

| | |
|--|----|
| 1、安装中文输入法..... | 3 |
| 2、配置网关等 linux 和 CentOS 下网卡启动、配置等 ifcfg-eth0 教程..... | 5 |
| 3、Linux 设置 SSH 开机启动..... | 8 |
| 4、Linux 安装 JDK 7..... | 8 |
| 5、Linux JDK 7 环境变量配置..... | 9 |
| 6、secureCRT 上传下载使用..... | 11 |
| 7、解压文件..... | 12 |
| 8、Linux 安装 Mysql..... | 12 |
| 9、Linux 安装 zookeeper..... | 17 |
| 10、Linux 安装 dubbo 服务维护..... | 19 |
| 11、Linux 中 nginx 的维护..... | 25 |
| 12、Linux 中监听具体端口..... | 26 |
| 13、Linux 中 redis 安装共享 session..... | 26 |
| 14、Linux 下使用 yum 安装 mysql..... | 28 |
| 15、Linux 中 tomcat 部署 war 包..... | 31 |
| 16、Linux 中 mysql 插入问题记录..... | 33 |
| 17、Linux 服务器被肉鸡如何解决?..... | 33 |
| 18、Ubuntu install rar..... | 35 |
| 19、Ubuntu install mysql timer..... | 35 |
| 20、Ubuntu add user..... | 36 |
| 21、connect mysql..... | 37 |
| 22、查看系统版本,清除缓存,删除大文件及 du 使用..... | 37 |
| 23、文件夹授权..... | 38 |
| 24、解决 python-pip install error..... | 39 |
| 25、linux 新建分区, 格式化硬盘, 持久生效..... | 39 |
| 26、Ubuntu 开启防火墙..... | 40 |
| 27、sudo: parse error in /etc/sudoers near line 24 ...报错..... | 41 |
| 28、How to safely abort apt-get install?..... | 41 |
| 29、How to add interface in ubuntu?..... | 41 |
| 30、Global search file in linux..... | 42 |
| 31、gitlab 服务器文件移动时, 修改 git url..... | 42 |
| 32、idea 2016.1 注册服务器..... | 43 |
| 33、ubuntu 安装 notepad++..... | 43 |
| 34、linux 建立快捷方式..... | 43 |
| 35、ubuntu 开放 root 并且免密..... | 43 |
| On the Server..... | 43 |
| On the Client..... | 44 |
| 36、ubuntu 配置国内免费 registry mirror..... | 44 |
| 使用方法..... | 44 |
| 37、vim 编辑替换..... | 44 |
| 38、安装 go 以及 godep..... | 45 |
| 39、K8S-openshift 安装..... | 45 |
| 40、十条命令在一分钟内检查 Linux 服务器性能..... | 45 |
| 41、ubuntu 系统升级..... | 46 |

| | |
|---|----|
| 42、ubuntu 静态网卡配置..... | 46 |
| 43、ubuntu 设置开机启动..... | 47 |
| 44、redis 设置密码..... | 48 |
| 45、docker error could not delete the default bridge network: network bridge has active endpoints..... | 52 |
| 46、Ubuntu 环境下挂载新硬盘 | 52 |
| ①、硬盘分区 Hard disk add new partition..... | 52 |
| ②、硬盘格式化 Format hard disk..... | 57 |
| ③、挂载硬盘分区 Mount hard disk partition..... | 58 |
| 附录 1: fdisk 命令详解 Appendix part 1: | 60 |
| 附录 2: mkfs 命令详解 Appendix part 2: | 60 |
| 附录 3: mount 命令详解 Appendix part 3: | 61 |
| 附录 4: fstab 配置详解 Appendix part 4: | 61 |
| 47、centos 7 自定义开机启动..... | 62 |
| 48、文件授权..... | 63 |
| 49、Ubuntu-16.04 安装搜狗拼音输入法..... | 63 |
| 50、docker delete exited container..... | 64 |
| 51、Ubuntu 安装 QQ..... | 65 |
| 52、Ubuntu 安装 LibreOffice Project..... | 65 |
| 53、Ubuntu 安装 vokoscreen 录屏软件..... | 65 |
| 54、Ubuntu 创建 idea 执行快捷方式 | 65 |
| 55、Docker-compose 启动报错解决(pip install docker-compose) | 65 |

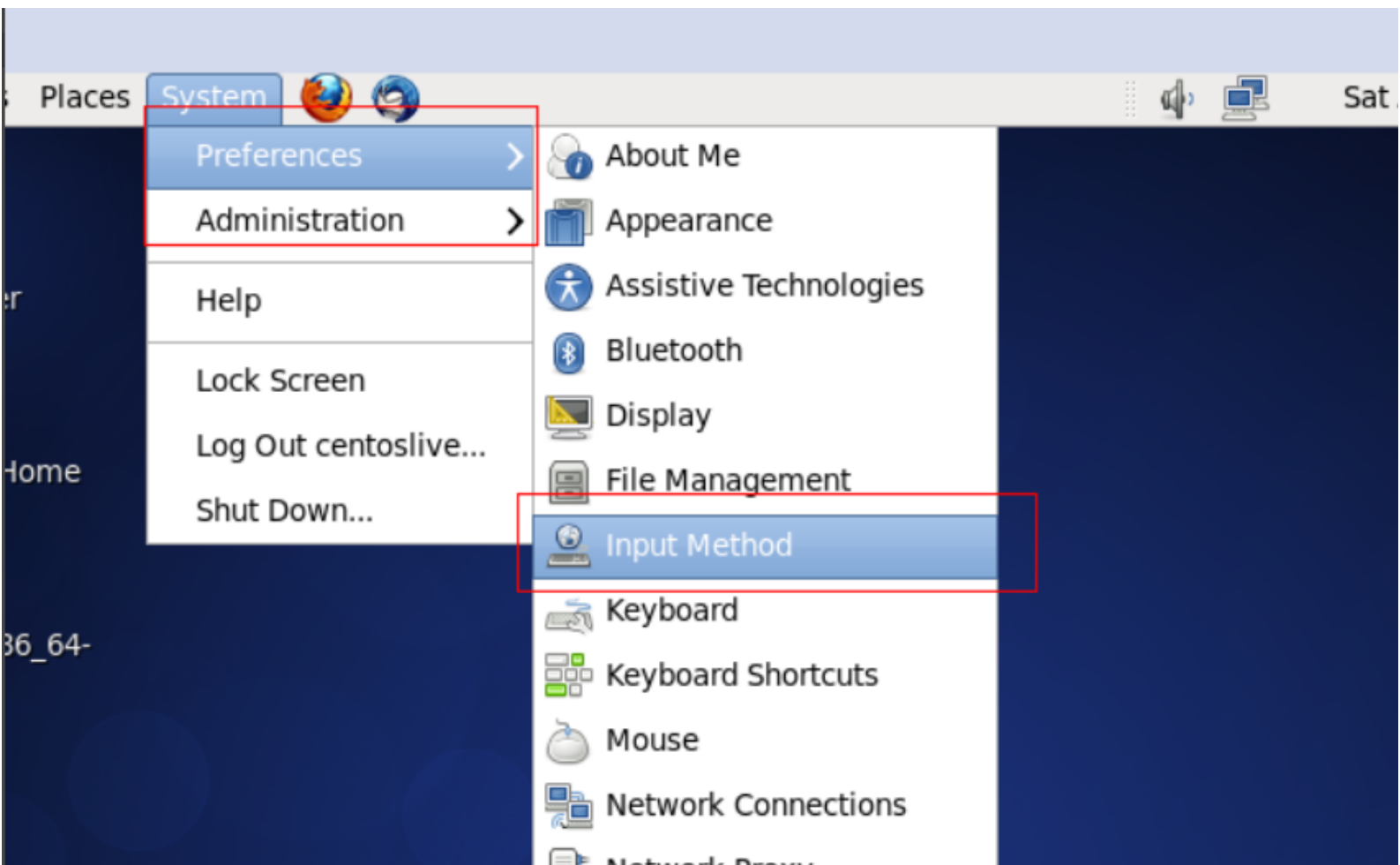
1、安装中文输入法

1.首先进入命令形式的客户端

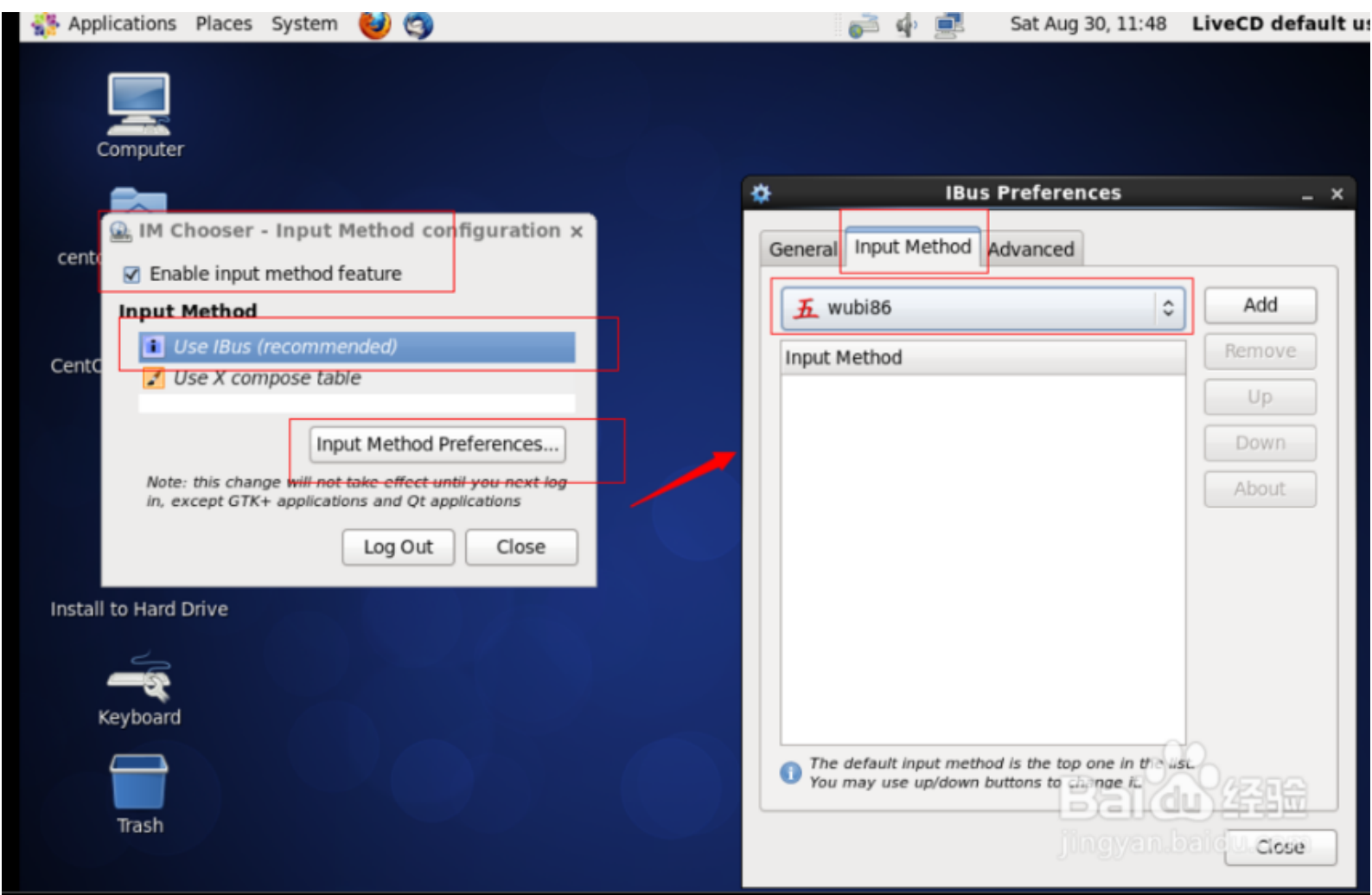
2.切换到 root 用户，输入命令"`su root`"即可，接着输入 `yum install "@Chinese Support"` 命令按 `enter` 执行

3.输入命令后系统会下载一些东西来安装，这个不用管，后面会出来一个页面如图，输入"`y`"即可，后面一个也是，完成后退出客户端

4.这时按照图片中的操作选择，会有"`Input Method`" 选项，开始默认是没有的，有了上面的操作才会出现



5.选择后会出现图片左侧的选项框，点击"`Enable input method feature`" 并选择"`Use IBus(recommended)`",点击 "`Input Method Preferences`" ,会出现图片右边的对话框，在下拉框中选择你需要的输入法后，点击当前对话框右边的"`Add`"按钮，这时输入添加已经成功



