## TODO

自上而下重要性

<https://blog.csdn.net/bamboolsu/article/details/79002359>

https://blog.csdn.net/HuangLin\_Developer/article/details/78732306

<https://blog.csdn.net/xiezhi123456/article/details/80030593>

<https://blog.csdn.net/weixin_33890499/article/details/86841601>

<https://blog.csdn.net/shihai1991/article/details/52483830>

<https://blog.csdn.net/tq08g2z/article/details/78627653>

## tomcat

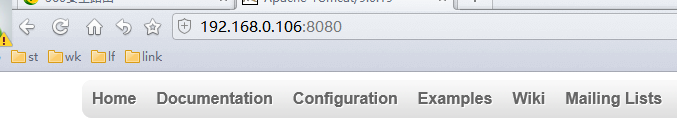
key@ubuntu:/usr/local/tomcat$ wget <http://mirrors.tuna.tsinghua.edu.cn/apache/tomcat/tomcat-9/v9.0.19/bin/apache-tomcat-9.0.19.tar.gz>

tar zxvf apache-tomcat-9.0.7.tar.gz

先修改一下tomcat文件夹的使用权限，否则在当前用户下不能进入bin目录

chmod 755 -R apache-tomcat-9.0.19/

key@ubuntu:/usr/local/tomcat/apache-tomcat-9.0.19/bin$ ./startup.sh



## JDK 安装

### 1、下载地址

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

jdk-8u151-linux-x64.tar.gz

key@key-u:~$ tar -zxvf jdk-8u151-linux-x64.tar.gz

mv jdk1.8.0\_151/ jdk8

key@key-u:/usr/lib/jvm$ sudo cp -rf /home/key/jdk8/ .

key@key-u:/usr/lib/jvm$ sudo vi /etc/profile

export JAVA\_HOME=/usr/lib/jvm/jdk8

export JRE\_HOME=${JAVA\_HOME}/jre

export CLASSPATH=.:$JAVA\_HOME/lib:$JAVA\_HOME/jre/lib:$CLASSPATH

export PATH=$JAVA\_HOME/bin:$JAVA\_HOME/jre/bin:$PATH

key@key-u:/usr/lib/jvm$ source /etc/profile

key@key-u:/usr/lib/jvm$ java -version

java version "1.8.0\_151"

Java(TM) SE Runtime Environment (build 1.8.0\_151-b12)

Java HotSpot(TM) 64-Bit Server VM (build 25.151-b12, mixed mode)

TOMCAT

用root权限，然后再启动tomcat，

key@key-u:/usr/local/tomcat85/bin$ sudo ./catalina.sh start

[sudo] password for key:

Neither the JAVA\_HOME nor the JRE\_HOME environment variable is defined

At least one of these environment variable is needed to run this program

否则tomcat启动成功，也不会被访问

用key 启动

# Status 500 – Internal Server Error

# Jenkins ~/.jenkins/secrets

# 忘记Jenkins管理员密码的解决办法

# admin/12345678

# Jenkins

Jenkins是一款开源自动化服务器，旨在自动化连续集成和交付软件所涉及的重复技术任务。 Jenkins是基于Java的，可以从Ubuntu软件包安装，也可以通过下载和运行其Web应用程序ARchive（WAR）文件进行安装 - 构成一个完整的Web应用程序的集合，该文件旨在在服务器上运行。

在本教程中，我们将通过添加其Debian软件包存储库来安装Jenkins，然后使用该存储库使用apt-get安装软件包

### 先决条件

要遵循本教程，您将需要：

* **一个Ubuntu 16.04 或者 Ubuntu18.04服务器**  
  配置了非root用户和防火墙，遵循[Ubuntu 16.04初始服务器设置指南](http://www.howtoing.com/initial-server-setup-with-ubuntu-16-04/" \t "_blank) 。 我们建议从至少1 GB的RAM开始。 请参阅[为大师](https://jenkins.io/doc/book/hardware-recommendations/)准备[正确的硬件，](https://jenkins.io/doc/book/hardware-recommendations/)以指导生产Jenkins安装的容量。
* 安装Java SDK

sudo apt-get install openjdk-8-jdk

#sudo apt-get install openjdk-7-jdk 早些系统可以安装

服务器设置完成后，您就可以随时随地。

## 第1步 - 安装Jenkins

包含在默认Ubuntu软件包中的Jenkins版本往往落后于项目本身的最新版本。 为了利用最新的修复和功能，我们将使用项目维护的软件包来安装Jenkins。

首先，我们将存储库密钥添加到系统。

wget -q -O - https://pkg.jenkins.io/debian/jenkins-ci.org.key | sudo apt-key add -

