

一台新开的服务器要做什么

新开了一台阿里学生机，几个人拿来练手玩。这里记录一下新开的服务器要做些什么

连接服务器

- 在阿里控制台修改 root 用户密码
- Xshell 等客户端通过 用户名/密码 连接服务器即可
- 后续也可以配置密钥，通过密钥连接

改主机名

刚开的服务器，随机给的主机名好丑，所以先改主机名

- 查看主机名

```
[root@izuf6fag228dhcruoylsq2z ~]# uname -n  
izuf6fag228dhcruoylsq2z
```

- 临时改名为 **aliyun**

```
[root@izuf6fag228dhcruoylsq2z ~]# hostname aliyun  
[root@izuf6fag228dhcruoylsq2z ~]# exit
```

此命令的作用是暂时的修改 linux 的主机名，它的存活时间是 linux 当前的运行时间，即在下次重启前的运行时间内。一般修改以后立即生效，但这只是临时生效，不是永久生效

- 永久改名（适用于 centos 7）

```
[root@aliyun ~]# hostnamectl set-hostname aliyun  
[root@aliyun ~]# shutdown -r now
```

待重启后，重新连接即可永久改名为 **aliyun**

创建用户和用户组及赋权

既然是多用户使用，当然是创建不同的用户以免相互干扰了

linux 中多用户优点如下：

Linux 系统是一个多用户多任务的分时操作系统，任何一个要使用系统资源的用户，都必须首先向系统管理员申请一个账号，然后以这个账号的身份进入系统。

用户的账号一方面可以帮助系统管理员对使用系统的用户进行跟踪，并控制他们对系统资源的访问；另一方面也可以帮助用户组织文件，并为用户提供安全性保护。

- 创建用户

```
[root@aliyun ~]# adduser tai
[root@aliyun ~]# passwd tai
Changing password for user tai.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
```

使用 `adduser` 命令即可创建用户，会自动为创建的用户指定主目录、系统shell版本

- 赋权

```
[root@aliyun ~]# cd /home # 进入 /home 目录
[root@aliyun home]# ls # 查看用户目录
admin tai # tai 即为刚才创建的用户目录

[root@aliyun home]# chmod 755 -R tai # 赋予 tai 用户相关权限
```

- 创建用户组

```
[root@aliyun /]# groupadd group1 # 创建 group1 用户组
[root@aliyun /]# usermod -g group1 tai # 将用户 tai 添加到用户组 group
```

- 为其它用户赋予 `sudo` 权限

```
chmod u+w /etc/sudoers # 添加 sudo 文件的写权限
vi /etc/sudoers # 编辑 sudoers 文件
```

找到这行 `root ALL=(ALL) ALL`，在他下面添加 `xxx ALL=(ALL) ALL`（这里的 `xxx` 是你的用户名），添加 `%group1 ALL=(ALL) NOPASSWD: ALL`，保存退出
用户组 `group1` 即具有 `sudo` 权限了

上传和下载

- 使用 `rz` 与 `sz` 命令

`rz`：上传文件 `sz`：下载文件 初始状态并不能直接使用这两个命令，需要安装 `lrzsz` 工具包：

```
yum install lrzsz
```

安装成功后即可在客户端使用上传和下载命令了

安装 Python3

linux 系统初始都自带了 python2, 然而在 2019 年已经过去了 29.4% 的今天, 当然是要装 python3

初始的 python2 最好不用动, linux 有些命令会需要到 python2 的支持

- python 下载源 首先, 在[这里](#)找到指定的 python 版本的下载链接
- 下载

```
[xuan@aliyun ~]$ wget https://www.python.org/ftp/python/3.7.3/Python-3.7.3.tar.xz
```

- 解压

```
[xuan@aliyun ~]$ tar xvf Python-3.7.3.tar.xz
```

- 进入解压文件夹目录下

```
[xuan@aliyun ~]$ cd Python-3.7.3
```

- 创建安装目录 (在当前用户文件夹下)

```
[xuan@aliyun ~]$ mkdir -p /home/xuan/local/python3
```

- 配置安装目录 简单的配置安装目录 (但会有 `pip` 不能用的问题)

```
[xuan@aliyun ~]$ ./configure --prefix=/home/xuan/local/python3
```

不加 `--with-ssl` 参数会导致 `pip` 功能不能用, 可以加上 `--with-ssl` 参数, 重新编译安装

带有 `--with-ssl` 参数

```
[xuan@aliyun ~]$ ./configure --prefix=/home/xuan/local/python3 --enable-optimizations --with-ssl
```

加了 `--with-ssl` 参数可能会导致安装报错, 此时需要检查 `openssl-devel` 包:

```
[xuan@aliyun ~]$ rpm -aq|grep openssl
openssl-libs-1.0.2k-16.el7_6.1.x86_64
openssl-1.0.2k-16.el7_6.1.x86_64
```