6.重启系统，ctrl+space 空格，就可以输入了，看下图效果



看看，有了吧你

百度一下



2、配置网关等 linux 和 CentOS 下网卡启动、配置等 ifcfg-

eth0 教程

it 动力总结系统安装好后，通过以下二个步骤就可以让你的系统正常上网（大多正常情况下）。

步骤 1、配置/etc/sysconfig/network-scripts/ifcfg-eth0 里的文件。it 动力的 CentOS 下的 ifcfg-eth0 的配置详情：

```
[root@localhost ~]# vim /etc/sysconfig/network-scripts/ifcfg-eth0
DEVICE="eth0"
HWADDR="00:0C:29:FD:FF:2A"
NM_CONTROLLED="yes"
```

```
ONBOOT="yes"
IPADDR=192.168.1.31
NETMASK=255.255.255.0
GATEWAY=192.168.1.1
BOOTPROTO=static
```

前面三行是系统自带的，后面就是手动添加的。
这样设置后，记得重启网卡：

```
[root@localhost ~]# /etc/init.d/network stop
[root@localhost ~]# /etc/init.d/network start
```

行了，现在就可以 **PING** 得通网关了，如果还得上网，不必须设置 **DNS**。设置 **DNS** 就详见步骤 2

步骤 2、修改 dns

```
[root@localhost ~]# vi /etc/resolv.conf
在里面添加二个 dns:
nameserver 202.96.134.133
nameserver 8.8.8.8
```

ok，大功告成，即可上网了！！！！

以下是参考，如果按上面步骤不行，可以详细看以下内容：

网络接口配置文件

```
[root@localhost ~]# cat /etc/sysconfig/network-scripts/ifcfg-eth0
# Intel Corporation 82545EM Gigabit Ethernet Controller (Copper)
TYPE=Ethernet      #网卡类型
DEVICE=eth0        #网卡接口名称
ONBOOT=yes         #系统启动时是否自动加载
BOOTPROTO=static   #启用地址协议 --static:静态协议 --bootp 协议 --dhcp 协议
IPADDR=192.168.1.11 #网卡 IP 地址
NETMASK=255.255.255.0 #网卡网络地址
GATEWAY=192.168.1.1 #网卡网关地址
DNS1=10.203.104.41  #网卡 DNS 地址
HWADDR=00:0C:29:13:5D:74 #网卡设备 MAC 地址
BROADCAST=192.168.1.255 #网卡广播地址
```

重新导入 **ifcfg-eth0** 网络配置文件

```
[root@localhost ~]# /etc/init.d/network reload
Shutting down interface eth0:          [ OK ]
Shutting down loopback interface:      [ OK ]
Bringing up loopback interface:        [ OK ]
Bringing up interface eth0:            [ OK ]
```

网卡接口关闭与激活

```
[root@localhost ~]# ifdown eth0 #关闭网络
```

```
[root@localhost ~]# ifup eth0 #启动网络
```

网络服务启动与关闭

方法一:

```
[root@localhost ~]# service network stop #关闭网络服务
```

```
[root@localhost ~]# service network start #启动网络服务
```

```
[root@localhost ~]# service network restart #重启网络服务
```

方法二:

```
[root@localhost ~]# /etc/init.d/network stop
```

```
[root@localhost ~]# /etc/init.d/network start
```

```
[root@localhost ~]# /etc/init.d/network restart
```

网卡状态查询

```
[root@localhost ~]# service network status
```

Configured devices:

lo eth0

Currently active devices:

lo eth0

临时配置网卡信息，无需重启。

```
[root@localhost ~]# ifconfig eth0 10.1.1.10 netmask 255.0.0.0
```

查看网卡接口信息，默认列出所有接口

```
[root@localhost ~]# ifconfig
```

```
eth0    Link encap:Ethernet  HWaddr 00:0C:29:13:5D:74
        inet addr:192.168.1.11  Bcast:192.168.1.255  Mask:255.255.255.0
        inet6 addr: fe80::20c:29ff:fe13:5d74/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:413 errors:0 dropped:0 overruns:0 frame:0
        TX packets:572 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:47701 (46.5 KiB)  TX bytes:64842 (63.3 KiB)
        Base address:0x2000 Memory:d8920000-d8940000

lo       Link encap:Local Loopback
        inet addr:127.0.0.1  Mask:255.0.0.0
        inet6 addr: ::1/128 Scope:Host
        UP LOOPBACK RUNNING  MTU:16436  Metric:1
        RX packets:407 errors:0 dropped:0 overruns:0 frame:0
        TX packets:407 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:0
        RX bytes:70759 (69.1 KiB)  TX bytes:70759 (69.1 KiB)
```

查看当前路由及网关信息

```
[root@localhost ~]# netstat -r
```

Kernel IP routing table

| Destination | Gateway | Genmask | Flags | MSS | Window | irtt | Iface |
|-------------|---------|---------------|-------|-----|--------|------|-------|
| 192.168.1.0 | * | 255.255.255.0 | U | 0 | 0 | 0 | eth0 |

```

169.254.0.0 * 255.255.0.0 U 0 0 0 eth0
default 192.168.1.1 0.0.0.0 UG 0 0 0 eth0DNS: 主机名: CentOS 主
DNS: 202.106.46.151 第二 DNS:202.106.0.20 第三 DNS:8.8.8.8 网络配置: eth0 静态 ip:
192.168.1.106 子网掩码 255.255.255.0 默认网关 IP
192.168.1.1DEVICE=eth0IPADDR=192.168.1.106NETMASK=255.255.255.0BROADCAST=19
2.168.1.255ONBOOT=yesBOOTPROTO=noneGATEWAY=192.168.1.1TYPE=Ethernet"/etc/sys
config/network-scripts/ifcfg-eth0" 11L, 187C

```

3、Linux 设置 SSH 开机启动

chkconfig --level 2345 sshd on --level 指定系统在 2345 运行级别时，通常在 35 中开启就行，开启（on）sshd 服务，关闭则用（off）临时控制启动停止则用：/etc/rc.d/init.d/sshd start | stop | restart

4、Linux 安装 JDK 7

1、先卸载服务器自带的 jdk 软件包

java -version #查看服务器是否安装过 jdk

java version "1.6.0_17"

OpenJDK Runtime Environment (IcedTea6 1.7.4) (rhel-1.21.b17.el6-i386)

OpenJDK Client VM (build 14.0-b16, mixed mode)

rpm -qa |grep gcj #查看服务器安装的 jdk 软件包信息

libgcj-4.4.4-13.el6.i686

java-1.5.0-gcj-1.5.0.0-29.1.el6.i686

yum -y remove java-1.5.0-gcj-1.5.0.0-29.1.el6.i686 #卸载软件包

2、安装 jdk8.0

解压配置安装

mkdir -p /usr/lib/java

sudo tar zxvf jdk-8u73-linux-x64.tar.gz -C /usr/lib/java


```
# sudo mv /usr/lib/java/jdk1.8.0_73/ /usr/lib/java/jdk8
```

3、添加 jdk8.0 到系统环境变量

```
# cp /etc/profile /etc/profile.bak #备份
```

```
# vi /etc/profile #编辑,在最后添加下面的内容
```

```
export JAVA_HOME=/usr/lib/java/jdk8
```

```
export JRE_HOME=${JAVA_HOME}/jre
```

```
export CLASSPATH=.:${JAVA_HOME}/lib:${JRE_HOME}/lib
```

```
export PATH=${JAVA_HOME}/bin:$PATH
```

```
# source /etc/profile #使配置文件立即生效
```

由于系统中可能会有默认的其他版本 JDK，所以，为了将我们安装的 JDK 设置为默认 JDK 版本，还要进行如下工作。

```
update-alternatives --install /usr/bin/java java /usr/lib/jvm/java7/bin/java 300
```

```
update-alternatives --install /usr/bin/javac javac /usr/lib/jvm/java7/bin/javac 300
```

```
update-alternatives --install /usr/bin/jar jar /usr/lib/jvm/java7/bin/jar 300
```

```
update-alternatives --install /usr/bin/javah javah /usr/lib/jvm/java7/bin/javah 300
```

```
update-alternatives --install /usr/bin/javap javap /usr/lib/jvm/java7/bin/javap 300
```

执行下面命令，设置默认版本，此命令执行后，系统会列出当前存在的各种 JDK 版本，会提示你选择

```
# update-alternatives --config java
```

4、测试

```
# java -version # 测试是否成功
```

```
java version "1.7.0_76"
```

```
Java(TM) SE Runtime Environment (build 1.7.0_76-b13)
```

```
Java HotSpot(TM) 64-Bit Server VM (build 24.76-b04, mixed mode)
```

5、Linux JDK 7 环境变量配置

按变量的生存周期来划分，Linux 变量可分为两类，它们的修改方法如下：

(1) 永久的：需要修改配置文件，变量永久生效。

常见的配置文件包括：

(1-1) /etc/profile：对所有用户生效；此文件为系统的每个用户设置环境信息,当用户第一次登录时,该文件被执行；并从/etc/profile.d 目录的配置文件中搜集 shell 的设置

例如：编辑/etc/profile 文件，添加 CLASSPATH 变量

```
# vi /etc/profile
```

添加一行：

```
export CLASSPATH=../JAVA_HOME/lib:$JAVA_HOME/jre/lib
```

修改后需要执行重新登录才能生效，也可以执行命令 source /etc/profile 来生效

(1-2) /etc/bashrc：对所有用户生效；为每一个运行 bash shell 的用户执行此文件.当

bash shell 被打开时,该文件被读取
编辑方法如上, 不再赘述

(1-3) **~/.bash_profile**: 仅会对当前用户有效; 每个用户都可使用该文件输入专用于自己使用的 **shell** 信息,当用户登录时,该文件仅仅执行一次

例如: 编辑 **guok** 用户目录(/home/guok)下的.bash_profile

```
$ vi /home/guok/.bash.profile
```

添加如下内容:

```
export CLASSPATH=./JAVA_HOME/lib;$JAVA_HOME/jre/lib
```

修改后需要执行重新登录才能生效, 也可以执行命令 **source /etc/profile** 来生效

(1-4) **~/.bashrc**: 仅会对当前用户有效; 该文件包含专用于你的 **bash shell** 的 **bash** 信息, 当登录时以及每次打开新的 **shell** 时,该文件被读取

编辑方法如上, 不再赘述

另外, **~/.bashrc** 等中设定的变量(局部)只能继承/etc/profile 中的变量,他们是"父子"关系

综述, 对上述文件修改, 添加你需要的变量, 在启动一个 **shell** (终端, **terminal**) 时, 你所定义的变量均会生效的。

(2) 临时的: 使用 **export** 命令声明即可, 变量只在当前的 **shell(BASH)**或其子 **shell(BASH)**下是有效的,在关闭 **shell** 后失效, 再打开新 **shell** 时就没有这个变量, 需要使用的話还需要重新定义

在 **shell** 的命令行下直接使用[**export** 变量名=变量值] 定义变量

环境变量的查看

(1) 使用 **echo** 命令查看单个环境变量。例如:

```
echo $PATH
```

(2) 使用 **env** 查看所有环境变量。例如:

```
env
```

(3) 使用 **set** 查看所有本地定义的环境变量。例如:

```
set
```

另外, **unset** 可以删除指定的环境变量。

常用的环境变量

PATH 决定了 **shell** 将到哪些目录中寻找命令或程序

HOME 当前用户主目录

HISTSIZE 历史记录数

LOGNAME 当前用户的登录名

HOSTNAME 指主机的名称

SHELL 当前用户 **Shell** 类型

LANGUAGE 语言相关的环境变量, 多语言可以修改此环境变量

MAIL 当前用户的邮件存放目录

PS1 基本提示符, 对于 **root** 用户是#, 对于普通用户是\$

6、secureCRT 上传下载使用

SecureCR 下的文件传输协议有 ASCII 、Xmodem 、Ymodem 、Zmodem

ASCII: 这是最快的传输协议, 但只能传送文本文件。

Xmodem: 这种古老的传输协议速度较慢, 但由于使用了 CRC 错误侦测方法, 传输的准确率可高达 99.6%。

Ymodem: 这是 Xmodem 的改良版, 使用了 1024 位区段传送, 速度比 Xmodem 要快。

Zmodem: Zmodem 采用了串流式 (streaming) 传输方式, 传输速度较快, 而且还具有自动改变区段大小和断点续传、快速错误侦测等功能。这是目前最流行的文件传输协议

设定默认上传[下载](#)目录

options->session options ->Terminal->Xmodem/Zmodem 下
在右边设置上传和下载的目录

使用 Zmodem 从客户端上传文件到 linux 服务器

1. 在用 SecureCRT 登陆 linux 终端。

2. 选中你要放置上传文件的路径, 在目录下然后输入 rz 命令, SecureCRT 会弹出文件选择对话框, 在查找范围中找到你要上传的文件, 按 Add 按钮。然后 OK 就可以把文件上传到 linux 上了。

或者在 Transfer->Zmodem Upoad list 弹出文件选择对话框, 选好文件后按 Add 按钮。然后 OK 窗口自动关闭。然后在 linux 下选中存放文件的目录, 输入 rz 命令。linux 就把那个文件上传到这个目录下了。