添加密钥后，系统将返回OK 。 接下来，我们将Debian包存储库地址附加到服务器的sources.list ：

echo deb http://pkg.jenkins.io/debian-stable binary/ | sudo tee /etc/apt/sources.list.d/jenkins.list

当这两个都到位时，我们将运行update ，以便apt-get将使用新的存储库：

sudo apt-get update

最后，我们将安装Jenkins及其依赖项，包括Java：

sudo apt-get install jenkins

**以上安装不成功的朋友，是由于Jenkins官网的校验码出现了问题。**  
指定早期版本可以安装。

sudo apt-get install jenkins=2.138.1

现在Jenkins及其依赖项已经到位，我们将启动Jenkins服务器。

## 第2步 - 开始Jenkins

使用systemctl我们将启动Jenkins：

sudo systemctl start jenkins

由于systemctl不显示输出，我们将使用其status命令来验证它是否成功启动：

sudo systemctl status jenkins

如果一切顺利，输出的开始应显示服务处于活动状态，并配置为启动时启动：

● jenkins.service - LSB: Start Jenkins at boot time

Loaded: loaded (/etc/init.d/jenkins; bad; vendor preset: enabled)

Active:active (exited) since Thu 2017-04-20 16:51:13 UTC; 2min 7s ago

Docs: man:systemd-sysv-generator(8)

现在Jenkins正在运行，我们将调整防火墙规则，以便我们可以从网络浏览器到达Jenkins以完成初始设置。

## 第3步 - 打开防火墙

默认情况下，Jenkins在端口8080上运行，因此我们将使用ufw打开该端口：

sudo ufw allow 8080

检查UFW的状态可以看到新的规则。

sudo ufw status

我们应该看到，从任何地方都可以访问8080端口：

Status: active

To Action From

-- ------ ----

OpenSSH ALLOW Anywhere

8080 ALLOW Anywhere

OpenSSH (v6) ALLOW Anywhere (v6)

8080 (v6) ALLOW Anywhere (v6)

现在，Jenkins已安装，防火墙允许我们访问它，我们可以完成初始设置。

## 第3步 - 设置Jenkins

要设置我们的安装，我们将使用服务器域名或IP地址访问Jenkins的默认端口8080 ： http:// ip\_address\_or\_domain\_name :8080

我们应该看到“解锁Jenkins”屏幕，显示初始密码的位置

图片.png

在终端窗口中，我们将使用cat命令显示密码：

sudo cat /var/lib/jenkins/secrets/initialAdminPassword

我们将从终端复制32个字符的字母数字密码，并将其粘贴到“管理员密码”字段中，然后单击“继续”。 下一个屏幕提供安装建议的插件或选择特定插件的选项。

图片.png

我们将点击“安装建议的插件”选项，这将立即开始安装过程：

图片.png

安装完成后，系统将提示您设置第一个管理用户。 可以跳过此步骤，并使用上面使用的初始密码作为admin继续，但是我们将花一点时间创建用户。

**注意：**默认Jenkins服务器未加密，因此使用此表单提交的数据不受保护。 当您准备好使用此安装时，请按照指南[如何使用Nginx反向代理将SSL配置为Jenkins](http://www.howtoing.com/how-to-configure-jenkins-with-ssl-using-an-nginx-reverse-proxy/) 。 这将保护用户凭据和关于通过Web界面发送的构建的信息。

图片.png

一旦第一个管理员用户到位，你应该看到一个“Jenkins准备好了！” 确认屏幕。

图片.png

点击“开始使用Jenkins”来访问主要的Jenkins仪表板：

图片.png

查看

sudo netstat -plntu

此时，Jenkins已经成功安装。

默认端口是8080，有时候由于端口占用需要修改如下：

1，检查 /etc/init.d/jenkins 脚本，修改 do\_start 函数的 check\_tcp\_port 命令，端口号从 8080 换成 8082：

