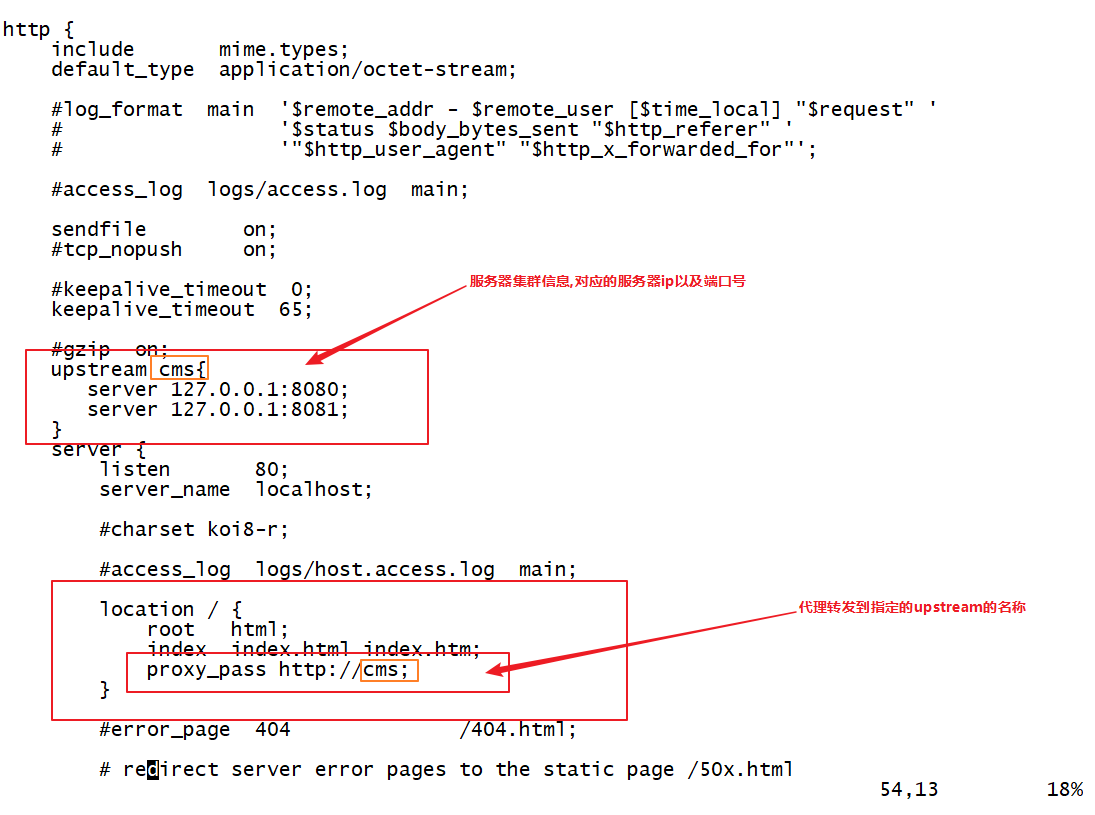
### 安装mysql

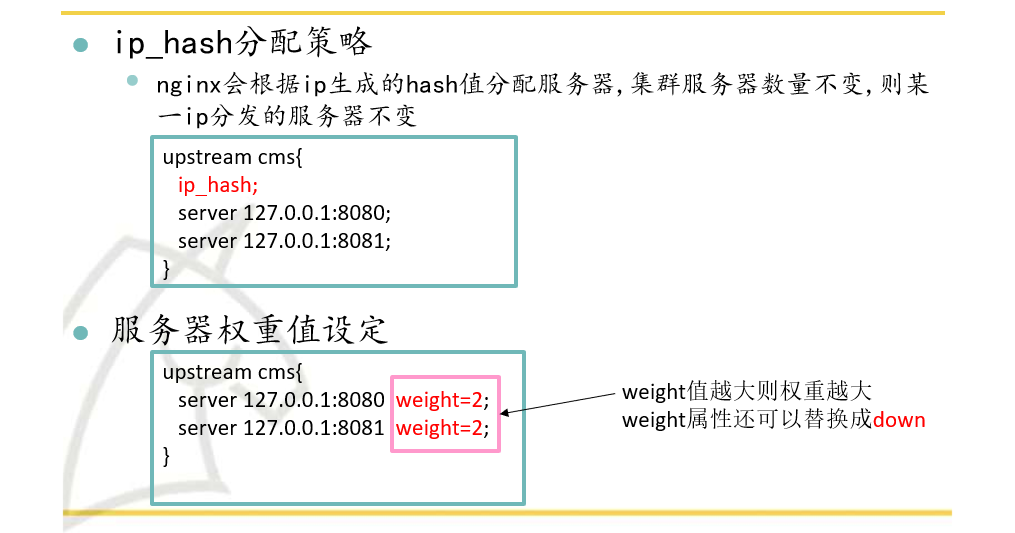
* 创建mysql文件夹
  + mkdir mysql
* 解压mysql的压缩包
  + tar xvf mysql-5.7.20-1.el7.x86\_64.rpm-bundle.tar -C mysql/
    - -C 表示解压到指定的目录
* 删除Mysql的MariaDB依赖
  + rpm -qa|grep mariadb [查找mariadb]
  + rpm -e mariadb包全名 –nodeps
* 依次安装mysql的rpm包(common,libs,client,server)
  + rpm -ivh mysql-community-common-5.7.20-1.el7.x86\_64.rpm
  + rpm -ivh mysql-community-libs-5.7.20-1.el7.x86\_64.rpm
  + rpm -ivh mysql-community-client-5.7.20-1.el7.x86\_64.rpm
  + rpm -ivh mysql-community-server-5.7.20-1.el7.x86\_64.rpm
* 启动mysql
  + service mysqld start
* 查找初始密码
  + vim /var/log/mysqld.log 或 grep password /var/log/mysqld.log
* 连接mysql
  + mysql -u root -p
* 修改mysql访问权限
  + use mysql; [进入mysql数据库]
  + update user set host='%' where user = 'root'; [更改用户的访问权限]
  + flush privileges; [刷新权限缓存]
* 3306加入白名单
  + /sbin/iptables -I INPUT -p tcp --dport 3306 -j ACCEPT

