

武汉云页移动科技有限公司

项目开发环境搭建技术文档

前言

本文档基于Windows平台，基本原理是在windows操作系统上安装虚拟机和开发集成环境(Pycharm)，将代码clone到虚拟机，在虚拟机中运行项目代码，便于开发人员进行本地开发、调试、测试。

开发人员阅读本文档后，应可独立完成开发环境搭建。

使用的软件

- 虚拟机：VMware-player-7.1.2-2780323.exe
- 虚拟机操作系统：ubuntu-14.04.3-desktop-i386.iso， 32位
- GIT 1.9.1
- Open-SSH
- PyCharm Professional Edition 4.5.4.exe
- Python 3.4
- Django 1.8.3

安装基础软件

1 安装虚拟机软件

在windows系统安装按步骤安装：VMware-player-7.1.2-2780323.exe

2 运行虚拟机软件，安装Ubuntu操作系统

ubuntu-14.04.3-desktop-i386.iso

Linux（Ubuntu）测试环境配置

以下内容基于在上一步的虚拟机中安装完成的ubuntu操作系统, 使用命令行操作, 使用‘Ctrl+Alt+T’组合键打开命令行终端窗口。

1 更新系统

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

2 安装系统依赖包

```
$ sudo apt-get install build-essential
```

3 配置虚拟机操作系统IP

3.1 检查ifconfig

```
$ ifconfig
```

使用ifconfig命令检查当前虚拟机操作系统的IP信息, 并记录下来

3.2 设置静态IP

```
$ sudo nano /etc/network/interfaces
```

在打开的窗口中, 根据实际情况编辑如下内容:

```
auto eth0  
iface eth0 inet static  
address 192.168.xxx.yyy  
netmask 255.255.255.0  
network 192.168.xxx.0  
broadcast 192.168.xxx.255  
gateway 192.168.xxx.2  
dns-nameservers 192.168.xxx.2
```

4 安装git、Open-SSH

```
$ sudo apt-get install git
$ sudo apt-get install openssh-server
```

5 配置SSH Key

```
$ mkdir ~/.ssh
$ chmod 700 ~/.ssh
$ ssh-keygen -t rsa -b 4096 -C "your_email@example.com" //
开发人员邮箱
```

6 获取提交代码的权限

```
$ cat ~/.ssh/id_rsa.pub
```

复制上面命令输出的所有内容，发送给代码仓库管理员即可

7 配置python3运行环境

ubuntu系统已自带python3，使用如下命令检查python3版本号

```
$ python3 --version
```

7.1 安装pip3

pip3是python的包管理工具

```
$ sudo apt-get install python3-pip
```

7.2 安装virtualenv

virtualenv用来隔离python运行环境的工具软件，可以用来将项目需要使用到的第三方python包隔离，而不会对系统自带的python环境造成污染。

```
$ sudo pip3 install virtualenv
```

7.3 使用virtualenv创建项目虚拟环境

```
$ cd ~/
$ virtualenv --python=/usr/bin/python3 venv
```

使用virtualenv命令创建名为"venv"的虚拟环境，且目录名为venv，并且在venv/bin 目录中已经包含了python3及pip3

7.4 安装ffmpeg组件

```
$ sudo add-apt-repository ppa:mc3man/trusty-media
$ sudo apt-get update
$ sudo apt-get dist-upgrade
$ sudo apt-get install ffmpeg
```

8 安装HTTP服务器(Apache, mod_wsgi)

```
$ sudo apt-get install apache2
```

如在安装过程中出现以下warning:

```
* Starting web server apache2
AH00558: apache2: Could not reliably determine the server's
fully qualified domain name, using 127.0.1.1.
Set the 'ServerName' directive globally to suppress this message
```

请修改 /etc/apache2/apache2.conf，并在文件中添加如下内容：

```
ServerName localhost
```

修改完毕后保存文件，重启apache2

```
$ sudo service apache2 restart
```

使用以下命令安装mod_wsgi:

```
$ sudo apt-get install libapache2-mod-wsgi-py3
```

9 安装Mysql数据库

```
$ sudo apt-get install python-dev libpython-dev libpq-dev  
$ sudo apt-get install libevent-dev libmysqlclient-dev  
$ sudo apt-get install mysql-server  
$ sudo pip3 install mysqlclient==1.3.6
```

安装mysql-server过程中需自行设置mysql root用户的密码

10 安装Mysql管理工具及创建数据库

如果开发人员精通于使用命令行操作mysql数据库，那么此节可以跳过

10.1 安装phpmyadmin

通过apt-get命令可以快速安装phpmyadmin

```
$ sudo apt-get install phpmyadmin apache2-utils
```

在安装过程中，需要输入一些参数配置，大致有如下几步：

- 选择Apache2服务器
- phpmyadmin dbconfig-common 选择'YES'
- 输入Mysql root用户密码
- 设置phpmyadmin登录密码