2，修改 /etc/default/jenkins 文件，将端口 8080 改成 8082

image.png

3，重启Jenkins

sudo systemctl restart jenkins

## 结论

在本教程中，我们使用项目提供的软件包安装了Jenkins，启动了服务器，打开了防火墙，并创建了一个管理用户。 在这一点上，您可以开始探索Jenkins。

完成探索后，如果您决定继续使用Jenkins，请按照指南， [如何使用Nginx反向代理使用SSL配置Jenkins，](http://www.howtoing.com/how-to-configure-jenkins-with-ssl-using-an-nginx-reverse-proxy/)以保护密码以及任何敏感的系统或将发送的产品信息在您的机器和服务器之间的纯文本。

## 参考

链接：https://www.jianshu.com/p/845f267aec52

# 搭建gitlab

1.安装依赖包，运行命令

sudo apt-get install curl openssh-server ca-certificates postfix

执行完成后，出现邮件配置，选择Internet那一项（不带Smarthost的）

2.利用清华大学的镜像（https://mirror.tuna.tsinghua.edu.cn/help/gitlab-ce/）来进行主程序的安装。

首先信任 GitLab 的 GPG 公钥:

curl https://packages.gitlab.com/gpg.key 2> /dev/null | sudo apt-key add - &>/dev/null

3.添加GitLab软件包存储库。

curl -sS https://packages.gitlab.com/install/repositories/gitlab/gitlab-ce/script.deb.sh | sudo bash

更新为xenial（目前自动修改好了）

1.cat /etc/apt/sources.list.d/gitlab\_gitlab-ce.list

旧

deb https://packages.gitlab.com/gitlab/gitlab-ce/ubuntu/ zesty main

deb-src https://packages.gitlab.com/gitlab/gitlab-ce/ubuntu/ zesty main

新

deb https://packages.gitlab.com/gitlab/gitlab-ce/ubuntu/ xenial main

deb-src https://packages.gitlab.com/gitlab/gitlab-ce/ubuntu/ xenial main

接下来刷新apt缓存。

sudo apt-get update

接下来，安装GitLab CE软件包。

sudo apt-get install gitlab-ce

4执行命令

sudo gitlab-ctl reconfigure

5.打开 sshd 和 postfix 服务

service sshd start

service postfix start

6.要使 GitLab 社区版的 Web 界面可以通过网络进行访问，我们需要允许 80 端口通过防火墙，这个端口是 GitLab 社区版的默认端口。为此需要运行下面的命令

sudo iptables -A INPUT -p tcp -m tcp --dport 80 -j ACCEPT

7.检查GitLab是否安装好并且已经正确运行，输入下面的命令

sudo gitlab-ctl status

如果得到类似下面的结果，则说明GitLab运行正常

run: gitlab-workhorse: (pid 1148) 884s; run: log: (pid 1132) 884s

run: logrotate: (pid 1150) 884s; run: log: (pid 1131) 884s

run: nginx: (pid 1144) 884s; run: log: (pid 1129) 884s

run: postgresql: (pid 1147) 884s; run: log: (pid 1130) 884s

run: redis: (pid 1146) 884s; run: log: (pid 1133) 884s

run: sidekiq: (pid 1145) 884s; run: log: (pid 1128) 884s

run: unicorn: (pid 1149) 885s; run: log: (pid 1134) 885s

8.在浏览器地址栏中输入：http://ip，即可访问GitLab的Web页面,首次使用时，浏览器Web页面会提示设置密码

Key/Gone0\*\*\*

===========================================================================================================

1.默认已经启动，gitlab监听的是80端口，所以可以直接ip访问，或者可以修改如下配置文件添加域名：

/etc/gitlab/gitlab.rb

所以要更改gitlab监听端口，如下的site configuration已经默认添加在nginx中：

cd /var/opt/gitlab/nginx/conf

vim gitlab-http.conf

---- site configuration(截取一部分) ----

upstream gitlab-workhorse {

server unix:/var/opt/gitlab/gitlab-workhorse/socket;

}

server {

listen \*:80; --修改端口

server\_name localhost;

server\_tokens off; ## Don't show the nginx version number, a security best practice

将其中的80改为其它端口即可，如我的是8099，如下，重启gitlab等待网关恢复，重新访问：http://ip:8080

gitlab-ctl restart

2.在GitLab上新建一个项目test\_gitlab，刚开始仓库地址是http://localhost/yulei/test\_gitlab.git .

