

## - VPS

這是我租用了Linode的VPS後，在上面架設一個deploy用的Rails環境的一些筆記，為什麼使用Ubuntu 10.04 LTS的版本單純是因為之前用過，覺得沒什麼特別問題因此使用。

### 環境

**OS:** Ubuntu 10.04 LTS

**HTTP Server:** nginx

**Ruby:** 1.9.2-p290

**Rails:** 3.1.3

### 更新系統

```
$ apt-get update
$ apt-get upgrade
```

### 設定Hostname

```
$ echo "gogojimmy" > /etc/hostname      #將gogojimmy替換成你想要的hostname
$ hostname -F /etc/hostname
```

### 更新/etc/hosts

```
$ vi /etc/hosts
```

在127.0.0.1下面加上自己的網域

```
127.0.0.1      localhost.localdomain    localhost
12.34.56.78    gogojimmy.net                  gogojimmy
```

### 設定時區

```
$ dpkg-reconfigure tzdata
```

使用date指令若可以看到回應出現在時間的話表示你設定成功

###建立deploy用的帳號並提升權限 ---

```
$ useradd deployer      #建立deploy用帳號deployer
$ passwd deployer       #建立deployer的密碼
$ mkdir /home/deployer  #建立deployer的家目錄
$ chown -R deployer /home/deployer #為deployer設定家目錄權限
$ visudo                #設定sudoer
```

在root ALL=(ALL) ALL下面加入這一行，可以讓deployer這個帳號使用sudo指令來獲得root的所有權限

```
deployer ALL=(ALL) ALL
```

### 設定SSH

```
$ su deployer      #切換帳號成deployer
$ ssh-keygen        #產生ssh key pair
$ more ~/.ssh/id_rsa.pub  #這個key的內容日後用來貼到你想要deploy專案的deploy key
```

\*\*\*Note: 下面這一步若你本機上還沒有ssh key pair的話請先使用ssh-keygen先產生\*\*\*

```
$ vi ~/.ssh/authorized_keys      #將你本機的身_rsa.pub內容貼入authorized_keys
```

```
$ chmod 711 ~/.ssh
$ chmod 644 ~/.ssh/authorized_keys
```

這樣以後就可以免密碼登入ssh了。###安裝系統環境套件---

```
$ sudo apt-get install gcc
$ sudo apt-get install build-essential
$ sudo apt-get install bison openssl libreadline6 libreadline6-dev
$ sudo apt-get install curl git-core zlib1g zlib1g-dev libssl-dev libyaml-dev
$ sudo apt-get install libsqlite3-0 libsqlite3-dev sqlite3 libxml2-dev
$ sudo apt-get install libxslt-dev autoconf libc6-dev
```

## 安裝mysql

---

```
$ sudo apt-get install mysql-server libmysqlclient15-dev
```

## 安裝nginx

---

將nginx的source貼到/etc/apt/sources.list:

```
$ sudo deb http://nginx.org/packages/ubuntu/ lucid nginx
$ sudo ddeb-src http://nginx.org/packages/ubuntu/ lucid nginx
$ sudo -s
$ sudo dnginx=stable
$ sudo dadd-apt-repository ppa:nginx/$nginx
$ sudo dapt-get update
$ sudo dapt-get install nginx
$ nginx -v                                     #看的到版本編號的話表示已安裝成功
$ sudo /etc/init.d/nginx start                 #啟動nginx
```

## 安裝Ruby

---

```
$ wget http://ftp.ruby-lang.org/pub/ruby/1.9/ruby-1.9.2-p290.tar.gz
$ tar -xzf ruby-1.9.2-p290.tar.gz
$ cd ruby-1.9.2-p290
$ ./configure
$ make
$ make test
$ make install
$ ruby -v                                     #看的到版本編號表示已經安裝成功
```

## Rubygems

---

Ruby 1.9.2內建的Rubygems版本較低，因此我們先將他升級到最新版本

```
$ sudo gem update --system
```

將以下內容貼到~/.gemrc，讓gem安裝時不要安裝ri跟rdoc文件以節省安裝時間

```
:sources:
- http://gems.rubyforge.org
- http://gems.github.com
gem: --no-ri --no-rdoc
```

## 安裝Rails

---

```
$ sudo gem install Rails
```

## 安裝ImageMagick與RMagick

---

先上 [Imagemagick.org](http://imagemagick.org)上找到最新的版本，使用wget下載

```
$ wget ftp://ftp.kddlabs.co.jp/graphics/ImageMagick/ImageMagick-6.6.9-10.tar.gz
$ tar xvfz ImageMagick-6.6.9-10.tar.gz
```

```
$ cd ImageMagick-6.6.9-10
$ ./configure
$ make
$ sudo make install
```

## 安裝RMagick

```
$ sudo gem install rmagick
```

## 安裝Passenger

```
$ sudo /usr/local/bin/passenger-install-nginx-module
```

安裝時選擇1預設安裝就可以了。###測試環境，使用octopress --- 我們來安裝octopress來測試剛剛的環境是否OK，請先參考[Octopress Setup](<http://octopress.org/docs/setup/>)來建立你的octopress，設定好後使用下面指令來上傳到你的linode。

```
$ rake generate
$ rake deploy
```

再來我們去更改一下nginx內的設定

```
$ sudo vi /etc/nginx/nginx.conf
```

在中間加上這段Server敘述：

```
server {
    listen 80; ## listen for ipv4
    listen [::]:80 default ipv6only=on; ## listen for ipv6

    server_name example.com www.example.com;
    access_log /var/log/nginx/example.access.log;

    location / {
        root /path/to/deployed/files;
    }
}
```

然後重新啟動你的nginx，你就可以在你的網址上看到你的Octopress了！

```
sudo /etc/init.d/nginx restart
```

**\*\*NOTE\*\***:如果在重新啟動nginx時看到nginx: [emerg] bind() to 0.0.0.0:80 failed (98: Address already in use)的錯誤訊息，那是因為nginx重複重新啟動佔用了port的關係，使用下面指令殺掉後再啟動就可解決了。

```
sudo killall -9 nginx
```

有什麼問題都歡迎留言討論！