安装完成后，在apache配置文件中添加phpmyadmin配置：

```
$ sudo nano /etc/apache2/apache2.conf
```

向文件中添加如下内容：

```
Include /etc/phpmyadmin/apache.conf
```

重启apache2服务

```
$ sudo service apache2 restart
```

10.2 创建数据库

phpmyadmin安装完成后，即可以通过浏览器访问

```
http://localhost/phpmyadmin
```

创建名为 `alattng` 的数据库，字符集选用 `utf8(utf8_general_ci)`

11 代码克隆及配置

以下命令均在虚拟环境中执行，在执行命令前，请确保已经执行了11.1节的激活项目虚拟环境操作

11.1 激活项目虚拟环境

```
$ source ~/venv/bin/activate
```

使用上面的命令激活项目的虚拟环境后，命令行的行首将显示虚拟环境名，例如：

```
(venv)user@vm: ~$
```

提示：可使用`deactivate`命令退出项目虚拟环境

11.2 克隆项目

```
(venv)user@vm: ~$ git clone https://github.com/alattng/alattng.git
```

11.3 配置代码

代码克隆完成后，进入`alattng`项目目录，切换到`dev`分支，接着再进入`alattng`配置文件目录：

```
(venv)user@vm: ~$ cd lattng
(venv)user@vm: ~$ git checkout dev
(venv)user@vm: ~$ cd lattng
```

- 将目录内的 `settings_copy_for_reference.py` 文件复制为 `settings.py`
- 将目录内的 `wsgi_copy_for_reference.py` 文件复制为 `wsgi.py`

- 修改 settings.py 文件中的数据库相关配置, 因为开发环境DEBUG为True, 所以修改上半部分配置即可:

```
DEFAULT_DB = {
    'ENGINE': 'django.db.backends.mysql',
    'NAME': 'alattning', #10.2节创建的数据库名称
    'HOST': 'localhost',
    'PORT': '3306',
    'USER': 'root',
    'PASSWORD': '第9步中设置的数据密码',
}
```

11.4 安装项目依赖包

```
(venv)user@vm: ~$ sudo apt-get install libjpeg-dev
(venv)user@vm: ~$ cd ~/alattning
(venv)user@vm: ~$ pip3 install -r requirements.txt
(venv)user@vm: ~$ pip install -r requirements.txt
```

11.5 项目数据库及数据初始化

```
(venv)user@vm: ~$ python manage.py makemigrations
(venv)user@vm: ~$ python manage.py migrate
(venv)user@vm: ~$ python manage.py createsuperuser //回车后根据提示创建超管用户
```

11.6 启动项目服务

```
(venv)user@vm: ~$ python manage.py runserver
```

如果在启动项目时出现如下错误:

```
cannot import name 'GEOSEException'
```

请将 alattning/settings.py 中 INSTALLED_APPS 里的 django.contrib.gis 注释掉:

```
django.contrib.gis ==> # django.contrib.gis
```

然后重新执行启动命令。

项目启动后可通过浏览器访问系统首页：

```
http://127.0.0.1:8000
```

11.7 使用apache提供web服务

本节仅供参考，无需在虚拟机内操作，开发环境中不需要使用web服务器提供的服务

编辑apache配置文件：

```
$ sudo nano /etc/apache2/sites-available/000-default.conf
```

将如下内容添加到000-default.conf的“DocumentRoot /var/www/html”之后（请自行替换内容里的'bob'字符串）：

```
Alias /static /home/bob/alatting/static
<Directory /home/bob/alatting/static>
    Require all granted
</Directory>
<Directory /home/bob/alatting/alatting>
    <Files wsgi.py>
        Require all granted
    </Files>
</Directory>

WSGIDaemonProcess alatting python- path=/home/bob:/home/bob
/venv/lib/python3.4/site-packages
WSGIProcessGroup alatting
WSGIScriptAlias / /home/bob/alatting/alatting/wsgi.py
```

保存重启：

```
$ sudo service apache2 restart
```

访问地址：

```
$ http://localhost/poster/1
```

至此虚拟机环境搭建完成

配置Windows开发环境

上面介绍了测试环境的搭建过程，如果开发人员更擅长使用Ubuntu桌面环境，可以不阅读此章节，直接在Ubuntu中安装Pycharm，进行项目开发。

1 安装Pycharm IDE

在windows系统中安装 PyCharm Professional Edition 4.5.4.exe

2 IDE配置

- 进入菜单 'Tools' --> 'Start SSH Session' – 'Edit Credential', 打开SSH Session编辑窗口，输入虚拟机各项信息，并保存
- 进入菜单 'Tools' --> 'Deployment' --> 'Browse Remote Host', 添加新的远程主机。
- 添加主机信息，Type选择SFTP，输入SFTP Host IP地址，Port为22，Root Path为 /home/{用户目录}/alattng
- 从远程虚拟主机里打开项目源码目录， 双击文件即可进行编辑开发。

3 提交源码

由于源码保存在Ubuntu虚拟机中，所以需进入虚拟机内的项目目录下进行代码提交相关操作