### 安装Nginx

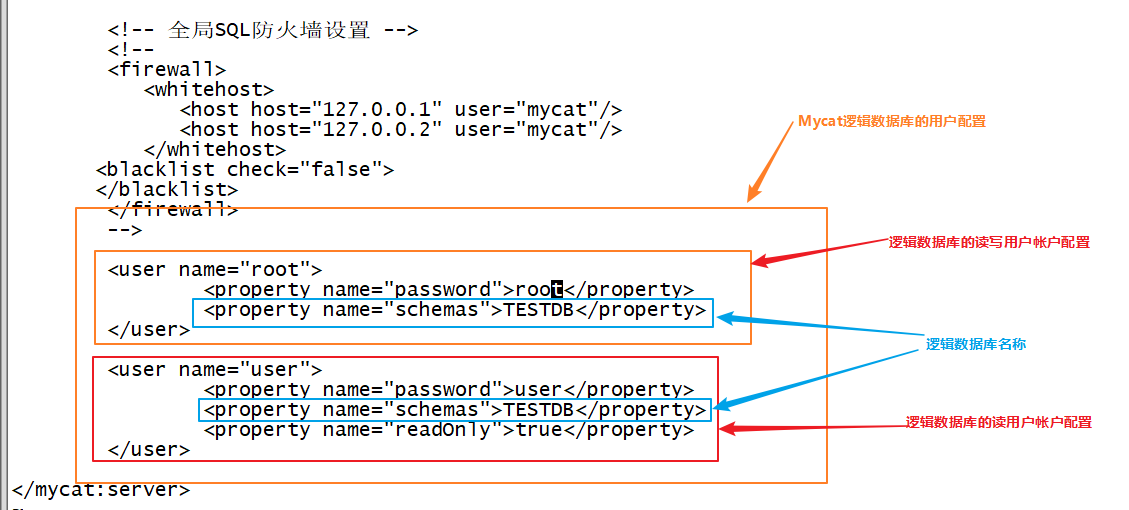
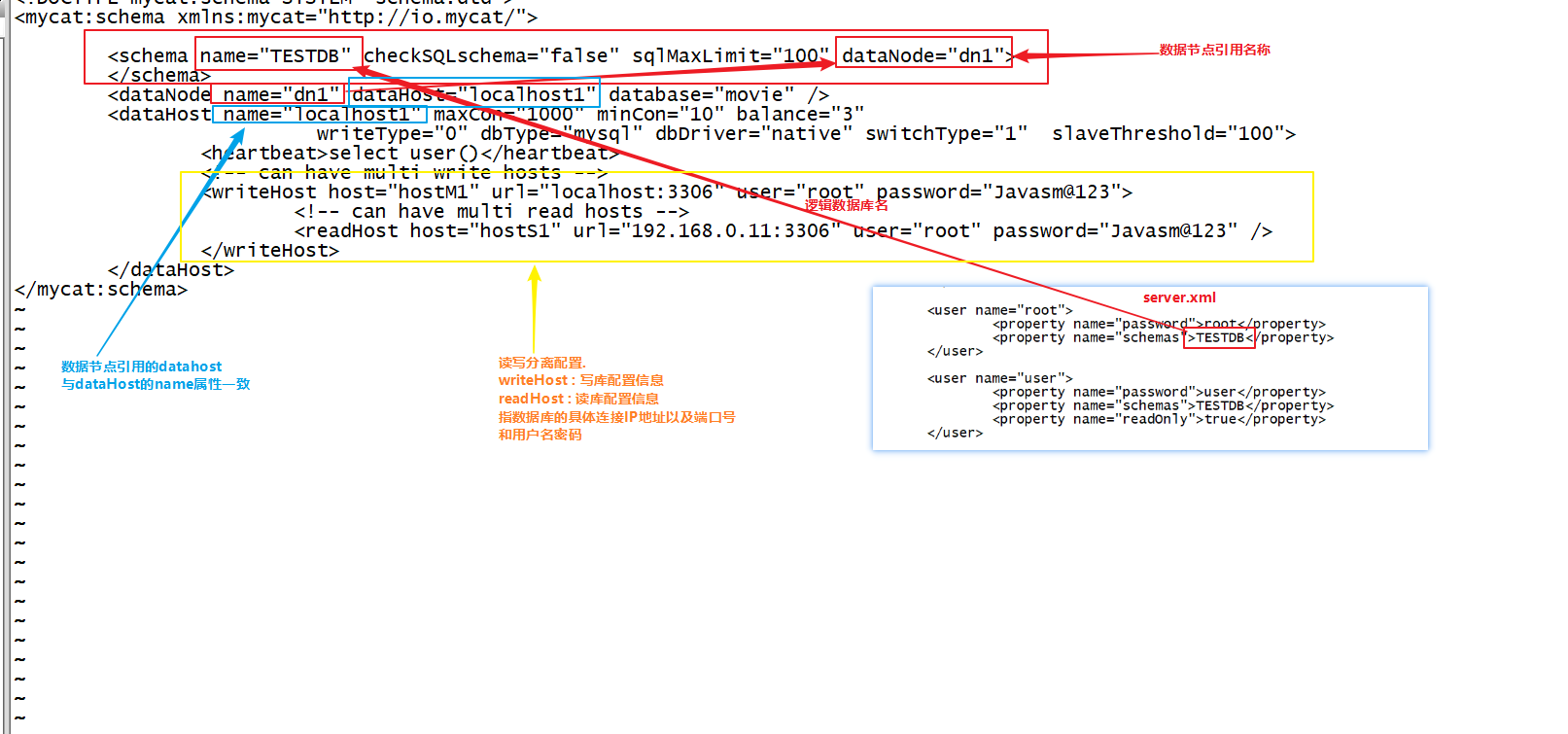
* 安装nginx相关的依赖
  + yum -y install zlib zlib-devel openssl openssl-devel
  + yum -y install patch
* 解压Nginx压缩包
  + tar zxvf nginx-1.10.3.tar.gz
* 创建nginx相关的目录
  + mkdir /usr/local/nginx/
* 将nginx中的文件移至/usr/local/nginx
  + mv nginx-1.10.3/\* /usr/local/nginx/
* 配置nginx编译环境

**./configure --prefix=/usr/local/nginx --pid-path=/usr/local/nginx/nginx.pid --error-log-path=/usr/local/nginx/logs/error.log --http-log-path=/usr/local/nginx/logs/access.log --with-http\_stub\_status\_module --with-http\_ssl\_module --with-http\_gzip\_static\_module --with-http\_realip\_module --with-stream --http-client-body-temp-path=/usr/local/nginx/tmp/client/ --http-proxy-temp-path=/usr/local/nginx/tmp/proxy/**