使用 Zmodem 下载文件到客户端:

sz filename

zmodem 接收可以自行启动. 下载的文件存放在你设定的默认下载目录下。

如果 rz、sz 命令无效, 所以请大家先安装 rzs-0.12.20-853.2.i586.rpm
具体安装方法为:

rpm -ivh rzs-0.12.20-853.2.i586.rpm

1. 遇到 rz、sz 命令无效

-bash: rz: command not found

方法 1：自动安装 lrzsz:

```
# yum -y install lrzsz
```

方法 2：手动安装

2.1 下载

地址: <http://www.ohse.de/uwe/software/lrzsz.html>

下载到一个压缩包文件: lrzsz-0.12.20.tar.gz

2.2 解压

```
tar -xzf lrzsz-0.12.20.tar.gz
```

2.3 安装

```
cd lrzsz-0.12.20
```

```
./configure --prefix=/usr/local/lrzsz
```

```
sudo make
```

```
sudo make install
```

2.4 创建连接

```
cd /usr/bin
```

```
sudo ln -s /usr/local/lrzsz/bin/lrz rz
```

```
sudo ln -s /usr/local/lrzsz/bin/lzs sz
```

7、解压文件

注意红色 -C

```
[root@172 local]# tar -xzf xxx.tar.gz -C /usr/local/xxx
```

```
[root@172 local]# ls
```

```
zip -r tomcat.zip ./tomcat/*
```

```
unzip tomcat.zip
```

8、Linux 安装 Mysql

①yum 安装方法

查看已经安装的 mysql

```
yum list installed mysql*
```

```
rpm -qa | grep mysql*
```

```
yum install mysql
```

```
yum install mysql-server
```

```
yum install mysql-devel
chkconfig -R mysql /var/lib/mysql
chmod -R 770 /var/lib/mysql
service mysqld start
mysql
SET PASSWORD FOR 'root'@'localhost' = PASSWORD('Hisilc');
```

更新

如要其他机器能访问，在 `mysql.user` 中添加一个 `Host` 为 `'%'` 的 `user`，然后 `flush privileges;`，最后防火墙加一句类似这样的语句即可(开通 3306 端口)：

```
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 3306 -j ACCEPT
```

如果要 **reboot** 自启动：

```
chkconfig --levels 345 mysqld on
```

yum 卸载 mysql

```
yum -y remove mysql*
```

如果是 **rpm** 安装的话卸载方式：

```
rpm -e mysql
```

② 普通自己安装方法

一、编译安装 MySQL 前的准备工作

安装编译源码所需的工具和库

[sql] view plaincopy 在 CODE 上查看代码片派生到我的代码片

```
01.yum install gcc gcc-c++ ncurses-devel perl
```

安装 cmake，从 <http://www.cmake.org> 下载源码并编译安装

[sql] view plaincopy 在 CODE 上查看代码片派生到我的代码片

```
01.wget http://www.cmake.org/files/v2.8/cmake-2.8.10.2.tar.gz
```

```
02.tar -xzf cmake-2.8.10.2.tar.gz
```

```
03.cd cmake-2.8.10.2
```

```
04. ./bootstrap ; make ; make install
```

```
05.cd ~
```

二、设置 MySQL 用户和组

新增 mysql 用户组

[sql] view plaincopy 在 CODE 上查看代码片派生到我的代码片
01.groupadd mysql

新增 mysql 用户

[sql] view plaincopy 在 CODE 上查看代码片派生到我的代码片
01.useradd -r -g mysql mysql

三、新建 MySQL 所需要的目录

新建 mysql 安装目录

[sql] view plaincopy 在 CODE 上查看代码片派生到我的代码片
01.mkdir -p /usr/local/mysql

新建 mysql 数据库数据文件目录

[sql] view plaincopy 在 CODE 上查看代码片派生到我的代码片
01.mkdir -p /data/mysqlldb

四、下载 MySQL 源码包并解压

从 <http://dev.mysql.com/downloads/mysql/> 直接下载源码，解压 mysql-5.6.16.tar.gz (<http://www.quseqi.com/>这个网站就是用的 5.6.16 版本)

[sql] view plaincopy 在 CODE 上查看代码片派生到我的代码片
01.wget <http://www.kakapart.com/files/mysql-5.6.16.tar.gz>
02.tar -zxv -f mysql-5.6.16.tar.gz
03.cd mysql-5.6.16

五、编译安装 MySQL

从 mysql5.5 起，mysql 源码安装开始使用 cmake 了，设置源码编译配置脚本。
设置编译参数

[sql] view plaincopy 在 CODE 上查看代码片派生到我的代码片
01.cmake \
02.-DCMAKE_INSTALL_PREFIX=/usr/local/mysql \

```
03.-DMYSQL_UNIX_ADDR=/usr/local/mysql/mysql.sock \
04.-DDEFAULT_CHARSET=utf8 \
05.-DDEFAULT_COLLATION=utf8_general_ci \
06.-DWITH_INNOBASE_STORAGE_ENGINE=1 \
07.-DWITH_ARCHIVE_STORAGE_ENGINE=1 \
08.-DWITH_BLACKHOLE_STORAGE_ENGINE=1 \
09.-DMYSQL_DATADIR=/data/mysqlldb \
10.-DMYSQL_TCP_PORT=3306 \
11.-DENABLE_DOWNLOADS=1
```

-DCMAKE_INSTALL_PREFIX=dir_name 设置 mysql 安装目录
-DMYSQL_UNIX_ADDR=file_name 设置监听套接字路径，这必须是一个绝对路径名。默认为/tmp/mysql.sock
-DDEFAULT_CHARSET=charset_name 设置服务器的字符集。
缺省情况下，MySQL 使用 latin1 的（CP1252 西欧）字符集。cmake/character_sets.cmake 文件包含允许的字符集名称列表。
-DDEFAULT_COLLATION=collation_name 设置服务器的排序规则。
-DWITH_INNOBASE_STORAGE_ENGINE=1
-DWITH_ARCHIVE_STORAGE_ENGINE=1
-DWITH_BLACKHOLE_STORAGE_ENGINE=1
-DWITH_PERFSCHEMA_STORAGE_ENGINE=1 存储引擎选项：

MyISAM，MERGE，MEMORY，和 CSV 引擎是默认编译到服务器中，并不需要明确地安装。

静态编译一个存储引擎到服务器，使用-DWITH_engine_STORAGE_ENGINE= 1

可用的存储引擎值有：ARCHIVE, BLACKHOLE, EXAMPLE, FEDERATED, INNOBASE (InnoDB), PARTITION (partitioning support), 和 PERFSCHEMA (Performance Schema)

-DMYSQL_DATADIR=dir_name 设置 mysql 数据库文件目录
-DMYSQL_TCP_PORT=port_num 设置 mysql 服务器监听端口，默认为 3306
-DENABLE_DOWNLOADS=bool 是否要下载可选的文件。例如，启用此选项（设置为 1），cmake 将下载谷歌所使用的测试套件运行单元测试。
注：重新运行配置，需要删除 CMakeCache.txt 文件

[sql] view plaincopy 在 CODE 上查看代码片派生到我的代码片

```
01.rm CMakeCache.txt
```

编译源码

[sql] view plaincopy 在 CODE 上查看代码片派生到我的代码片

```
01.make
```

安装

[sql] view plaincopy 在 CODE 上查看代码片派生到我的代码片

```
01.make install
```

六、修改 mysql 目录所有者和组

修改 mysql 安装目录

[sql] view plaincopy 在 CODE 上查看代码片派生到我的代码片

01.cd /usr/local/mysql

02.chown -R mysql:mysql .

修改 mysql 数据库文件目录

[sql] view plaincopy 在 CODE 上查看代码片派生到我的代码片

01.cd /data/mysqlldb

02.chown -R mysql:mysql .

七、初始化 mysql 数据库

[sql] view plaincopy 在 CODE 上查看代码片派生到我的代码片

01.cd /usr/local/mysql

02.scripts/mysql_install_db --user=mysql --datadir=/data/mysqlldb

八、复制 mysql 服务启动配置文件

[sql] view plaincopy 在 CODE 上查看代码片派生到我的代码片

01.cp /usr/local/mysql/support-files/my-default.cnf /etc/my.cnf

注：如果/etc/my.cnf 文件存在，则覆盖。

九、复制 mysql 服务启动脚本及加入 PATH 路径

[sql] view plaincopy 在 CODE 上查看代码片派生到我的代码片

cp support-files/mysql.server /etc/init.d/mysqld

vim /etc/profile

```
PATH=/usr/local/mysql/bin:/usr/local/mysql/lib:$PATH
```

```
export PATH
```

source /etc/profile

十、启动 mysql 服务并加入开机自启动(可选这个步骤，以后可以自己启动的)

service mysqld start

chkconfig --level 35 mysqld on

十一、检查 mysql 服务是否启动

[sql] view plaincopy 在 CODE 上查看代码片派生到我的代码片

01.netstat -tulnp | grep 3306

02.mysql -u root -p

密码为空，如果能登陆上，则安装成功。

十二、修改 MySQL 用户 root 的密码

[sql] view plaincopy 在 CODE 上查看代码片派生到我的代码片

01.mysqladmin -u root password '123456'

注：也可运行安全设置脚本，修改 MySQL 用户 root 的密码，同时可禁止 root 远程连接，移除 test 数据库和匿名用户。

[sql] view plaincopy 在 CODE 上查看代码片派生到我的代码片

01./usr/local/mysql/bin/mysql_secure_installation

十三、可能会出现的错误

[sql] view plaincopy 在 CODE 上查看代码片派生到我的代码片

01.问题:

02.Starting MySQL..The server quit without updating PID file
([FAILED]/mysql/Server03.mylinux.com.pid).

03.解决:

04.修改/etc/my.cnf 中 datadir,指向正确的 mysql 数据库文件目录

[sql] view plaincopy 在 CODE 上查看代码片派生到我的代码片

01.问题:

02.ERROR 2002 (HY000): Can't connect to local MySQL server through socket '/tmp/mysql.sock'
(2)

03.解决:

04.新建一个链接或在 mysql 中加入 -S 参数, 直接指出 mysql.sock 位置。

05.ln -s /usr/local/mysql/data/mysql.sock /tmp/mysql.sock

06.

07./usr/local/mysql/bin/mysql -u root -S /usr/local/mysql/data/mysql.sock

[sql] view plaincopy 在 CODE 上查看代码片派生到我的代码片

01.MySQL 问题解决: -bash:mysql:command not found

02.因为 mysql 命令的路径在/usr/local/mysql/bin 下面,所以你直接使用 mysql 命令时,

03.系统在/usr/bin 下面查此命令,所以找不到了

04. 解决办法是:

05. ln -s /usr/local/mysql/bin/mysql /usr/bin 做个链接即可

9、Linux 安装 zookeeper

下载 zookeeper

安装配置

解压配置安装

```
# mkdir -p /usr/lib/zookeeper
```

```
#tar zxvf zookeeper-3.4.6.tar.gz -C /usr/lib/zookeeper
```

```
#mv /usr/lib/zookeeper/zookeeper-3.4.6 /usr/lib/zookeeper/node-01
```

```
#mkdir -p /usr/lib/zookeeper/node-01/data
```

```
#mkdir -p /usr/lib/zookeeper/node-01/logs
```

```
[root@iZ94uh2833tZ node-01]# cd data
```

```
[root@iZ94uh2833tZ data]# vi myid
```

```
1
```

```
"myid" [New] 1L, 2C written
```

```

vi /etc/sysconfig/iptables
# Generated by iptables-save v1.4.7 on Tue Nov 10 06:24:44 201
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -p udp -m state --state NEW -m udp --dport 137 -j ACCEPT
-A INPUT -p udp -m state --state NEW -m udp --dport 138 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 139 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT
#zookeeper
-A INPUT -p tcp -m state --state NEW -m tcp --dport 2181 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 2881 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 3881 -j ACCEPT
## dubbo-admin-tomcat:8080
-A INPUT -m state --state NEW -m tcp -p tcp --dport 8080 -j ACCEPT

-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A FORWARD -j REJECT --reject-with icmp-host-prohibited
COMMIT

```

~

"sysconfig/iptables" 23L, 969C written

```

[root@iZ94uh2833tZ etc]# service iptables start
iptables: Applying firewall rules: [ OK ]
[root@iZ94uh2833tZ etc]# clear
[root@iZ94uh2833tZ etc]# service iptables status
#开机启动
[root@iZ94uh2833tZ etc]# vi /etc/rc.local
#!/bin/sh
#
# This script will be executed *after* all the other init scripts.
# You can put your own initialization stuff in here if you don't
# want to do the full Sys V style init stuff.