发现缺少 `openssl-devel` 包, 安装:

```
[xuan@aliyun ~]$ yum install openssl-devel -y
```

安装成功后即可通过编译运行

如果在不同用户, 还是有遇到 `pip` 报 `SSL` 错误的情况:

- 修改 Modules 文件夹下的 Setup 文件（python 解压后的路径下） 大概在 Setup 文件的 217 行左右，按如下修改，尤其注意 `_socket`

```
# Socket module helper for socket(2)
_socket socketmodule.c

# Socket module helper for SSL support; you must comment out the other
# socket line above, and possibly edit the SSL variable:
SSL=/etc/ssl
_ssl _ssl.c \
    -DUSE_SSL -I$(SSL)/include -I$(SSL)/include/openssl \
    -L$(SSL)/lib -lssl -lcrypto
```

再次编译安装

- 编译安装

```
[xuan@aliyun ~]$ make # 编译
[xuan@aliyun ~]$ make install # 安装
```

或者直接编译 + 安装

```
[xuan@aliyun ~]$ make & make install
```

安装报错解决

这里第一次安装报错了：`zipimport.ZipImportError: can't decompress data; zlib not available`

是因为缺少 `zlib` 的相关工具包导致的；那么，缺什么就装什么

打开终端，安装 `zlib` 相关依赖包：

```
yum -y install zlib*
```

安装完成后重新编译安装 `python`，再一次报错了：`ModuleNotFoundError: No module named '_ctypes'`

是因为 `python3.7` 需要一个新的包 `libffi-devel`

安装 `libffi-devel`：

```
yum install libffi-devel -y
```

安装完成后再次编译安装 `python`：

```
make & make install
```

终于安装成功

```
Looking in links: /tmp/tmpeo54cgzv
Collecting setuptools
Collecting pip
Installing collected packages: setuptools, pip
Successfully installed pip-19.0.3 setuptools-40.8.0
[1]+ Done make
```

为 python3 建立软链接

Linux 下的软链接其实就相当于 Windows 下的快捷方式

```
ln -s /home/xuan/local/python3/bin/python3.7 /usr/bin/python3
```

第一个路径是刚刚 python3 的安装路径下的 python 可执行文件，把这个路径链接到 /usr/bin 中，并创建 python3 命令

同理，可以为 python3 下的 pip 创建一个名为 pip3 的软链：

```
ln -s /home/xuan/local/python3/bin/pip3.7 /usr/bin/pip3
```

至此，python2 与 python3 可以共用

Git

安装 git 的目的就不言而喻了

- 安装 目标服务器是 centos 7, yum 流中自带 git, 就比较简单了:

```
yum install git
```

- 配置用户

```
git config --global user.email "YourEmail"
git config --global user.name "YourName"
```

这里要讲一下，一开始我以为这里的 `--global` 参数指的是全局配置，多用户共享那种。结果发现，`--global` 只在当前 linux 用户下生效，即指的是当前用户的全局配置 实现方式是在当前用户文件夹下生成 `./gitconfig` 文件，记录该用户的全局配置：

```
[user]
  email = YourEmail
  name = YourName
[push]
  default = simple
```

至此，git 也可以使用了

首次使用 `git push` 命令上传的时候，可能会报错：

```
warning: push.default is unset; its implicit value has changed in Git 2.0 from
'matching' to 'simple'
```

原因也就如字面意思，git 版本大于 2.0 以后，`push.default` 初始没有配置，需要配置之后才能 push，这里有两种方案：

- 消除此警告并保留以前的习惯

```
git config --global push.default matching
```

- 消除此警告并采用新的设置值

```
git config --global push.default simple
```

具体区别不明，保守一些选择了 `simple`

还不会用 Git 的同学可以看[这里](#)

mysql

一个服务器怎么能没有数据库呢？

- 安装

Reference

[Xshell登录阿里云服务器ECS](#)

[阿里云服务器永久修改主机名](#)

[Linux创建用户和赋权限](#)

[Linux用户及用户组添加和删除操作](#)

[centos添加和删除用户及 xxx is not in the sudoers file.This incident will be reported. 的解决方法](#)

[XShell-上传、下载文件（使用sz与rz命令）](#)

[关于在centos下安装python3.7.0以上版本时报错ModuleNotFoundError: No module named '_ctypes'的解决办法](#)

[Python安装常见问题\(1\):zipimport.ZipImportError: can't decompress data](#)

[python pip 出现locations that require TLS/SSL异常处理方法 pip安装第三方包出现ssl错误的解决方法](#)

[手把手教你在Linux环境下安装Python3](#)

[Git 2.x 中git push时遇到 push.default 警告的解决方法](#)

ShadowMimosa
2019. 4. 10