Contents

[Install Linux 1](#_Toc454972092)

[linux shell脚本 1](#_Toc454972093)

[apt-get 1](#_Toc454972094)

[python使用 2](#_Toc454972095)

# Install Linux

usb+\*.iso 安装ubuntu, … 见ubuntu网站

// 解决ubuntu卡，原因是virtualBox资源分配

关闭ubuntu

控制->设置->系统->内存大小，处理器数量，硬件加速，

控制->设置->显示->显卡->3D,2D加速

打开终端 Ctrl+Alt+T

Ubuntu桌面入门指南

https://help.ubuntu.com/

<http://wiki.ubuntu.org.cn/Ubuntu%E6%A1%8C%E9%9D%A2%E5%85%A5%E9%97%A8%E6%8C%87%E5%8D%97>

桌面 -> 控制台 Ctrl + Alt + F1 (F2, …F6)

控制台 –> 桌面 Ctrl + Alt + F7

Switch between window Alt + Tab

# linux shell脚本

## apt-get

从网上下载包，并安装到本地

手工下载 dpkg 包，而后 ”dpkg -i 包名“ 也可以,但找包，检查dependency 太辛苦，这就是 apt-get 帮助做的事情. 从哪里下载, 官方url, 在/etc/apt/sources.list中，可以看看

$sudo vi /etc/apt/sources.list, sources list 中的格式如下

deb http://site.example.com/debian distribution component1 component2 component3

deb-src http://site.example.com/debian distribution component1 component2 component3

开头 deb 代表编译好的包， des-src代表源代码

distribution 代表发行版本，比如 12.07 是 precise

更改了 /etc/apt/sources.list ， 要用用 apt-get update 更新 index 文件, 更新的 index 文件在 /var/lib/apt/lists/

安装 ppa

PPA，表示 Personal Package Archives，也就是个人软件包集。

有很多软件因为种种原因，不能进入官方的 Ubuntu 软件仓库。 为了方便 Ubuntu 用户使用，launchpad.net 提供了 ppa，允许用户建立自己的软件仓库， 自由的上传软件。PPA 也被用来对一些打算进入 Ubuntu 官方仓库的软件，或者某些软件的新版本进行测试。

sudo add-apt-repository ppa:user/ppa-name

查询已经安装的package , 使用 dpkg -S 命令

apt-get -y install 这个指令则是跳过系统提示，直接安装。

apt-get install packagename //安装一个新软件包（参见下文的aptitude）

apt-get remove packagename //卸载一个已安装的软件包（保留配置文档）

apt-get remove --purge packagename //卸载一个已安装的软件包（删除配置文档）

apt-get autoremove packagename //删除包及其依赖的软件包

apt-get autoremove --purge packagname //删除包及其依赖的软件包+配置文件，比上面的要删除的彻底一点

apt-get update //更新软件列表。

apt-get upgrade //更新软件

apt-get dist-upgrade //更新系统版本

命令帮助

apt-get --help

文件系统操作

$ mkdir -p src/main/scala // create directory

$ mv src dst // rename directory

$ rm -rf directoryName // delete directory

$ touch build.gradle // create file

$ rm filename // delete files

$ unlink /usr/local/bin/sts // delete link

## python使用

json输出 | python –m json.tool

open/close port

ufw: uncomplicated firewall

Nmap: Network Mapper(网络映射器)

<https://nmap.org/man/zh/>

设计目标是快速地扫描大型网络，使用原始IP报文来发现网络上有哪些主机，那些主机提供什么服务(应用程序名和版本)，那些服务运行在什么操作系统(包括版本信息)，它们使用什么类型的报文过滤器/防火墙,以及一堆其它功能。

通常用于安全审核,也可做一些日常的工作，比如查看整个网络的信息，管理服务升级计划，以及监视主机和服务的运行。

“所感兴趣的端口表格”是其中的关键。那张表列出端口号，协议，服务名称和状态。状态可能是 open(开放的)，filtered(被过滤的)， closed(关闭的)，或者unfiltered(未被过滤的)。

Open(开放的)意味着目标机器上的应用程序正在该端口监听连接/报文。

filtered(被过滤的) 意味着防火墙，过滤器或者其它网络障碍阻止了该端口被访问，Nmap无法得知 它是 open(开放的) 还是 closed(关闭的)。

$sudo apt-get install –y nmap

SSH: secure shell 安全外壳协议

传统的网络服务程序，如：ftp、pop和telnet在本质上都是不安全的，因为它们在网络上用明文传送口令和数据，很容易受到“中间人”（man-in-the-middle）这种方式的攻击。所谓“中间人”的攻击方式， 就是“中间人”冒充真正的服务器接收你传给服务器的数据，然后再冒充你把数据传给真正的服务器.

通过使用SSH，你可以把所有传输的数据进行加密.