```

```

touch /var/lock/subsys/local
su - root -c '/usr/lib/zookeeper/node-01/bin/zkServer.sh start'

```

10、Linux 安装 dubbo 服务维护

在 `/usr/lib/hisilc/service/` 下 对应 `xxx` 文件下,建立 `service-xxx.sh` 格式如下:

```
#!/bin/sh
##javaenv
export JAVA_HOME=/usr/lib/java/jdk7
export JRE_HOME=$JAVA_HOME/jre
##servicename
APP_NAME=user
SERVICE_DIR=/usr/lib/hisilc/service/$APP_NAME
SERVICE_NAME=hisilc-service-$APP_NAME
JAR_NAME=$SERVICE_NAME.jar
PID=$SERVICE_NAME.pid

cd $SERVICE_DIR

case "$1" in

    start)
        nohup $JRE_HOME/bin/java -Xms256m -Xmx512m -jar $JAR_NAME >/dev/null 2>&1 &
        echo $! > $SERVICE_DIR/$PID
        echo "=== start $SERVICE_NAME"
        ;;

    stop)
        kill `cat $SERVICE_DIR/$PID`
        rm -rf $SERVICE_DIR/$PID
        echo "=== stop $SERVICE_NAME"

        sleep 5
        ##
        ## edu-service-aa.jar
        ## edu-service-aa-bb.jar
        P_ID=`ps -ef | grep -w "$SERVICE_NAME" | grep -v "grep" | awk '{print $2}'`
        if [ "$P_ID" == "" ]; then
            echo "=== $SERVICE_NAME process not exists or stop success"
        else
            echo "=== $SERVICE_NAME process pid is:$P_ID"
            echo "=== begin kill $SERVICE_NAME process, pid is:$P_ID"
            kill -9 $P_ID
        fi
    esac
```

```

fi
;;

restart)
    $0 stop
    sleep 2
    $0 start
    echo "=== restart $SERVICE_NAME"
    ;;

*)
    ## restart
    $0 stop
    sleep 2
    $0 start
    ;;

esac
exit 0

```

防火墙开启 **dubbo** 端口

```

vi /etc/sysconfig/iptables
# Generated by iptables-save v1.4.7 on Tue Nov 10 06:24:44 201
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -p udp -m state --state NEW -m udp --dport 137 -j ACCEPT
-A INPUT -p udp -m state --state NEW -m udp --dport 138 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 139 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT
#dubbo service jar port
-A INPUT -p tcp -m state --state NEW -m tcp --dport 20880 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 20881 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 20882 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 20883 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 20884 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 20885 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 20886 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 20887 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 20889 -j ACCEPT

```

```
-A INPUT -p tcp -m state --state NEW -m tcp --dport 20990 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 20991 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 20992 -j ACCEPT
```

```
-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A FORWARD -j REJECT --reject-with icmp-host-prohibited
COMMIT
```

```
# Completed on Tue Nov 10 06:24:45 2015
```

启动所有 **dubbo** 服务

xxx 为包名 user,store,order 等等

```
/usr/lib/hisilc/service/xxx/./service-xxx.sh start
```

```
/usr/lib/hisilc/service/appVersion/./service-appVersion.sh start
/usr/lib/hisilc/service/complaint/./service-complaint.sh start
/usr/lib/hisilc/service/coreEmployee/./service-coreEmployee.sh start
/usr/lib/hisilc/service/equipment/./service-equipment.sh start
/usr/lib/hisilc/service/locationIndex/./service-locationIndex.sh start
/usr/lib/hisilc/service/order/./service-order.sh start
/usr/lib/hisilc/service/signContract/./service-signContract.sh start
/usr/lib/hisilc/service/sms/./service-sms.sh start
/usr/lib/hisilc/service/store/./service-store.sh start
/usr/lib/hisilc/service/subCompany/./service-subCompany.sh start
/usr/lib/hisilc/service/user/./service-user.sh start
/usr/lib/hisilc/service/zone/./service-zone.sh start
```

主要启动:

```
/usr/lib/hisilc/service/provider/./service-provider.sh start
```

```
/usr/lib/hisilc/service/order/./service-order.sh start
/usr/lib/hisilc/service/user/./service-user.sh start
/usr/lib/hisilc/service/store/./service-store.sh start
/usr/lib/hisilc/service/zone/./service-zone.sh start
/usr/lib/hisilc/service/locationIndex/./service-locationIndex.sh start
```

```
/usr/lib/hisilc/service/appVersion/./service-appVersion.sh start
/usr/lib/hisilc/service/complaint/./service-complaint.sh start
/usr/lib/hisilc/service/coreEmployee/./service-coreEmployee.sh start
/usr/lib/hisilc/service/equipment/./service-equipment.sh start
/usr/lib/hisilc/service/signContract/./service-signContract.sh start
```

```
/usr/lib/hisilc/service/sms/./service-sms.sh start
/usr/lib/hisilc/service/subCompany/./service-subCompany.sh start
```

```
/usr/lib/hisilc/service/order/./service-order.sh stop
/usr/lib/hisilc/service/user/./service-user.sh stop
/usr/lib/hisilc/service/store/./service-store.sh stop
/usr/lib/hisilc/service/zone/./service-zone.sh stop
/usr/lib/hisilc/service/locationIndex/./service-locationIndex.sh stop
```

暂停所有 **dubbo** 服务:

```
/usr/lib/hisilc/service/appVersion/./service-appVersion.sh stop
/usr/lib/hisilc/service/complaint/./service-complaint.sh stop
/usr/lib/hisilc/service/coreEmployee/./service-coreEmployee.sh stop
/usr/lib/hisilc/service/equipment/./service-equipment.sh stop
/usr/lib/hisilc/service/locationIndex/./service-locationIndex.sh stop
/usr/lib/hisilc/service/order/./service-order.sh stop
/usr/lib/hisilc/service/signContract/./service-signContract.sh stop
/usr/lib/hisilc/service/sms/./service-sms.sh stop
/usr/lib/hisilc/service/store/./service-store.sh stop
/usr/lib/hisilc/service/subCompany/./service-subCompany.sh stop
/usr/lib/hisilc/service/user/./service-user.sh stop
/usr/lib/hisilc/service/zone/./service-zone.sh stop
```

重启 所有 **dubbo** 服务

```
/usr/lib/hisilc/service/appVersion/./service-appVersion.sh restart
/usr/lib/hisilc/service/complaint/./service-complaint.sh restart
/usr/lib/hisilc/service/coreEmployee/./service-coreEmployee.sh restart
/usr/lib/hisilc/service/equipment/./service-equipment.sh restart
/usr/lib/hisilc/service/locationIndex/./service-locationIndex.sh restart
/usr/lib/hisilc/service/order/./service-order.sh restart
/usr/lib/hisilc/service/signContract/./service-signContract.sh restart
/usr/lib/hisilc/service/sms/./service-sms.sh restart
/usr/lib/hisilc/service/store/./service-store.sh restart
/usr/lib/hisilc/service/subCompany/./service-subCompany.sh restart
/usr/lib/hisilc/service/user/./service-user.sh restart
/usr/lib/hisilc/service/zone/./service-zone.sh restart
```

```
/usr/lib/hisilc/service/order/./service-order.sh start
/usr/lib/hisilc/service/user/./service-user.sh start
/usr/lib/hisilc/service/store/./service-store.sh start
```

```
/usr/lib/hisilc/service/order/./service-order.sh stop
/usr/lib/hisilc/service/user/./service-user.sh stop
/usr/lib/hisilc/service/store/./service-store.sh stop
```

```
/usr/lib/hisilc/service/provider/./service-provider.sh start
/usr/lib/hisilc/service/provider/./service-provider.sh stop
cd /usr/lib/hisilc/service/provider/lib/
rz -y
cd /usr/lib/hisilc/service/provider/
rz -y
```

修改 **config** 依赖配置 文件 (覆盖原有的)

```
cd /usr/lib/hisilc/service/appVersion/lib/
rz -y
cd /usr/lib/hisilc/service/complaint/lib/
rz -y
cd /usr/lib/hisilc/service/coreEmployee/lib/
rz -y
cd /usr/lib/hisilc/service/equipment/lib/
rz -y
cd /usr/lib/hisilc/service/locationIndex/lib/
rz -y
cd /usr/lib/hisilc/service/order/lib/
rz -y
cd /usr/lib/hisilc/service/signContract/lib/
rz -y
cd /usr/lib/hisilc/service/sms/lib/
rz -y
cd /usr/lib/hisilc/service/store/lib/
rz -y
cd /usr/lib/hisilc/service/subCompany/lib/
rz -y
cd /usr/lib/hisilc/service/user/lib/
rz -y
```

```
cd /usr/lib/hisilc/service/zone/lib/  
rz -y
```

部署并覆盖 **jar** 包 **dubbo** 服务 **jar** 包

```
cd /usr/lib/hisilc/service/appVersion/  
rz -y  
cd /usr/lib/hisilc/service/complaint/  
rz -y  
cd /usr/lib/hisilc/service/coreEmployee/  
rz -y  
cd /usr/lib/hisilc/service/equipment/  
rz -y  
cd /usr/lib/hisilc/service/locationIndex/  
rz -y  
cd /usr/lib/hisilc/service/order/  
rz -y  
cd /usr/lib/hisilc/service/signContract/  
rz -y  
cd /usr/lib/hisilc/service/sms/  
rz -y  
cd /usr/lib/hisilc/service/store/  
rz -y  
cd /usr/lib/hisilc/service/subCompany/  
rz -y  
cd /usr/lib/hisilc/service/user/  
rz -y  
cd /usr/lib/hisilc/service/zone/  
rz -y
```

```
cd /usr/lib/hisilc/service/app/  
rz -y
```

部署 **dext war** 包

```
cd /usr/lib/tomcat/tomcat-1/webapps/  
rz -y  
cp dext.war /usr/lib/tomcat/tomcat-2/webapps/
```

```
cd /usr/lib/tomcat/tomcat-2/webapps/  
rz -y
```

启动 **tomcat**

```
/usr/lib/tomcat/tomcat-1/bin/startup.sh  
/usr/lib/tomcat/tomcat-2/bin/startup.sh
```

停止


```
/usr/lib/tomcat/tomcat-1/bin/shutdown.sh
/usr/lib/tomcat/tomcat-2/bin/shutdown.sh
重启:
/usr/lib/tomcat/tomcat-1/bin/restart.sh
/usr/lib/tomcat/tomcat-2/bin/restart.sh
```

启动 **dubbo-monitor** 监控

```
/usr/lib/dubbo-monitor/bin/restart.sh
```

启动 控制台

```
/usr/lib/dubbo-admin-tomcat/bin/startup.sh
```

修改 **dubbo** 服务 启动命令参数

```
vi /usr/lib/hisilc/service/appVersion/service-appVersion.sh
vi /usr/lib/hisilc/service/complaint/service-complaint.sh
vi /usr/lib/hisilc/service/coreEmployee/service-coreEmployee.sh
vi /usr/lib/hisilc/service/equipment/service-equipment.sh
vi /usr/lib/hisilc/service/locationIndex/service-locationIndex.sh
vi /usr/lib/hisilc/service/order/service-order.sh
vi /usr/lib/hisilc/service/signContract/service-signContract.sh
vi /usr/lib/hisilc/service/sms/service-sms.sh
vi /usr/lib/hisilc/service/store/service-store.sh
vi /usr/lib/hisilc/service/user/service-user.sh
vi /usr/lib/hisilc/service/zone/service-zone.sh
```

11、Linux 中 nginx 的维护

启动 **nginx**

```
/usr/local/nginx/sbin/nginx
```

重启

```
/usr/local/nginx/sbin/nginx -s reload
```

测试 **Nginx** 是否安装成功

```
/usr/local/nginx/sbin/nginx -t
```

测试Nginx是否安装成功

```
# /usr/local/nginx/sbin/nginx -t
```

keenalived 安装在 etc/keepalived 下面

网卡选择问题 ,默认

```
vi /etc/resolv.conf
```

```
options timeout:1 attempts:1 rotate
nameserver 112.74.212.1
nameserver 100.100.2.137
```

nameserver 100.100.2.136

12、Linux 中监听具体端口

```
ss -tnlp | grep 20880
netstat -ntlp
| grep 20880
```

```
ss -tnlp | grep -E 'nginx|redis|java'
```