* 编译安装
  + make
  + make install
* 配置nginx.conf
  + vim /usr/local/nginx/conf/nginx.conf
  + 添加upstream 服务器组
  + 
* 校验nginx配置文件
  + /usr/local/nginx/sbin/nginx –t
* 启动nginx
  + /usr/local/nginx/sbin/nginx -c /usr/local/nginx/conf/nginx.conf
* 重启
  + /usr/local/nginx/sbin/nginx -s reload
* 其他配置



### 安装Mycat

* 从定义和分类来看，它是一个开源的分布式数据库系统，是一个实现了 MySQL 协议的Server，前端用户可以把它看作是一个数据库代理，用 MySQL客户端和命令行访问，MySQL 原生（Native）协议与多个 MySQL 服务器通信，也可以用 JDBC 协议与大多数主流数据库服务器通信，其核心功能是读写分离，分表分库，即将一个大表水平分割为 N 个小表，存储在后端 MySQL 服务器里或者其他数据库里。
* 解压Mycat安装包
  + tar zxvf Mycat-server-1.6-RELEASE-20161028204710-linux.tar.gz
* 将文件夹移至/usr/local/
  + mv mycat /usr/local/
* 修改server.xml
  + 进入mycat/conf目录
  + vim server.xml 修改user相关信息(也可以使用默认信息)
    - 
  + 修改schema.xml
    - schema配置文件是用来描述逻辑数据库中的数据节点和读写库配置信息
    - 
    - 测试Mycat
  + Mycat 运行命令
    - * + Mycat start 启动
        + Mycat stop 停止
        + Mycat restart 重新启动
  + 查看Mycat运行日志
    - * + tail –f mycat.log
  + 调整日志等级
    - * + 修改log4j2.xml的日志等级为debug
    - 连接命令
  + 使用mysql命令登入Mycat
  + mysql -u root -p -P8066 -h 127.0.0.1
  + mycat默认端口号为8066

### 安装solr

#### solr简介

* Solr 是Apache下的一个顶级开源项目，采用Java开发，它是基于Lucene的全文搜索服务器。Solr提供了比Lucene更为丰富的查询语言，同时实现了可配置、可扩展，并对索引、搜索性能进行了优化。
* Solr可以独立运行，运行在Jetty、Tomcat等这些Servlet容器中，Solr 索引的实现方法很简单，用 POST 方法向 Solr 服务器发送一个描述 Field 及其内容的 XML 文档，Solr根据xml文档添加、删除、更新索引 。Solr 搜索只需要发送 HTTP GET 请求，然后对 Solr 返回Xml、json等格式的查询结果进行解析，组织页面布局。Solr不提供构建UI的功能，Solr提供了一个管理界面，通过管理界面可以查询Solr的配置和运行情况

#### 安装solr

* 解压solr
  + - tar zxvf solr-5.5.5.tgz
* 创建Solr数据和安装目录
  + - mkdir -p /data/solr /usr/local/solr
* 安装Solr
  + - 进入 solr-5.5.5/bin/ 目录
    - ./install\_solr\_service.sh solr的tgz安装包路径 -d /data/solr -i /usr/local/solr
* Solr安装时,会自动创建一个solr用户,可以进入/etc/passwd中查看
  + - cat /etc/passwd
    - 如未创建用户则进行手工创建
    - groupadd solr
    - useradd -g solr solr
* 给Solr用户授权,防止创建core无权限的情况
  + - chown -R solr.solr /data/solr /usr/local/solr
* 查看Solr状态
  + - service solr status
* Solr命令
  + - 启动 - service solr start
    - 停止 – service solr stop
    - 重新启动 – service solr restart
  + 创建一个新的core
    - su - solr -c "/usr/local/solr/solr/bin/solr create -c newcore -n data\_driven\_schema\_configs"
    - core相当于一个文档集,存放着文档,文档字段类型配置,索引等等信息
  + 访问Solr管理界面
    - http://localhost:8983/solr

#### 主从同步

主库负责写入数据,从库负责读数据,并让从库同步主库的数据,在Mysql中,官方提供Master/Slave同步方案.