把localhost 换成本机的IP地址，修改方式如下：

修改gitlab.yml文件：

[root@localhost config]# cd /opt/gitlab/embedded/service/gitlab-rails/config

[root@localhost config]# vim gitlab.yml

development:

<<:\*base

test:

<<:\*base

gravatar:

enabled:true

gitlab:

host:ip

port:端口

修改gitlab下host值为实际的域名或服务器IP地址即可，保存退出

重启GitLab

gitlab-ctl restart

3.解决AIP 不可用

/var/opt/gitlab/gitlab-shell/config.yml中的 gitlab\_url: http://localhost/，此项默认为

http://localhost/ 如果不Gitlab服务器配置的是子域名，或者是其它域名，则此项必须修

改为最终Gitlab服务器的域名，内网IP如果给Gitlab配置了端口，记住也要将http://localhost

改为http://ip:8080这样的带端口形式

gitlab安装完毕后，默认开机启动，如果不想占用内存  
sudo gitlab-ctl start/stop

## 创建新的仓库

可是这个是当个仓库的，多个仓库怎么办呢？？？我们通过.repo

手动单个操作

git remote add kg001 http://192.168.0.106/android-9.0.0\_r35/frameworks/native.git

git push -u kg001 –all

#输入账号key/Gone\*\*\*\*

git push -u kg001 –tags

GitLab 11.2 发布，可批量导入 Android 操作系统代码

<https://www.oschina.net/news/99221/gitlab-11-2-released>

导入具有多个子结构的复杂项目结构是一项繁琐且耗时的任务。通过对 XML 清单文件的新支持，现在可以批量导入包含多个存储库的大型项目结构，包括来自 Android 开源项目（AOSP）的 Android 操作系统代码。

目前的参考资料还少了

## 卸载Gitlab

# 停止gitlab

sudo gitlab-ctl stop

# 查看进程

ps -e | grep gitlab

# 删除所有包含gitlab的文件及目录

find / -name gitlab | xargs rm -rf

# 卸载

sudo apt-get remove gitlab-ee

# 检查还有没有卸载的gitlab相关软件

dpkg --get-selections | grep gitlab

gitlab-ee deinstall

# 再执行

sudo apt-get --purge remove gitlab-ee

## 参考

https://blog.csdn.net/qq\_36467463/article/details/78283874

# git+repo+

## 安装gitolite server

#从网站上获取gitolite的代码，是安装文件

yck@ubuntu:~$ git clone git://github.com/sitaramc/gitolite

yck@ubuntu:~$ mkdir bin

yck@ubuntu:~$ ./gitolite/install -ln

yck@ubuntu:~$ ssh-keygen -t rsa#生成的id\_rsa.pub，可随意改git.pub

echo "PATH=$HOME/bin:$PATH" > ~/.bashrc #配下环境变量

gitolite setup -pk git.pub  #生成repository。

#配置用户名和邮箱

yck@ubuntu:~$ git config --global user.name key.guan

yck@ubuntu:~$ git config --global user.email [key.guan@qq.com](mailto:key.guan@qq.com)

## Gerrit安装

wget <https://gerrit-releases.storage.googleapis.com/gerrit-3.0.0.war>

yck@ubuntu:~$ java -jar gerrit-3.0.0.war init -d gerrit

Create '/home/key/aosp/home/yck/gerrit' [Y/n]? Y

Authentication method [openid/?]: http

Install Verified label [y/N]? y

Behind reverse proxy [y/N]? y

其他的默认即可

### 认证方式dev\_yck

修改gerrit认证方式

gerrit有多种身份验证方法，身份验证方法决定了如何登录Gerrit。

OPENID,如果你想挂入某个现有的身份验证提供方（例如GoogleAccounts），那么可以使用OpenID。

development\_become\_any\_account,如果是用于测试和学习，可以选择最简单的development\_become\_any\_account。

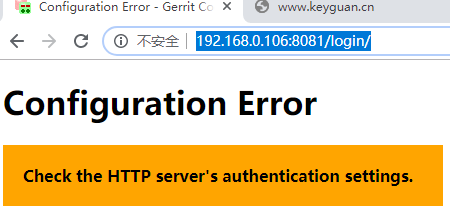
HTTP认证也是可选的认证方式，此认证方式下需要配置Apache的反向代理，并在Apache中配置Web站点的口令认证，通过口令认证后gerrit在创建账号的过程中会询问用户的邮件地址并发送确认邮件。

