

使用 Vagrant 打造跨平台开发环境



Vagrant 是一款用来构建虚拟开发环境的工具，非常适合 php/python/ruby/java 这类语言开发 web 应用，“代码在我机子上运行没有问题”这种说辞将成为历史。

我们可以通过 Vagrant 封装一个 Linux 的开发环境，分发给团队成员。成员可以在自己喜欢的桌面系统（Mac/Windows/Linux）上开发程序，代码却能统一在封装好的环境里运行，非常霸气。

安装步骤

1. 安装 VirtualBox

虚拟机还是得依靠 VirtualBox 来搭建，免费小巧。

下载地址：<https://www.virtualbox.org/wiki/Downloads>

* 虽然 Vagrant 也支持 VMware，不过 VMware 是收费的，对应的 Vagrant 版本也是收费的

2. 安装 Vagrant

下载地址：<http://downloads.vagrantup.com/> 根据提示一步步安装。

此外，还得下载官方封装好的基础镜像：

Ubuntu precise 32 VirtualBox <http://files.vagrantup.com/precise32.box>

Ubuntu precise 64 VirtualBox <http://files.vagrantup.com/precise64.box>

如果你要其他系统的镜像，可以来这里下载：<http://www.vagrantbox.es/>

* 鉴于国内网速，我们将下载镜像的步骤单独剥离出来了

3. 添加镜像到 Vagrant

假设我们下载的镜像存放路径是 `~/box/precise64.box`，在终端里输入：

```
$ vagrant box add hahaha ~/box/precise64.box
```

`hahaha` 是我们给这个 box 命的名字，`~/box/precise64.box` 是 box 所在路径

4. 初始化开发环境

创建一个开发目录（比如：`~/dev`），你也可以使用已有的目录，切换到开发目录里，用 `hahaha` 镜像初始化当前目录的环境：

```
$ cd ~/dev # 切换目录
$ vagrant init hahaha # 初始化
$ vagrant up # 启动环境
```

你会看到终端显示了启动过程，启动完成后，我们就可以用 SSH 登录虚拟机了，剩下的步骤就是在虚拟机里配置你要运行的各种环境和参数了。

```
$ vagrant ssh # SSH 登录
$ cd /vagrant # 切换到开发目录，也就是宿主机上的 `~/dev`
```

`~/dev` 目录对应虚拟机中的目录是 `/vagrant`

Windows 用户注意：Windows 终端并不支持 ssh，所以需要安装第三方 SSH 客户端，比如：Putty、Cygwin 等。

5. 其他设置

Vagrant 初始化成功后，会在初始化的目录里生成一个 `Vagrantfile` 的配置文件，可以修改配置文件进行个性化的定制。

Vagrant 默认是使用端口映射方式将虚拟机的端口映射本地从而实现类似 `http://localhost:80` 这种访问方式，这种方式比较麻烦，新开和修改端口的时候都得编辑。相比较而言，host-only 模式显得方便多了。打开 `Vagrantfile`，将下面这行的注释去掉（移除 `#`）并保存：

```
config.vm.network :private_network, ip: "192.168.33.10"
```

重启虚拟机，这样我们就能用 `192.168.33.10` 访问这台机器了，你可以把 IP 改成其他地址，只要不产生冲突就行。

6. 打包分发

当你配置好开发环境后，退出并关闭虚拟机。在终端里对开发环境进行打包：

```
$ vagrant package
```

打包完成后会在当前目录生成一个 `package.box` 的文件，将这个文件传给其他用户，其他用户只要添加这个 box 并用其初始化自己的开发目录就能得到一个一模一样的开发环境了。

7. 常用命令

```
$ vagrant init # 初始化
$ vagrant up # 启动虚拟机
$ vagrant halt # 关闭虚拟机
$ vagrant reload # 重启虚拟机
$ vagrant ssh # SSH 至虚拟机
$ vagrant status # 查看虚拟机运行状态
$ vagrant destroy # 销毁当前虚拟机
```

更多内容请查阅官方文档 <http://docs.vagrantup.com/v2/cli/index.html>

8. 注意事项

使用 Apache/Nginx 时会出现诸如图片修改后但页面刷新仍然是旧文件的情况，是由于静态文件缓存造成的。需要对虚拟机里的 Apache/Nginx 配置文件进行修改：

```
# Apache 配置添加：
EnableSendfile off
```

```
# Nginx 配置添加：
sendfile off;
```