Linux 高可用 HA 之 Nginx+Redis+Tomcat 集群实现 session 保持和共享

http://www.dwhd.org/20150604_095952.html

nproc 查看 cpu 是几核的

13、Linux 中 redis 安装共享 session

查看文档 http://www.dwhd.org/20150604_095952.html

1.构建 tomcat-redis-session-manager-master

```
[root@Legion100 /tmp]# wget -c http://www.dwhd.org/script/tar_gz_bz2/gradle-2.4-all.zip
[root@Legion100 /tmp]# unzip -q gradle-2.4-all.zip -d /usr/local/
[root@Legion100 /tmp]# ln -sv /usr/local/gradle-2.4 /usr/local/gradle
[root@Legion100 /tmp]# echo "export PATH=/usr/local/gradle/bin:\$PATH" >
/etc/profile.d/gradle2.4.sh
[root@Legion100 /tmp]# . /etc/profile.d/gradle2.4.sh
[root@Legion100 /tmp]# which gradle
/usr/local/gradle/bin/gradle
[root@Legion100 /tmp]# git clone https://github.com/jcoleman/tomcat-redis-session-manager.git
[root@Legion100 /tmp]# cd tomcat-redis-session-manager/
[root@Legion100 /tmp/tomcat-redis-session-manager]# mv build.gradle build.gradle_backup
[root@Legion100 /tmp/tomcat-redis-session-manager]# wget -q
http://www.dwhd.org/script/build.gradle
[root@Legion100 /tmp/tomcat-redis-session-manager]# gradle build -x test copyJars
:compileJava
Download https://repo1.maven.org/maven2/org/apache/tomcat/tomcat-catalina/7.0.27/tomcat-
catalina-7.0.27.pom
Download https://repo1.maven.org/maven2/redis/clients/jedis/2.5.2/jedis-2.5.2.pom
```

Download <https://repo1.maven.org/maven2/org/sonatype/oss/oss-parent/7/oss-parent-7.pom>
Download <https://repo1.maven.org/maven2/org/apache/commons/commons-pool2/2.2/commons-pool2-2.2.pom>
Download <https://repo1.maven.org/maven2/org/apache/commons/commons-parent/33/commons-parent-33.pom>
Download <https://repo1.maven.org/maven2/org/apache/apache/13/apache-13.pom>
Download <https://repo1.maven.org/maven2/org/apache/tomcat/tomcat-servlet-api/7.0.27/tomcat-servlet-api-7.0.27.pom>
Download <https://repo1.maven.org/maven2/org/apache/tomcat/tomcat-juli/7.0.27/tomcat-juli-7.0.27.pom>
Download <https://repo1.maven.org/maven2/org/apache/tomcat/tomcat-annotations-api/7.0.27/tomcat-annotations-api-7.0.27.pom>
Download <https://repo1.maven.org/maven2/org/apache/tomcat/tomcat-api/7.0.27/tomcat-api-7.0.27.pom>
Download <https://repo1.maven.org/maven2/org/apache/tomcat/tomcat-util/7.0.27/tomcat-util-7.0.27.pom>
Download <https://repo1.maven.org/maven2/org/apache/tomcat/tomcat-catalina/7.0.27/tomcat-catalina-7.0.27.jar>
Download <https://repo1.maven.org/maven2/redis/clients/jedis/2.5.2/jedis-2.5.2.jar>
Download <https://repo1.maven.org/maven2/org/apache/commons/commons-pool2/2.2/commons-pool2-2.2.jar>
Download <https://repo1.maven.org/maven2/org/apache/tomcat/tomcat-servlet-api/7.0.27/tomcat-servlet-api-7.0.27.jar>
Download <https://repo1.maven.org/maven2/org/apache/tomcat/tomcat-juli/7.0.27/tomcat-juli-7.0.27.jar>
Download <https://repo1.maven.org/maven2/org/apache/tomcat/tomcat-annotations-api/7.0.27/tomcat-annotations-api-7.0.27.jar>
Download <https://repo1.maven.org/maven2/org/apache/tomcat/tomcat-api/7.0.27/tomcat-api-7.0.27.jar>
Download <https://repo1.maven.org/maven2/org/apache/tomcat/tomcat-util/7.0.27/tomcat-util-7.0.27.jar>
:processResources UP-TO-DATE
:classes
:jar
:javadoc
:javadocJar
:sourcesJar
:assemble
:check
:build
:copyJars

BUILD SUCCESSFUL

Total time: 27.804 secs

This build could be faster, please consider using the Gradle Daemon:

http://gradle.org/docs/2.4/userguide/gradle_daemon.html

```
[root@Legion100 /tmp/tomcat-redis-session-manager]# mkdir /tmp/jar
```

```
[root@Legion100 /tmp/tomcat-redis-session-manager]# cp -a build/libs/*.jar /tmp/jar/ && cp -a dist/*.jar /tmp/jar/
```

```
[root@Legion100 /tmp/tomcat-redis-session-manager]#
```

```
[root@Legion100 ~]# service redis-server stop
```

```
停止 redis-server: [确定]
```

```
[root@Legion100 ~]# service redis-server start
```

```
正在启动 redis-server: [确定]
```

```
[root@Legion100 ~]# service nginx start
```

```
正在启动 nginx: [确定]
```

```
[root@Legion100 ~]# ss -tnlp | grep -E 'nginx|redis|java'
```

2.安装 redis

```
wget http://download.redis.io/releases/redis-2.8.13.tar.gz
```

3.启动 redis

```
service redis start
```

14、Linux 下使用 yum 安装 mysql

1、安装

查看有没有安装过:

```
yum list installed mysql*
```

```
rpm -qa | grep mysql*
```

查看有没有安装包:

```
yum list mysql*
```

安装 mysql 客户端:

```
yum install mysql
```

安装 mysql 服务器端:

```
yum install mysql-server
```

```
yum install mysql-devel
```

www.2cto.com

2、启动&&停止

数据库字符集设置

mysql 配置文件/etc/my.cnf 中加入 default-character-set=utf8

启动 mysql 服务:

```
service mysqld start 或者/etc/init.d/mysqld start
```

开机启动:

```
chkconfig -add mysqld, 查看开机启动设置是否成功 chkconfig --list | grep mysql*
```

mysqld 0:关闭 1:关闭 2:启用 3:启用 4:启用 5:启用 6:关闭

停止:

```
service mysqld stop
```

2、登录

创建 root 管理员:

```
mysqladmin -u root password 123456
```

www.2cto.com

登录:

```
mysql -u root -p 输入密码即可。
```

忘记密码:

```
service mysqld stop
```

```
mysqld_safe --user=root --skip-grant-tables
```

```
mysql -u root
```

```
use mysql
```

```
update user set password=password("new_pass") where user="root";
```

```
flush privileges;
```

3、远程访问

开放防火墙的端口号

mysql 增加权限: mysql 库中的 user 表新增一条记录 host 为“%”, user 为“root”。

4、Linux MySQL 的几个重要目录

www.2cto.com

数据库目录

`/var/lib/mysql/`

配置文件

`/usr/share /mysql` (mysql.server 命令及配置文件)

相关命令

`/usr/bin` (mysqladmin mysqldump 等命令)

启动脚本

`/etc/rc.d/init.d/` (启动脚本文件 mysql 的目录)

卸载 mysql

`yum -y remove mysql*`

如果是 rpm 安装的话:

`rpm -e mysql`

15、Linux 中 tomcat 部署 war 包

远程复制 **tomcat**

`scp /usr/lib/tomcat.zip root@120.24.0.124:/usr/lib`

部署 **dcxt war** 包

`cd /usr/lib/tomcat/tomcat-1/webapps/`

`rz -y`

`cp dcxt.war /usr/lib/tomcat/tomcat-2/webapps/`

`cd /usr/lib/tomcat/tomcat-2/webapps/`

`rz -y`

`vi /usr/lib/tomcat/tomcat-2/webapps/dcxt/WEB-INF/classes/struts-home.xml`

启动 **tomcat**

```
/usr/lib/tomcat/tomcat-1/bin/startup.sh
/usr/lib/tomcat/tomcat-2/bin/startup.sh
停止
/usr/lib/tomcat/tomcat-1/bin/shutdown.sh
/usr/lib/tomcat/tomcat-2/bin/shutdown.sh
ps -ef | grep tomcat
cd /usr/lib/tomcat/tomcat-1/webapps/dcxt/WEB-INF/classes/cn/hisilc/web/dao/impl
rz -y
cd /usr/lib/tomcat/tomcat-2/webapps/dcxt/WEB-INF/classes/cn/hisilc/web/dao/impl
rz -y
cd /usr/lib/tomcat/tomcat-1/webapps/dcxt/WEB-INF/classes/cn/hisilc/web/action/
rz -y
cd /usr/lib/tomcat/tomcat-2/webapps/dcxt/WEB-INF/classes/cn/hisilc/web/action/
rz -y

cd /usr/lib/tomcat/tomcat-1/webapps/dcxt/WEB-INF/classes/cn/hisilc/web/utils/
rz -y

cd /usr/lib/tomcat/tomcat-2/webapps/dcxt/WEB-INF/classes/cn/hisilc/web/utils/
rz -y

cd /usr/lib/tomcat/tomcat-1/webapps/dcxt/WEB-INF/classes/cn/hisilc/web/domain/
rz -y
cd /usr/lib/tomcat/tomcat-2/webapps/dcxt/WEB-INF/classes/cn/hisilc/web/domain/
rz -y

cd /usr/lib/tomcat/tomcat-1/webapps/dcxt/WEB-INF/classes/
rz -y
cd /usr/lib/tomcat/tomcat-2/webapps/dcxt/WEB-INF/classes/
rz -y

cd /usr/lib/tomcat/tomcat-1/conf/
rz -y
cd /usr/lib/tomcat/tomcat-2/conf/
rz -y
cd /usr/lib/tomcat/tomcat-2/webapps/dcxt/
启动 nginx
/usr/local/nginx/sbin/nginx
重启
/usr/local/nginx/sbin/nginx -s reload
```

修改配置文件

[vi /usr/lib/tomcat/tomcat-1/webapps/dcxt/WEB-INF/classes/beans.xml](#)

[vi /usr/lib/tomcat/tomcat-2/webapps/dcxt/WEB-INF/classes/beans.xml](#)

根服务器 war 包部署

```
cd /usr/lib/tomcat/tomcat-1/webapps/
```

```
rz -y
```

```
cd /usr/lib/tomcat/tomcat-2/webapps/
```

```
rz -y
```

```
cd /usr/lib/tomcat/tomcat-1/webapps/hisilc_server/WEB-INF/classes/
```

```
rz -y
```

```
cd /usr/lib/tomcat/tomcat-2/webapps/hisilc_server/WEB-INF/classes/
```

```
rz -y
```

```
cd /usr/lib/tomcat/tomcat-1/webapps/dcxt/WEB-INF/classes/cn/hisilc/hongbao/action/
```

```
rz -y
```

```
cd /usr/lib/tomcat/tomcat-2/webapps/dcxt/WEB-INF/classes/cn/hisilc/hongbao/action/
```

```
rz -y
```

查询并关闭进程

1. `ps -ef | grep tomcat` 查找 tomcat 的进程

2. 杀掉 tomcat 进程, 命令: `kill -9 进程名 (第三列)`

16、Linux 中 mysql 插入问题记录

根服务器 mysql 启动方式

```
/etc/init.d/mysql
```

```
service mysqld start
```

17、Linux 服务器被肉鸡如何解决？

限制远程登录 ip

一般有两种方法:

1、利用 linux 服务器防火墙 (操作系统自带的 iptables 即可), 仅允许指定 IP 访问 sshd 端口

2、利用 linux 服务器配置文件: `/etc/hosts.allow`, 加入行: (假定你的 ip

是 210.13.218.11)
sshd:210.13.218.11:allow

服务器被肉鸡是指开了 3389 端口微软终端服务（microsoftterminalservice），又有弱密码的高速服务器，俗称“肉鸡”。如果我们的服务器被肉鸡了怎么办呢？如何处理呢？下面我们就来详细的介绍下服务器被肉鸡的解决方法：

一、立即执行

1、更改系统管理员账户的密码，密码长度不小于 8 位并且使用大写字母/小写字母/数字/特殊字符组合；

2、更改远程登录端口并开启防火墙限制允许登录的 IP，防火墙配置只开放特定的服务端口并对 FTP、数据库等这些不需要对所有用户开放的服务进行源 IP 访问控制；

3、检查是否开放了未授权的端口

windows 在 CMD 命令行输入 netstat/ano，检查端口；有开放端口的根据 PID 检查进程，删除对应路径文件（根据 PID 检查进程步骤：开始-->运行-->输入“msinfo32”软件环境-->正在运行的任务）

linux 输入命令 netstat-anp 查看

4、安装专门的查杀病毒防木马软件，对服务器进行全盘病毒扫描和查杀；

5、假删除系统中未知账户，windows 系统还需要检查注册表中的 SAM 键值是否有隐藏账户；

6、假如有 WEB 服务的，限制 web 运行账户对文件系统的访问权限，只开放仅读权限。

二、后期防御

重点操作：开启云盾所有功能，特别是网站安全防护(自动防御所有 WEB 攻击)、网站后门检测（实时检测服务器上的后门程序）、主机密码破解防御

操作步骤：登录【云盾控制台】--【服务设置】进行开启

处理步骤：(重置系统--手工修改各个密码--开启云盾所有服务)

1、手工修改密码

密码长度不小于 8 位并且使用大写字母、小写字母、数字、特殊字符组合

至少包括:

- a.服务器登陆密码
- b.数据库连接密码
- c.网站后台密码
- d.FTP 密码
- e.其他服务器管理软件密码

2、系统加固

- a.到云盾控制台开启云盾所有服务，尤其是云盾网站安全防御（可以抵御黑客利用网站应用程序的漏洞入侵服务器，并且有专业的安全团队时刻关注国内安全动态，一旦发现新的漏洞出现，会立刻更新防护规则，防止黑客利用新漏洞入侵网站）
- b.建议站长把网站后台隐藏起来，尽量在保证网站正常运行的前提下，把网站后台目录名改的长长的。（比如：/mamashuomingziyaochangheikecaizhaobudao/）
- c.windows 系统要及时更新系统补丁

18、Ubuntu install rar

- 1、下载 <http://www.rarlab.com/download.htm>
- 2、安装 `tar zxvf rarlinux-3.8.0.tar.gz cd rar make make install`
- 3、rar 压缩 解压 将/etc 目录压缩为 etc.rar 命令为：`rar a etc.rar /etc` 将 etc.rar 解压 命令为：`rar x etc.rar unrar -e etc.tar`

19、Ubuntu install mysql timer

```
#timer shell
```

```
sudo vim /usr/sbin/backmysql
```

```
#add command
```

```
mysqldump -h"$(ifconfig eth0 | grep "inet addr" | cut -d ':' -f 2 | cut -d ' ' -f 1)" -uroot -p"@vpclubdev" --all-databases > /data/mysql/vpclub-"$(ifconfig eth0 | grep "inet addr" | cut -d ':' -f 2 | cut -d ' ' -f 1)"-"$(date +%Y-%m-%d_%H%M%S)".sql
```

```
#grant privileges  
chmod +x /usr/sbin/backmysql
```

```
sudo vim /etc/crontab
```

```
#add line
```

```
40 1 * * * root sudo /usr/sbin/backmysql
```

```
#重启 redis 解决占用内存过大问题  
cd /docker-init
```

```
sudo docker-compose restart redis
```

```
# 备份并恢复 sql
```

```
(导出)mysqldump -h172.16.45.3 -uroot -p -P3301 --all-databases > vpclub-back.sql
```

```
(导入)mysqldump -h120.76.240.113 -uroot -P3301 -p --all-databases < vpclub-back.sql
```

```
(导入)  
mysql -h172.16.5.23 -uroot -p < vpclub-172.16.5.23-2016-11-22_101358.sql
```

20、Ubuntu add user

#If you are signed in as the root user, you can create a new user at any time by typing:

```
adduser vpclub  
visudo  
root    ALL=(ALL:ALL) ALL  
vpclub  ALL=(ALL:ALL) ALL    # DELETE THIS LINE
```

```
passwd vpclub vpclub.pub
```

21、connect mysql

```
mysql -uroot -p"@vpclubdev" -h172.16.45.3 -P3301
```

#create database

```
create database test DEFAULT CHARSET=utf8
```

#create user

```
CREATE USER kfman@'localhost' IDENTIFIED BY 'Kfman123!';
```

#grant user

```
grant all privileges on *.* to kfman@'%' identified by 'Kfman123!' ;
```

#find mysql version

```
SHOW VARIABLES LIKE "%version%";
```

```
SELECT @@version;
```

#**backup and recovery** according to (<https://dev.mysql.com/doc/mysql-backup-excerpt/5.7/en/reloading-sql-format-dumps.html>)

22、查看系统版本,清除缓存,删除大文件及 du 使用

```
lsb_release -a
```

查询硬盘占用空间

```
sudo du -h -d 1
```

du 命令 实现 **Linux** 某个文件夹下的文件按大小排序

1. df -lh

2. du -s /usr/* | sort -rn
这是按字节排序

3. du -sh /usr/* | sort -rn

这是按兆（M）来排序

4. 选出排在前面的 10 个

```
du -s /usr/* | sort -rn | head
```

5. 选出排在后面的 10 个

```
du -s /usr/* | sort -rn | tail
```

```
du -h --max-depth=0 user
```

```
du -sh --max-depth=2 | more
```

#删除查询到的文件

```
find . -name "*.jar" | xargs rm -rf
```

cache 释放:

1. To free pagecache:

```
[rootmin@localhost~]#echo 1 > /proc/sys/vm/drop_caches
```

2. To free dentries and inodes:

```
[rootmin@localhost~]#echo 2 > /proc/sys/vm/drop_caches
```

3. To free pagecache, dentries and inodes:

```
[rootmin@localhost~]#echo 3 > /proc/sys/vm/drop_caches
```

说明，释放前最好 sync 一下，防止丢数据

#先切换到 root 用户

```
su
```

#同步内存数据到磁盘

sync

#清空内存 **cache**

```
echo 3 > /proc/sys/vm/drop_caches
```

23、文件夹授权

```
sudo chmod a+w -R deployment
```

授权 a 所有用户 w 写权限，-R 递归 deployment 文件夹下面

24、解决 python-pip install error

```
export LC_ALL=C

#卸载 docker-compose

#先升级 pip，然后卸载

pip install --upgrade pip

pip uninstall docker-compose

#再安装最新版本

pip install docker-compose
```

25、linux 新建分区，格式化硬盘，持久生效

```
fdisk -l

fdisk /dev/vdb

fdisk -l

mkfs -t ext4 /dev/vdb1

fdisk -l

mkdir /data

df -h

mount /dev/vdb1 /data

df -h

cd /data
```

#永久生效

vim /etc/fstab

#add line

#mount /dev/sdb1 to /data

/dev/sdb1 /data ext4 defaults 0 2

26、**Ubuntu** 开启防火墙

```
sudo ufw enable
```

```
sudo ufw default deny
```

```
sudo ufw status
```

```
sudo ufw allow 80 允许外部访问 80 端口
```

```
sudo ufw delete allow 80 禁止外部访问 80 端口
```

```
sudo ufw allow from 192.168.1.1 允许此 IP 访问所有的本机端口
```

```
sudo ufw deny smtp 禁止外部访问 smtp 服务
```

```
sudo ufw delete allow smtp 删除上面建立的某条规则
```

```
sudo ufw deny proto tcp from 10.0.0.0/8 to 192.168.0.1 port 22 要拒绝所有的 TCP 流量从 10.0.0.0/8 到 192.168.0.1 地址的 22 端口
```

可以允许所有 RFC1918 网络（局域网/无线局域网的）访问这个主机（/8,/16,/12 是一种网络分级）：

```
sudo ufw allow from 10.0.0.0/8
```

```
sudo ufw allow from 172.16.0.0/12
```

```
sudo ufw allow from 192.168.0.0/16
```


27、**sudo: parse error in /etc/sudoers near line 24 ...报错**

```
pkexec visudo
```

28、**How to safely abort apt-get install?**

#起因,我想安装 sudo apt-get install oracle-java8-installer, 结果发现下载太慢! 想终止!

```
sudo dpkg --configure -a
```

```
sudo dpkg -r oracle-java8-installer
```

29、**How to add interface in ubuntu?**

```
sudo vim /etc/network/interfaces
```

#add line

```
auto lo

iface lo inet loopback

#auto eth0

# The primary network interface

auto eth1

# iface eth0 inet dhcp

iface eth1 inet static

    address 192.168.1.23

    netmask 255.255.255.0

    gateway 192.168.1.1
```

```
    dns-nameservers 192.168.1.1 223.5.5.5

auto eth0

iface eth0 inet static

    address 172.16.5.23

    netmask 255.255.0.0

    gateway 172.16.0.1

    dns-nameservers 172.16.0.1 223.5.5.5

    broadcast 255.255.255.255

    network 172.16.0.0
```

30、Global search file in linux

#输入想要搜索的 contend

```
grep -rn content
```

31、gitlab 服务器文件移动时，修改 git url

```
git remote set-url origin git@gitlab.vpclub:common/vp-common-config.git
```

#初始化 git

```
git initre
```

#链接远程仓库

```
git remote add origin https://github.com/RobinsChens/kubernetes.git
```

32、idea 2016.1 注册服务器

<http://idea.qinxi1992.cn/>

33、ubuntu 安装 notepad++

参考: <https://itsfoss.com/notepad-alternatives-for-linux/>

```
sudo add-apt-repository ppa:notepadqq-team/notepadqq  
  
sudo apt-get update  
sudo apt-get install notepadqq
```

34、linux 建立快捷方式

1.软连接

ln -s source dist

2.硬链接

ln source dist

3.删除链接 CREATE USER kfman@'%' IDENTIFIED BY 'Kfman123!';

unlink dist(加链接的文件或文件夹)

35、ubuntu 开放 root 并且免密

you want to login **AS** root via SSH so that you can run an update script.

You need to add your key to the root authorized_keys file on the **server**.
On the Server

1. sudo su - root
2. apt-get --assume-yes install openssh-client
3. ssh-keygen
4. vim .ssh/authorized_keys

5. Check your `sshd_config` settings (`/etc/ssh/sshd_config`) are correct one last time.

```
PermitRootLogin without-password
```

```
RSAAuthentication yes
```

```
PubkeyAuthentication yes
```

6. Ensure those settings take effect by restarting `ssh`: `service ssh restart`

On the Client

```
vim .ssh/config
```

```
#add host config
```

```
Host gitlab.vpclub
```

```
User git
```

```
Hostname 172.16.5.22
```

```
Port 22022
```

```
IdentityFile ~/.ssh/id_rsa
```

36、**ubuntu** 配置国内免费 registry mirror

由于国内特殊的网络环境，往往我们从 **Docker** Hub 中拉取镜像并不能成功，而且速度特别慢。

那么我们可以给 Docker 配置一个国内的 registry mirror，当我们需要的镜像在 mirror 中则直接返回，如果没有则从 Docker Hub 中拉取。是否使用 registry mirror 对 Docker 用户来说是透明的。

DaoCloud 在国内提供了首个 Docker Hub 镜像服务，而且免费，大大提高了国内 Docker 用户的使用热情，非常感谢 DaoCloud。

使用方法

修改 Docker 配置文件 `/etc/default/docker` 如下：

```
DOCKER_OPTS="--registry-mirror=http://aad0405c.m.daocloud.io"
```

使用 `service docker restart` 重启 Docker 服务即可。

37、**vim** 编辑替换

```
%s/-//g
```

```
#获取最后 6 位字符
```

```
echo "abcdefghijklmn" | tail -c 6
```

#删除当前光标向后的所有字符

```
d$
```

38、安装 go 以及 godep

```
wget https://storage.googleapis.com/golang/go1.5.1.linux-amd64.tar.gz
```

```
sudo tar -C /usr/local -xzf go1.5.1.linux-amd64.tar.gz
```

```
export GOROOT=/usr/local/go
```

```
export PATH=$PATH:$GOROOT/bin
```

```
export GOPATH=/root/k8s-cluster/github/gohome
```

```
#install godep
```

```
export GOBIN=$GOPATH/bin
```

```
export PATH=$GOPATH:$GOBIN:$PATH
```

```
go version
```

```
godep version
```

39、K8S-openshift 安装

参考 (<https://seanzhau.com/>)

40、十条命令在一分钟内检查 Linux 服务器性能

通过执行以下命令，可以在 1 分钟内对系统资源使用情况有个大致的了解。

- uptime

- dmesg | tail

- vmstat 1

- mpstat -P ALL 1

- pidstat 1

- iostat -xz 1

- free -m
- sar -n DEV 1
- sar -n TCP,ETCP 1
- top

41、ubuntu 系统升级

参考(<https://bbs.aliyun.com/read/289870.html>)

1. 更新软件列表：

[复制代码](#)

1. apt-get update

2. 升级软件：

[复制代码](#)

1. apt-get upgrade

3. 升级当前系统版本：

[复制代码](#)

1. apt-get dist-upgrade

4. 升级到新系统版本：

[复制代码](#)

1. do-release-upgrade -d

42、ubuntu 静态网卡配置

```
# interfaces(5) file used by ifup(8) and ifdown(8)
```

```
auto lo
```

```
iface lo inet loopback
```

```
auto eth1
```

```
# iface eth0 inet dhcp

iface eth1 inet static

    address 192.168.1.23

    netmask 255.255.255.0

    gateway 192.168.1.1

    dns-nameservers 192.168.1.1 223.5.5.5 8.8.4.4 8.8.8.8

auto eth0

iface eth0 inet static

    address 172.16.5.23

    netmask 255.255.0.0

    gateway 172.16.0.1

    dns-nameservers 172.16.0.1 223.5.5.5 8.8.4.4 8.8.8.8

    broadcast 255.255.255.255

    network 172.16.0.0
```

43、ubuntu 设置开机启动

```
#创建或者移动 数据存储路径 docker-storage 到/目录

cd /docker-storage

#使用上面 docker-storage 文件夹

###Starting Docker Storage as a Service
```

```
sudo chown -R root: /docker-storage

sudo nano /etc/init/docker-storage.conf

###copy words below into /etc/init/docker-storage.conf

    description "Docker init"
    start on runlevel [2345]
    stop on runlevel [016]
    respawn
    respawn limit 10 5

    chdir /docker-storage

    exec /usr/local/bin/docker-compose up
    #

# docker-storage service start
sudo service docker-storage start

#look at the current docker process
sudo docker ps
```