* 配置Master
  1. 编辑/etc/my.cnf
  2. 添加服务id配置
     + server-id=1 (保证唯一)
  3. 开启二进制日志文件
     + log-bin=/var/lib/mysql/mysql-bin
  4. 重启mysql
  5. 查看master状态信息
     + show master status; (File和Position需要给slave使用)
* 配置Slave
  1. 编辑/etc/my.cnf
  2. 添加服务id配置
     + server-id=2 (保证唯一)
  3. 重启数据库,登入mysql
  4. 停止slave
     + stop slave;
  5. 运行master命令

**change master to master\_host='192.168.0.2',**

**master\_port=3306, master\_user='root',**

**master\_password='root',**

**master\_log\_file='File值',**

**master\_log\_pos=** **Position值;**

* 1. 启动slave
     + start slave
  2. 查看slave状态
     + show slave status\G;
* 测试主从同步
  1. 在主库中插入一条数据/在mycat逻辑数据库中的表插入数据
  2. 查看从库表中数据是否已同步完成/在mycat逻辑数据库中的表查询数据

## yum

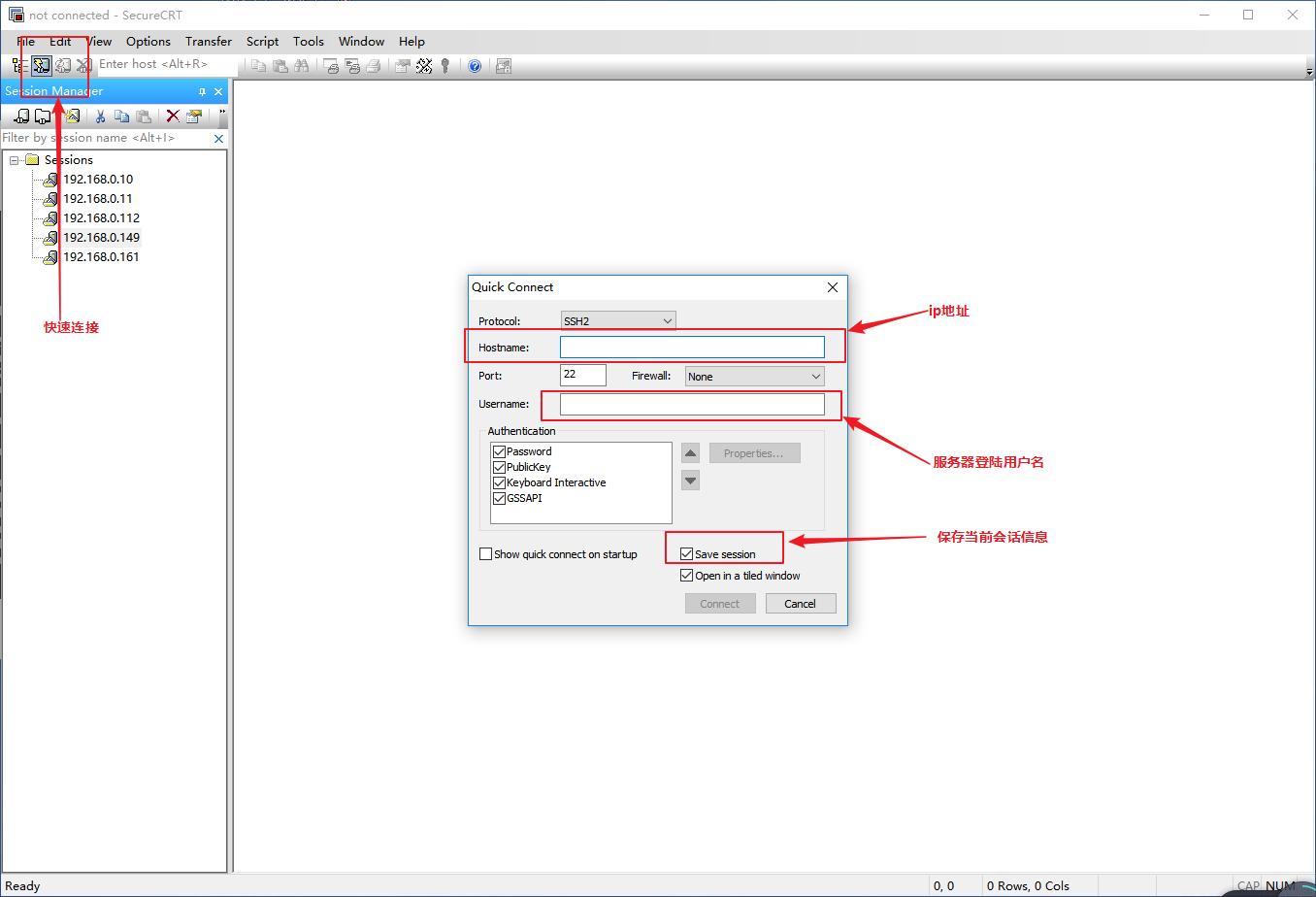
* yum（全称为 Yellow dog Updater, Modified）是一个在Fedora和RedHat以及CentOS中的Shell前端软件包管理器。基于RPM包管理，能够从指定的服务器自动下载RPM包并且安装，可以自动处理依赖性关系，并且一次安装所有依赖的软件包，无须繁琐地一次次下载、安装。