LDAP, LDAP全称Lightweight Directory Access Protocol，轻量目录访问协议。使用用户名和密码。

在etc/gerrit.config下的

[auth]

type = LDAP



### HTTP认证

配置gerrit.config，可参考下面

vi gerrit/etc/gerrit.config

[gerrit]

basePath = git

canonicalWebUrl = [http://192.168.0.106](http://192.168.0.106/):8081/

[database]

type = H2

database = db/ReviewDB

[auth]

type = HTTP

[sendemail]

smtpServer = localhost

[container]

user = yck

javaHome = /usr/lib/jvm/java-6-oracle/jre

[sshd]

listenAddress = \*:29418

[httpd]

listenUrl = proxy-<http://192.168.0.106:8081/>

[cache]

directory = cache

设置第一个gerrit用户（管理员）的帐号和密码

yck@ubuntu:~ $ touch gerrit/etc/passwords

yck@ubuntu:~ $ htpasswd gerrit/etc/passwords yck

这里的yck就是用来登录gerrit的用户名。以后要为gerirt增加用户，也需要通过htpasswd命令来添加。用htpasswd创建的用户时，并没有往gerrit中添加账号，只有当该用户通过web登陆gerrit服务器时，该账号才会被添加进gerrit数据库中。

## 安装apache2

安装apache2

sudo apt-get install apache2

### 配置端口映射

8080（浏览器中访问gerrit时输入的端口，暴露给用户）映射为

8081（内部端口，对用户不可见，实际访问的端口）

#### 设置要监听的端口

vi /etc/apache2/port.conf

#keyguan

Listen 8080

#### 设定代理映射

root@ubuntu:/etc/apache2# cat apache2.conf

Include gerrit.conf

新建gerrit.conf

yck@ubuntu:~$ sudo vi /etc/apache2/gerrit.conf

反向代理可参考下面

: root@ubuntu:/etc/apache2# cat gerrit.conf

<VirtualHost \*:8080>

ServerName 192.168.0.106

ProxyRequests Off

ProxyVia Off

ProxyPreserveHost On

<Proxy \*>

Order deny,allow

Allow from all

</Proxy>

<Location /login/>

AuthType Basic

AuthName "Gerrit Code Review"

AuthBasicProvider file

AuthUserFile /home/key/aosp/home/yck/gerrit/etc/passwords

Require valid-user

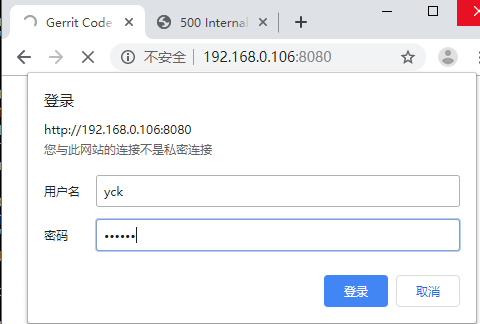
</Location>

ProxyPass / http://192.168.0.106:8081/

</VirtualHost>

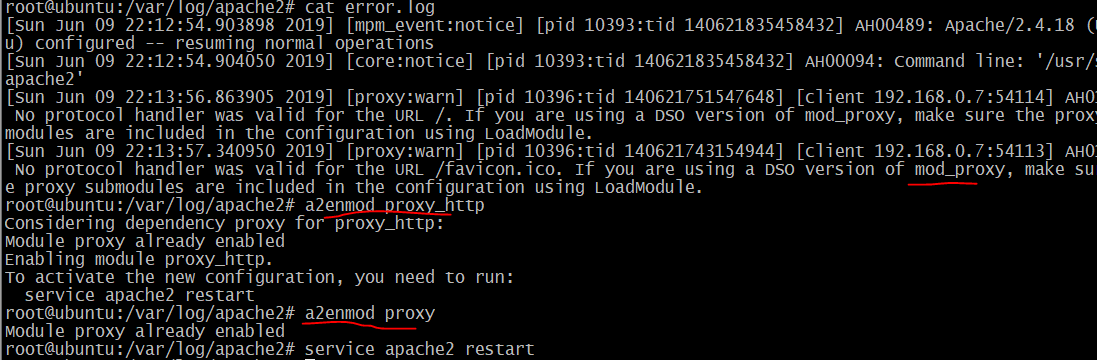
看日志？

重启一下系统生效！



### QA

报错日志/var/log/apache2



a2enmod proxy\_http

a2enmod proxy

root@ubuntu:/etc/apache2# service apache2 restart

AH00526: Syntax error on line 3 of /etc/apache2/gerrit.conf:

sudo a2enmod proxy

sudo /etc/init.d/apache2 restart

参考<http://www.makaidong.com/Apache/482875.shtml>

## 运行

重启服务

sudo /etc/init.d/apache2 restart

yck@ubuntu:~$ ./gerrit/bin/gerrit.sh restart

## 参考

主要参考：https://blog.csdn.net/bamboolsu/article/details/79002359

<https://download.csdn.net/download/weixin_35311884/10416091>

XXXX在ubuntu16上搭建gitlab（实测可用）

<https://blog.csdn.net/qq_36467463/article/details/78283874>

# XXXXGitlab

https://about.gitlab.com/installation/#ubuntu

sudo vim /etc/gitlab/gitlab.rb

GitLab默认会占用80、8080和9090端口，很不巧，Jenkins默认也会使用8080端口，所以，这一步操作将会修改GitLab的默认端口为11000、11001和11002，如果你还是希望使用默认端口的话，可以跳过这一步。

external\_url 'http://<你的服务器地址或域名>:11000'

# unicorn['port'] = 8080 –--> unicorn['port'] = 18080

prometheus['listen\_address'] = 'localhost:9090'

----------->

prometheus['listen\_address'] = 'localhost:19090'

sudo gitlab-ctl reconfigure

/var/log/gitlab

netstat -ap | grep 18080

tcp 0 0 \*:18080 \*:\* LISTEN 929/nginx

tcp 0 0 localhost:18080 localhost:43274 TIME\_WAIT -

tcp 0 0 localhost:18080 localhost:43464 TIME\_WAIT -

tcp 0 0 localhost:18080 localhost:43290 TIME\_WAIT -

tcp 0 0 localhost:18080 localhost:43306 TIME\_WAIT -

unix 3 [ ] 流 已连接 18080 1593/dbus-daemon @/tmp/dbus-JSAll0uWet

修改端口号码

## GitLab使用gitlab-ctl日志查看的方法

Gitlab 默认的日志文件存放在/var/log/gitlab 目录下

# 查看所有日志

|  |
| --- |
| sudo gitlab-ctl tail |

# 查看nginx 访问日志

|  |
| --- |
| sudo gitlab-ctl tail nginx/gitlab\_acces.log |

# 查看 postgresql 日志

sudo gitlab-ctl tail  postgresql

<http://blog.csdn.net/wangxicoding/article/details/43738137>

502问题\

默认帐号：root/12345678

GitLab external URL must include a schema and FQDN, e.g. <http://gitlab.example.com/>

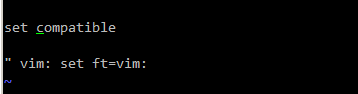
GitLab external URL schema FQDN

QA：~/.bashrc和.profile，/etc/profile关系

在Ubuntu下，初始使用vi的时候有点问题，就是在编辑模式下使用方向键的时候，并不会使光标移动，而是在命令行中出现[A [B [C [D之类的字母，而且编辑错误的话，就连平时关于的退格键(Backspace键)都使用不了，只能用Delete来删除。针对这个问题，网上的答案有很多，例如是安装完整版的vim啊，编辑/etc/vim/vimrc.tiny等方法，很杂很乱，在这里我就做个详细的总结吧：

一、编辑/etc/vim/vimrc.tiny

由于/etc/vim/vimrc.tiny的拥有者是root用户，所以要在root的权限下对这个文件进行修改。很简单，这个文件里面的倒数第二句话是“set compatible”，如下所示：



将“compatible”改成“nocompatible”非兼容模式就可以解决方向键变ABCD的问题了。接下来要解决Backspace键的问题也很简单，在刚才那句话后面再加一句：

backspace有几种工作方式，默认是vi兼容的。对新手来说很不习惯。对老vi 不那么熟悉的人也都挺困扰的。可以用

set backspace=indent,eol,start

来解决。

indent: 如果用了:set indent,:set ai 等自动缩进，想用退格键将字段缩进的删掉，必须设置这个选项。否则不响应。

eol:如果插入模式下在行开头，想通过退格键合并两行，需要设置eol。

start：要想删除此次插入前的输入，需设置这个。

就可以解决问题了。这个时候，无论对于哪个用户，使用vi都能很方便的按照我们的习惯去编辑文件了。

一、Ubuntu的  
默认root密码是随机的，即每次开机都有一个新的root密码。我们可以在终端输入命令 sudo passwd，然后输入当前用户的密码，  
enter，

二、终端会提示我们输入新的密码并确认，此时的密码就是root新密码。修改成功后，输入命令 su root，再输入新的密码就ok了。