44、redis 设置密码

1.通过配置文件进行配置

yum 方式安装的 redis 配置文件通常在/etc/redis.conf 中，打开配置文件找到

```
#requirepass foobared
```

去掉行前的注释，并修改密码为所需的密码,保存文件

```
requirepass myRedis
```

重启 redis

```
sudo service redis restart
```

```
#或者
```

```
sudo service redis stop
```

```
sudo redis-server /etc/redis.conf
```

这个时候尝试登录 redis，发现可以登上，但是执行具体命令是提示操作不允许

```
redis-cli -h 127.0.0.1 -p 6379
```

```
redis 127.0.0.1:6379>
```



```

redis 127.0.0.1:6379> keys *
(error) ERR operation not permitted
redis 127.0.0.1:6379> select 1
(error) ERR operation not permitted
redis 127.0.0.1:6379[1]>
尝试用密码登录并执行具体的命令看到可以成功执行
redis-cli -h 127.0.0.1 -p 6379 -a myRedis
redis 127.0.0.1:6379> keys *
1) "myset"
2) "mysortset"
redis 127.0.0.1:6379> select 1
OK
redis 127.0.0.1:6379[1]> config get requirepass
1) "requirepass"
2) "myRedis"

```

2.通过命令行进行配置

```

redis 127.0.0.1:6379[1]> config set requirepass my_redis
OK
redis 127.0.0.1:6379[1]> config get requirepass
1) "requirepass"
2) "my_redis"

```

无需重启 redis

使用第一步中配置文件中配置的老密码登录 redis，会发现原来的密码已不可用，操作被拒绝

```

1
2
3

```

```

redis-cli -h 127.0.0.1 -p 6379 -a myRedis
redis 127.0.0.1:6379> config get requirepass
(error) ERR operation not permitted

```

使用修改后的密码登录 redis，可以执行相应操作

```

?
1
2
3
4

```

```

redis-cli -h 127.0.0.1 -p 6379 -a my_redis
redis 127.0.0.1:6379> config get requirepass
1) "requirepass"
2) "my_redis"

```

尝试重启一下 redis，用新配置的密码登录 redis 执行操作，发现新的密码失效，redis 重新使用了配置文件中的密码

```
sudo service redis restart
Stopping redis-server: [ OK ]
Starting redis-server: [ OK ]
redis-cli -h 127.0.0.1 -p 6379 -a my_redis
redis 127.0.0.1:6379> config get requirepass
(error) ERR operation not permitted
redis-cli -h 127.0.0.1 -p 6379 -a myRedis
redis 127.0.0.1:6379> config get requirepass
1) "requirepass"
2) "myRedis"
```

除了在登录时通过 -a 参数制定密码外，还可以登录时不指定密码，而在执行操作前进行认证？

```
redis-cli -h 127.0.0.1 -p 6379
redis 127.0.0.1:6379> config get requirepass
(error) ERR operation not permitted
redis 127.0.0.1:6379> auth myRedis
OK
redis 127.0.0.1:6379> config get requirepass
1) "requirepass"
2) "myRedis"
```

3.master 配置了密码，slave 如何配置

若 master 配置了密码则 slave 也要配置相应的密码参数否则无法进行正常复制的。

slave 中配置文件内找到如下行，移除注释，修改密码即可

```
#masterauth mstpassword
```

```
1redis-cli -h 127.0.0.1 -p 6379 -a myRedis
2redis 127.0.0.1:6379> config get
3requirepass
```

(error) ERR operation not permitted

使用修改后的密码登录 redis，可以执行相应操作

?

```
1 redis-cli -h 127.0.0.1 -p 6379 -a my_redis
2 redis 127.0.0.1:6379> config get
3 requirepass
4 1) "requirepass"
5 2) "my_redis"
```

尝试重启一下 redis，用新配置的密码登录 redis 执行操作，发现新的密码失效，redis 重新使用了配置文件中的密码

?

```
sudo service redis restart
1 Stopping redis-server: [
2 OK ]
3 Starting redis-server: [
4 OK ]
5 redis-cli -h 127.0.0.1 -p 6379 -a my_redis
6 redis 127.0.0.1:6379> config get requirepass
7 (error) ERR operation not permitted
8 redis-cli -h 127.0.0.1 -p 6379 -a myRedis
9 redis 127.0.0.1:6379> config get requirepass
10 1) "requirepass"
11 2) "myRedis"
```

除了在登录时通过 -a 参数制定密码外，还可以登录时不指定密码，而在执行操作前进行认证。

?

```
redis-cli -h 127.0.0.1 -p 6379
1 redis 127.0.0.1:6379> config get
2 requirepass
3 (error) ERR operation not permitted
4 redis 127.0.0.1:6379> auth myRedis
5 OK
6 redis 127.0.0.1:6379> config get
7 requirepass
8 1) "requirepass"
9 2) "myRedis"
```

3.master 配置了密码，slave 如何配置

若 master 配置了密码则 slave 也要配置相应的密码参数否则无法进行正常复制的。

slave 中配置文件内找到如下行，移除注释，修改密码即可

?

1#masterauth mstpassword

45、docker error could not delete the default bridge network: network bridge has active endpoints

And the error is:

"FATA[0000] Error starting daemon: Error initializing network controller: could not delete the default bridge network: network bridge has active endpoints "

The solution is:

```
sudo rm -r /var/lib/docker/network
```

46、Ubuntu 环境下挂载新硬盘

(参考)

①、硬盘分区 | Hard disk add new partition

1、显示硬盘及所属分区情况。在终端窗口中输入如下命令:

sudo fdisk -lu

显示当前的硬盘及所属分区的情况。如下图所示:

系统提示: Disk /dev/sdb doesn't contain a valid partition table。

```
aofeng@aofeng-vb:~$ sudo fdisk -lu
```

Disk /dev/sda: 21.5 GB, 21474836480 bytes
 255 heads, 63 sectors/track, 2610 cylinders, total 41943040 sectors
 Units = sectors of 1 * 512 = 512 bytes
 Sector size (logical/physical): 512 bytes / 512 bytes
 I/O size (minimum/optimal): 512 bytes / 512 bytes
 Disk identifier: 0x000820cb

| Device | Boot | Start | End | Blocks | Id | System |
|--|------|----------|----------|----------|----|----------------------|
| /dev/sda1 | * | 2048 | 499711 | 248832 | 83 | Linux |
| Partition 1 does not end on cylinder boundary. | | | | | | |
| /dev/sda2 | | 501758 | 41940991 | 20719617 | 5 | Extended |
| /dev/sda5 | | 501760 | 2500607 | 999424 | 82 | Linux swap / Solaris |
| /dev/sda6 | | 2502656 | 12500991 | 4999168 | 83 | Linux |
| /dev/sda7 | | 12503040 | 41940991 | 14718976 | 83 | Linux |

Disk /dev/sdb: 214.7 GB, 214748364800 bytes
 255 heads, 63 sectors/track, 26108 cylinders, total 419430400 sectors
 Units = sectors of 1 * 512 = 512 bytes
 Sector size (logical/physical): 512 bytes / 512 bytes
 I/O size (minimum/optimal): 512 bytes / 512 bytes
 Disk identifier: 0x00000000

Disk /dev/sdb doesn't contain a valid partition table

2、对硬盘进行分区。在终端窗口中输入如下命令:

sudo fdisk /dev/sdb

如下图所示:

在 Command (m for help)提示符后面输入 m 显示一个帮助菜单。

在 Command (m for help)提示符后面输入 n, 执行 add a new partition 指令给硬盘增加一个新分区。

```
c  toggle the dos compatibility flag
d  delete a partition
l  list known partition types
m  print this menu
n  add a new partition
o  create a new empty DOS partition table
p  print the partition table
q  quit without saving changes
s  create a new empty Sun disklabel
t  change a partition's system id
u  change display/entry units
v  verify the partition table
w  write table to disk and exit
x  extra functionality (experts only)

Command (m for help): n
Command action
  e  extended
  p  primary partition (1-4)
e
Partition number (1-4): 1
First cylinder (1-26108, default 1): 1
Last cylinder, +cylinders or +size{K,M,G} (1-26108, default 26108):
Using default value 26108
```

在 Command (m for help)提示符后面输入 p，显示分区表。

系统提示如下：

Device Boot

/dev/sdb1

在 Command (m for help)提示符后面输入 w，保存分区表。

系统提示：The partition table has been altered!

```
Command (m for help): m
Command action
  a  toggle a bootable flag
  b  edit bsd disklabel
  c  toggle the dos compatibility flag
  d  delete a partition
  l  list known partition types
  m  print this menu
  n  add a new partition
  o  create a new empty DOS partition table
  p  print the partition table
  q  quit without saving changes
  s  create a new empty Sun disklabel
  t  change a partition's system id
  u  change display/entry units
  v  verify the partition table
  w  write table to disk and exit
  x  extra functionality (experts only)

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.
```

在终端窗口中输入如下命令：

```
sudo fdisk -lu
```

如下图所示：

系统已经识别了硬盘 /dev/sdb 的分区。


```

Disk /dev/sda: 21.5 GB, 21474836480 bytes
255 heads, 63 sectors/track, 2610 cylinders, total 41943040 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000820cb

```

| Device | Boot | Start | End | Blocks | Id | System |
|--|------|----------|----------|----------|----|----------------------|
| /dev/sda1 | * | 2048 | 499711 | 248832 | 83 | Linux |
| Partition 1 does not end on cylinder boundary. | | | | | | |
| /dev/sda2 | | 501758 | 41940991 | 20719617 | 5 | Extended |
| /dev/sda5 | | 501760 | 2500607 | 999424 | 82 | Linux swap / Solaris |
| /dev/sda6 | | 2502656 | 12500991 | 4999168 | 83 | Linux |
| /dev/sda7 | | 12503040 | 41940991 | 14718976 | 83 | Linux |

```

Disk /dev/sdb: 214.7 GB, 214748364800 bytes
255 heads, 63 sectors/track, 26108 cylinders, total 419430400 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x618837ef

```

| Device | Boot | Start | End | Blocks | Id | System |
|-----------|------|-------|-----------|------------|----|----------|
| /dev/sdb1 | | 63 | 419425019 | 209712478+ | 5 | Extended |

```

l  list known partition types
m  print this menu
n  add a new partition
o  create a new empty DOS partition table
p  print the partition table
q  quit without saving changes
s  create a new empty Sun disklabel
t  change a partition's system id
u  change display/entry units
v  verify the partition table
w  write table to disk and exit
x  extra functionality (experts only)

```

Command (m for help): p

```

Disk /dev/sdb: 214.7 GB, 214748364800 bytes
255 heads, 63 sectors/track, 26108 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x618837ef

```

| Device | Boot | Start | End | Blocks | Id | System |
|-----------|------|-------|-------|------------|----|----------|
| /dev/sdb1 | | 1 | 26108 | 209712478+ | 5 | Extended |


```
aofeng@aofeng-vb:~$ sudo fdisk /dev/sdb
Device contains neither a valid DOS partition table, nor Sun, SGI or OSF disklabel
Building a new DOS disklabel with disk identifier 0xa20a7e16.
Changes will remain in memory only, until you decide to write them.
After that, of course, the previous content won't be recoverable.

Warning: invalid flag 0x0000 of partition table 4 will be corrected by w(rite)

WARNING: DOS-compatible mode is deprecated. It's strongly recommended to
switch off the mode (command 'c') and change display units to
sectors (command 'u').

Command (m for help): m
Command action
 a toggle a bootable flag
 b edit bsd disklabel
 c toggle the dos compatibility flag
 d delete a partition
 l list known partition types
 m print this menu
 n add a new partition
 o create a new empty DOS partition table
 p print the partition table
 q quit without saving changes
```

②、硬盘格式化 | Format hard disk

1、显示硬盘及所属分区情况。在终端窗口中输入如下命令：

sudo mkfs -t ext4 /dev/sdb

说明：

-t ext4 表示将分区格式化成为 ext4 文件系统类型。

```
aofeng@aofeng-vb:~$ sudo mkfs -t ext4 /dev/sdb
[sudo] password for aofeng:
mke2fs 1.41.12 (17-May-2010)
/dev/sdb is entire device, not just one partition!
无论如何也要继续? (y,n) y
文件系统标签=
操作系统:Linux
块大小=4096 (log=2)
分块大小=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
13107200 inodes, 52428800 blocks
2621440 blocks (5.00%) reserved for the super user
第一个数据块=0
Maximum filesystem blocks=0
1600 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000, 7962624, 11239424, 20480000, 23887872

正在写入inode表: 完成
Creating journal (32768 blocks): 完成
Writing superblocks and filesystem accounting information: 完成
```