### yum 常用命令

* 查找包
  + yum search 指令名
* 安装包
  + yum install 搜索出来的包名
* 卸载包
  + yum remove 包名
* 默认yes安装
  + yum -y install 包名

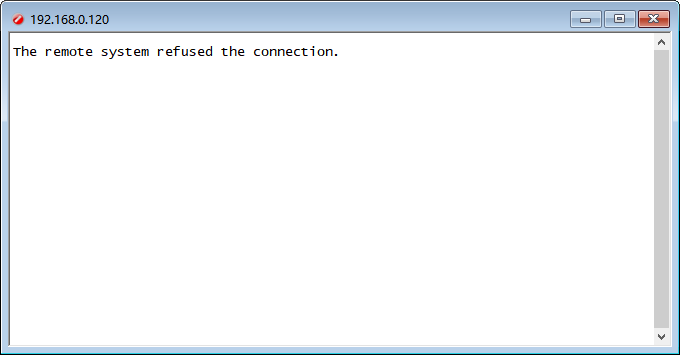
## Linux远程连接工具SCRT

SCRT是一款远程连接工具,可使用它来连接远程服务器. 并可在当前软件中操作系统中的文件等其他命令



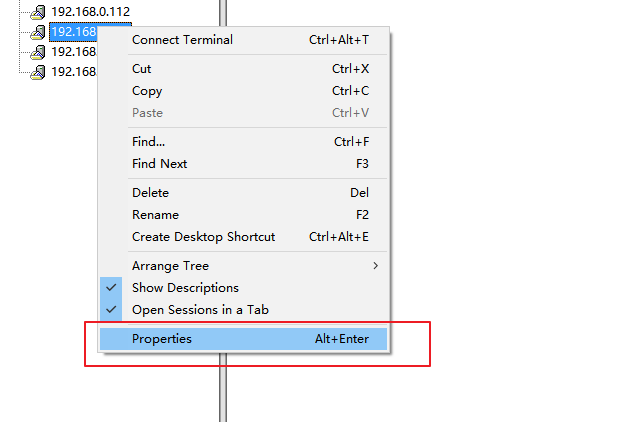