结构：SSH是由客户端和服务端的软件组成的,服务端是一个守护进程(daemon)，他在后台运行并响应来自客户端的连接请求。服务端一般是sshd进程，提供了对远程连接的处理，一般包括公共密钥认证、密钥交换、对称密钥加密和非安全连接。客户端包含ssh程序以及像scp（远程拷贝）、slogin（远程登陆）、sftp（安全文件传输）等其他的应用程序。

工作机制：本地的客户端发送一个连接请求到远程的服务端，服务端检查申请的包和IP地址再发送密钥给SSH的客户端，本地再将密钥发回给服务端，自此连接建立。

Linux一般会打开ssh网络服务(端口默认是22)，window通过Putty.exe访问

// config proxy

$ gedit ~/.bashrc

在打开的文件里添加

export http\_proxy=http://165.225.96.34:10015/

export https\_proxy=https://165.225.96.34:10015/

$ sudo gedit /etc/apt/apt.conf

Acquire::http::proxy "http://165.225.96.34:10015/";

Acquire::https::proxy "https://165.225.96.34:10015/";

$ sudo gedit /etc/environment

http\_proxy=http://165.225.96.34:10015/

https\_proxy=https://165.225.96.34:10015/

//有时候无法apt-getq

$sudo gedit /etc/resolvconf/resolv.conf.d/base

nameserver 8.8.8.8

nameserver 8.8.4.4

// install chrome

$ wget -q -O - https://dl-ssl.google.com/linux/linux\_signing\_key.pub | sudo apt-key add -

$ sudo sh -c 'echo "deb http://dl.google.com/linux/chrome/deb/ stable main" >> /etc/apt/sources.list.d/google.list'

$ sudo apt-get update

$ sudo apt-get install google-chrome-stable

// install notepad++

$ sudo -E add-apt-repository ppa:notepadqq-team/notepadqq

$ sudo apt-get update

$ sudo apt-get install notepadqq

//install jdk8 on ubuntu

open terminal via Ctrl+Alt+T (or search terminal)

$ sudo add-apt-repository ppa:webupd8team/java

(如果无法连接，用sudo -E add-apt-repository ppa:webupd8team/java)

$ sudo apt-get update

$ sudo apt-get install oracle-java8-installer

// verify installed java version

$ java -version

// configure java environment

$ sudo apt-get install oracle-java8-set-default

//install Intellij

Download ideaIU-2016.1.3.tar.gz

$cd /opt

$sudo tar –xvf ideaIU-2016.1.3.tar.gz

$sudo /opt/idea-IU-145.1617.8/bin/idea.sh //启动

// install spring tool suite

$ cd /opt

$ wget http://....tar.gz

$ sudo tar -xvf spring-tool-suite...tar.gz

$ ln -s /opt/sts-bundle/sts-...RELEASE/STS /usr/local/bin/sts

$ sts

if could not connect to network, please only set http and https, not set socks

// install gradle

download gradle

$sudo unzip <path to gradle-<>-all.zip>/gradle-<>-all.zip -f /opt/gradle

$ gedit ~/.bashrc

GRADLE\_HOME=<path to gradle bin file> (e.g. /opt/gradle/gradle-1.5/bin )

export GRADLE\_HOME

PATH=$PATH:$GRADLE\_HOME

export PATH

$source ~/.bashrc

$gradle -version

$gradle build –x test //exclude any task

$gradle bootRun

// install docker

$ wget -qO- https://get.docker.com/ | sh

$ sudo reboot

$ sudo gedit /etc/default/docker

export HTTP\_PROXY="http://161.92.51.225:8080/"

$ sudo service docker restart

// check port used and kill

$sudo netstat - tupln | grep 27017

$sudo kill -9 914 (914 is the pid to use port 27017)

//install MySQL

//安装过程会要求输入root用户的密码，所以有user: root, password: qzlin

$sudo apt-get install mysql-server

//客户端程序访问MySQL服务器

$mysql -u root -p //记得输入root密码

//操作数据库

mysql> show databases ; git

mysql> use infomation\_schema;

mysql> show tables ;

//root授权给用户test\_user,密码为test\_password使用数据库test\_db，有所有权限

mysql> GRANT ALL on test\_db.\* to test\_user IDENTIFIED by 'test\_password';

mysql> exit;

//install PostgreSQL

$sudo apt-get install postgresql

//启动postgresql服务器

$sudo /etc/init.d/postgresql start

$sudo /etc/init.d/postgresql stop

//访问服务器

$sudo -u postgres psql

postgres=# \l 显示所有数据库q

postgres=# CREATE DATABASE test; 创建数据库

postgres=# \c test 使用(连接)数据库

// install git

$sudo apt-get install git-core

$git --version

//install cURL

$ sudo apt-get install curl

// install hadoop

添加JAVA\_HOME (查看java安装路径$ update-alternatives --config java)

安装ssh

$ sudo apt-get install ssh

$ sudo apt-get install rsync

下载并解压hadoop-x.y.z.tar.gz,然后放入到opt目录中

$sudo mv Documents/hadoop/tools/hadoop-2.7.1 /opt/

设置环境变量

$gedit ~/.bashrc

HADOOP\_HOME=/opt/hadoop-2.7.1

PATH=$PATH:$HADOOP\_HOME/bin

export HADOOP\_HOME

export PATH

$source ~/.bashrc

检查hadoop version

$ hadoop version

java project

add \*.jar from $HADOOP\_HOME/bin/

mapper.java

reducer.java

JobTask.java

build

project$ export HADOOP\_CLASSPATH=bin

project$ hadoop project input/ncdc/sample.txt output