③、挂载硬盘分区 | Mount hard disk partition

1、显示硬盘挂载情况。在终端窗口中输入如下命令：

```
sudo df -l
```

新硬盘分区没有挂载，无法进入和查看。

在终端窗口中输入如下命令：

```
sudo mount -t ext4 /dev/sdb /devdata
```

说明：

指定硬盘分区文件系统类型为 ext4 ，同时将 /dev/sdb 分区挂载到目录 /devdata。

再次在终端窗口中输入如下命令：

```
sudo df -l
```

新硬盘分区已经挂载，如下图最下面的红色方框内容。

```

aofeng@aofeng-vb:~$ df -l
文件系统      1K-块      已用    可用  已用% 挂载点
/dev/sda6      4920636    3070444    1600236    66% /
none           507252      244     507008      1% /dev
none           512852      252     512600      1% /dev/shm
none           512852      360     512492      1% /var/run
none           512852       0     512852      0% /var/lock
/dev/sda1      240972      59689     168842     27% /boot
/dev/sda7     14487924    216896    13535080      2% /home
/dev/sr0        32370      32370       0    100% /media/VBOXADDITIONS_3.2
.12_68302
aofeng@aofeng-vb:~$ sudo mount -t ext4 /dev/sdb /devdata
[sudo] password for aofeng:
aofeng@aofeng-vb:~$ df -l
文件系统      1K-块      已用    可用  已用% 挂载点
/dev/sda6      4920636    3070444    1600236    66% /
none           507252      244     507008      1% /dev
none           512852      252     512600      1% /dev/shm
none           512852      360     512492      1% /var/run
none           512852       0     512852      0% /var/lock
/dev/sda1      240972      59689     168842     27% /boot
/dev/sda7     14487924    216896    13535080      2% /home
/dev/sr0        32370      32370       0    100% /media/VBOXADDITIONS_3.2
.12_68302
/dev/sdb       206424760    191756    195747244      1% /devdata

```

2、配置硬盘在系统启动自动挂载。在文件 `/etc/fstab` 中加入如下配置：

```
# /devdata was on /dev/sdb
```

```
UUID=37eaa526-5d96-4237-8468-603df5216ce9
```

```

# /etc/fstab: static file system information.
#
# Use 'blkid -o value -s UUID' to print the universally unique identifier
# for a device; this may be used with UUID= as a more robust way to name
# devices that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point>    <type>  <options>          <dump>  <pass>
proc           /proc                proc     nodev,noexec,nosuid 0          0
# / was on /dev/sda6 during installation
UUID=0b32eada-f81d-4d8e-874f-0af81b7e46ef /                ext4      errors=remount-ro 0
# /boot was on /dev/sda1 during installation
UUID=072fe42e-3456-4edf-82b6-35324a43f8a9 /boot            ext4      defaults          0
# /home was on /dev/sda7 during installation
UUID=4700cc40-8f81-440a-90fa-8d0d81ef79e7 /home            ext4      defaults          0
# swap was on /dev/sda5 during installation
UUID=a5866d8b-945b-4152-bf7e-84d9aa29da77 none             swap      sw                0
# /devdata wan on /dev/sdb
UUID=37eaa526-5d96-4237-8468-603df5216ce9 /devdata         ext4      defaults          0

```

===== `/etc/fstab` 配置 注意！！ =====

这一部分我的设置与原作者不同，用上面无法成功自动挂载的话，请尝试下面的配置：

```
fstab x
1 # /etc/fstab: static file system information.
2 #
3 # Use 'blkid -o value -s UUID' to print the universally unique identifier
4 # for a device; this may be used with UUID= as a more robust way to name
5 # devices that works even if disks are added and removed. See fstab(5).
6 #
7 # <file system> <mount point> <type> <options> <dump> <pass>
8 proc /proc proc nodev,noexec,nosuid
9 # / was on /dev/sda5 during installation
10 UUID=ff5251af-4286-4839-8612-86f4bce7269e / ext4 errors=remount-ro
11 # swap was on /dev/sda6 during installation
12 UUID=e093adfe-aa5c-43da-b23d-6464142ae8e6 none swap sw
13 /dev/sdb /Projects ext4 defaults
14 /dev/sdd /AMAX ext4 defaults
```

其中，“/Projects”与“/AMAX”都是挂载点，可以根据需要自定义。
上面是 ext4 格式硬盘为例子。如果是 ntfs，则“ext4”要修改为“auto”，“defaults”要修改为“nosuid,nodev,nofail”。

在 Ubuntu，可以使用更直观的方式：用系统自带的 Disks 或者 Disk Utility 界面化工具。

=====

=====

附录 1：fdisk 命令详解 | Appendix part 1:

fdisk 命令的语法如下：

fdisk [-b sectorsize] device

fdisk -l [-u] [device...]

fdisk -s partition...

fdisk -v

说明：

- b
- l
- u
- s
- v

附录 2：mkfs 命令详解 | Appendix part 2:

mkfs 命令的语法如下：

mkfs [-V] [-t fstype] [fs-options] filesys

说明：

- V
- t
- fs
- v

附录 3：mount 命令详解 | Appendix part 3:

mkfs 命令的语法如下：

```
mount [-afFnrsvw] [-t vfstype] [-L label] [-o options] device dir
mount [-l hv]
```

说明：

- a
- f
- F
- t vfstype
- L label
- l
- h
- v
- device
- dir

附录 4：fstab 配置详解 | Appendix part 4:

/etc/fstab 中一共有 6 列：

- file system：指定要挂载的文件系统的设备名称（如：/dev/sdb）。也可以采用 UUID，UUID 可以通过使用 blkid 命令来查看（如：blkid
- mount point：挂载点。就是自己手动创建一个目录，然后把分区挂载到这个目录下。
- type：用来指定文件系统的类型。如：ext3, ext4, ntfs 等。
- option dump：0 表示不备份；1 表示要将整个中的内容备份。此处建议设置为 0。
- pass：用来指定 fsck 如何来检查硬盘。0 表示不检查；挂载点为分区／（根分区）必须设置为 1，其他的挂载点不能设置为 1；如果有挂载 ass 设置成大于 1 的值，则在检查完根分区后，然后按 pass 的值从小到大依次检查，相同数值的同时检查。如：/home 和 /boot 的 pass 设置成 2，/devdata 的 pass 设置成 3，则系统在检查完根分区，接着同时检查/boot 和/home，再检查/devdata。

47、centos 7 自定义开机启动

服务脚本目录： /usr/lib/systemd/

有系统（system）和用户（user）之分，开机启动存在系统服务（system）里，即：
/lib/systemd/system/，用户登录后才能运行的程序，存在用户服务（user）里，
以.service 结尾。

nginx 开机启动距离

建立服务文件

```
[root@seanzhau.com ~]# vim /lib/systemd/system/nginx.service [Unit]
Description=nginx After=network.target [Service] Type=forking
ExecStart=/www/lanmps/init.d/nginx start
ExecReload=/www/lanmps/init.d/nginx restart
ExecStop=/www/lanmps/init.d/nginx stop PrivateTmp=true [Install]
WantedBy=multi-user.target # 参数说明[Unit]:服务的说明 Description:描述服务
After:描述服务类别[Service]服务运行参数的设置 Type=forking 是后台运行的形式
ExecStart 为服务的具体运行命令 ExecReload 为重启命令 ExecStop 为停止命令
PrivateTmp=True 表示给服务分配独立的临时空间注意：[Service]的启动、重启、停止命
令全部要求使用绝对路径[Install]服务安装的相关设置，可设置为多用户
```

以 754 的权限保存在目录： /lib/systemd/system

设置开机自启动

```
[root@seanzhau.com ~]# systemctl enable nginx.service
```

其他命令

| 任务 | 旧指令 | 新指令 |
|---------|-------------------------------|---|
| 自动启动 | chkconfig --level 3 httpd on | systemctl enable httpd.service |
| 关闭不自动启动 | chkconfig --level 3 httpd off | systemctl disable httpd.service |
| 检查服务状态 | service httpd status | systemctl status httpd.service（服务详细信息） systemctl is-active httpd.service（仅显示是否 Active） |

| 任务 | 旧指令 | 新指令 |
|------------|-----------------------|------------------------------------|
| 显示所有已启动的服务 | chkconfig -list | systemctl list-units -type=service |
| 启动某服务 | service httpd start | systemctl start httpd.service |
| 停止某服务 | service httpd stop | systemctl stop httpd.service |
| 重启某服务 | service httpd restart | systemctl restart httpd.service |

48、文件授权

chmod u=rwx,g=rwx,o=rwx filename

49、Ubuntu-16.04 安装搜狗拼音输入法

最新的 Ubuntu-16.04 已经在 4 月 21 日正式发布，本人在前两天进行了安装。新的特性还没来得及去了解，只是安装之后，发现搜狗输入法不能正常的安装。经过搜索与尝试，提供下相关的解决方法。

1. 相关软件

Linux 版搜狗拼音输入法下载地址：

<http://pinyin.sogou.com/linux/>

fonts-droid 下载地址:

http://ftp.cn.debian.org/debian/pool/main/f/fonts-android/fonts-droid_4.4.4r2-6_all.deb

2. 开始安装

```
# 查看依赖关系
sudo dpkg -i sogoupinyin 2.0.0.0068 amd64.deb
# 解决大部分依赖
sudo apt-get -f install
# 正常情况下会提示找不到 fonts-droid , 下面进行安装
sudo dpkg -i fonts-droid_4.4.4r2-6_all.deb
# 安装搜狗输入法
sudo dpkg -i sogoupinyin 2.0.0.0072 amd64.deb
```

如果没什么其他问题, 就已经安装成功。注销后再登陆即可切换搜狗输入法。

如果你安装的是 ubuntu 14.04 LTS 版本, 那么你直接双击下载的搜狗输入法是不行的。

这时候还需要提供如下操作

1. 安装 fcitx 输入法框架--sudo apt-get install fcitx-bin

注意: 这时候可能会提示你安装失败, 你需要跟新一个系统 sudo apt-get install
update;sudo apt-get install upgrade

2. 选择 fcitx 框架

System settings->Language Support->Language->keybard input method
system(fcitx)->右上角输入法->添加当前输入法->选择已经安装的输入法即可

50、docker delete exited container

```
docker ps -a | grep Exited | awk '{print $1}' | xargs docker rm -f
```


51、Ubuntu 安装 QQ

Download from <http://www.ubuntukeylin.com/application/show.php?lang=cn&id=279>

解压后安装

```
cd wine-qqintl
```

```
sudo dpkg -i fonts-wqy-microhei_0.2.0-beta-2_all.deb ttf-wqy-microhei_0.2.0-beta-2_all.deb
```

```
sudo apt-get install -f
```

```
sudo dpkg -i wine-qqintl_0.1.3-2_i386.deb
```

#记得手机 qq 上面需要关闭设备锁，不然提示 qq 不能登录

52、Ubuntu 安装 LibreOffice Project

Download

from

https://sourceforge.net/projects/projectlibre/files/ProjectLibre/1.6.2/projectlibre_1.6.2-1.deb/download

解压后安装

#防止有 dpkg 线程阻塞

```
sudo dpkg --configure -a
```

```
sudo dpkg -i projectlibre-1.6.2-1.deb
```

53、Ubuntu 安装 vokoscreen 录屏软件

参考 <http://www.mintos.org/skill/vokoscreen-capture.html>

54、Ubuntu 创建 idea 执行快捷方式

```
sudo ln -s /home/vpclub/idea-8/bin/idea.sh /usr/local/bin/idea
```

55、Docker-compose 启动报错解决(pip install docker-compose)

```
1.[root@cache1 www]# docker-compose
```

```
Traceback (most recent call last):
```

```
File "/usr/bin/docker-compose", line 5, in <module>
```

```
from pkg_resources import load_entry_point
```

```
File "/usr/lib/python2.7/site-packages/pkg_resources.py", line 3011, in  
<module>
```

```
parse_requirements(__requires__), Environment()  
File "/usr/lib/python2.7/site-packages/pkg_resources.py", line 626, in  
resolve  
raise DistributionNotFound(req)  
pkg_resources.DistributionNotFound: backports.ssl-match-hostname>=3.5
```

2.pip install --upgrade pip # 先升级下 pip

3.pip install docker-compose

4.# 执行如下命令

5.docker-compose --help

6.# 报如下错误:

7.Traceback (most recent call last):

8.File "/usr/bin/docker-compose", line 5, in <module>

9.from pkg_resources import load_entry_point

10.File "/usr/lib/python2.7/site-packages/pkg_resources.py", line 3011, in <module>

11.parse_requirements(__requires__), Environment()

12.File "/usr/lib/python2.7/site-packages/pkg_resources.py", line 626, in resolve

13.raise DistributionNotFound(req)

14.pkg_resources.DistributionNotFound: backports.ssl-match-hostname>=3.5

15.# 解决方法:

16.pip install --upgrade backports.ssl_match_hostname