### Scrt常见问题

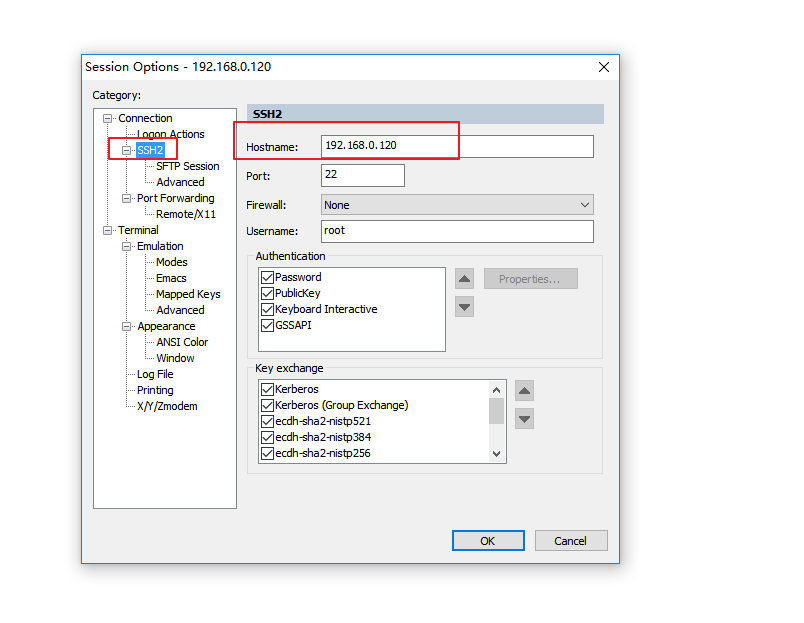
#### 连接不上服务器



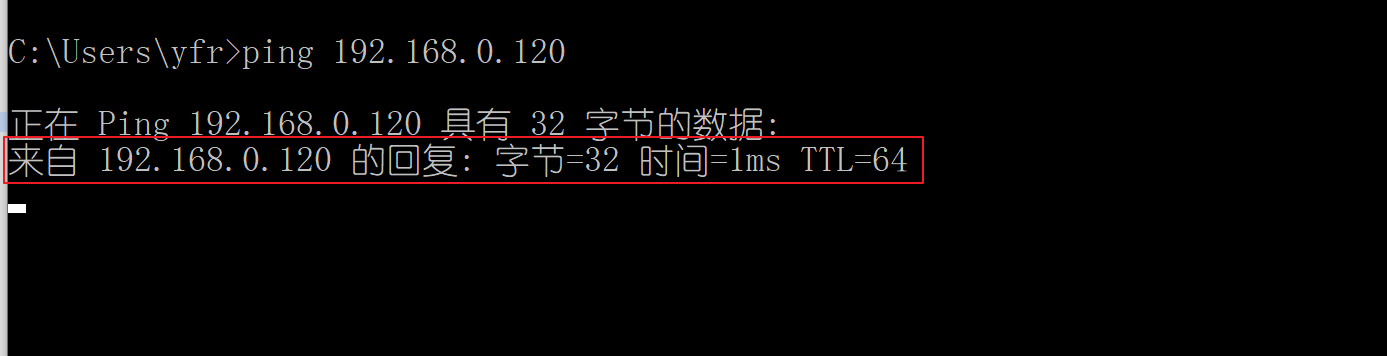
* 检查远程连接的服务器的IP地址

修改当前会话的IP地址





* 保证远程服务器的IP地址能够ping

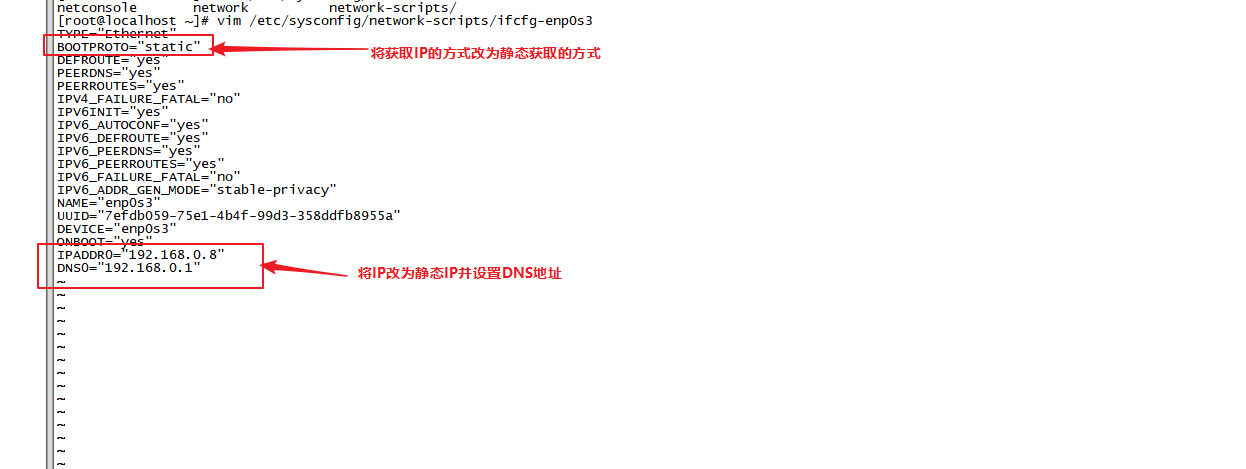


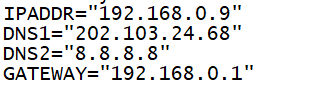
* IP重复一般是比较常见的问题
  + 使用ifconfig查找到当前默认网卡



* + 修改网卡配置信息

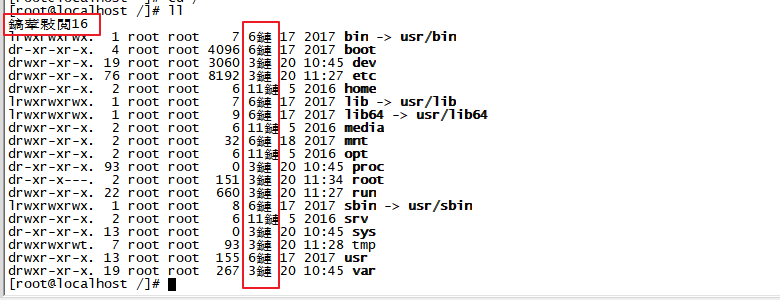
vim /etc/sysconfig/network-scripts/ifcfg-enp0s3



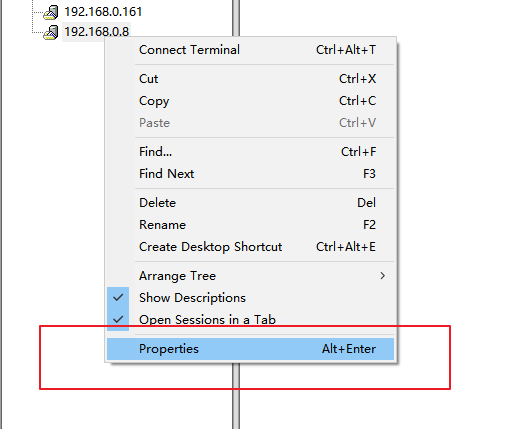
* + 连接外网
    - route add default gw 192.168.0.1
    - 
  + 重启网卡

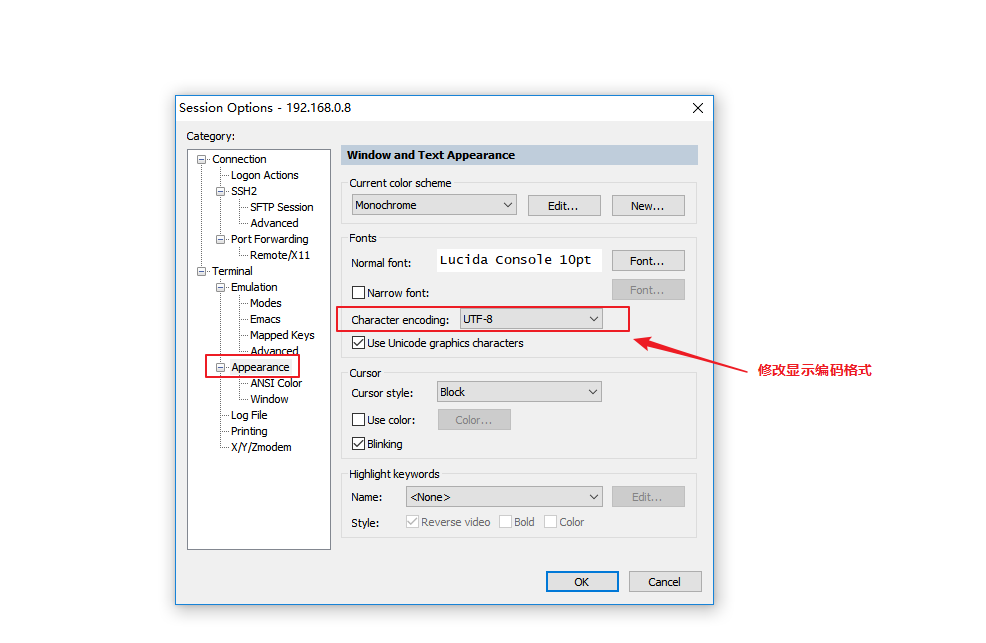
Service network restart

#### 连接窗口出现乱码



* 关闭当前的会话信息
* 选中会话配置属性



* 

Linux文件上传工具FlashFXP

