**Vagrant工具使用说明**

# 1 Vagrantfile配置说明

在开发目录下有一个文件Vagrantfile，里面包含有大量的配置信息，主要包括三个方面：Vagrant的一些基础配置、虚拟机的配置、SSH配置。Vagrant是使用Ruby开发的，所以它的配置语法也是R。uby的，但是我们没有学过Ruby的人还是可以跟着它的注释知道怎么配置一些基本项的配置。

**① box设置**

|  |
| --- |
| config.vm.box = "base" |

上面这配置展示了Vagrant要去启用那个box作为系统，也就是上面我们输入vagrant init Box名称时所指定的box，如果沒有输入box名称的话，那么默认就是base，VirtualBox提供了VBoxManage这个命令行工具，可以让我们设定VM，用modifyvm这个命令让我们可以设定VM的名称和内存大小等等，这里说的名称指的是在VirtualBox中显示的名称，我们也可以在Vagrantfile中进行设定，在Vagrantfile中加入如下这行就可以设定了：

|  |
| --- |
| config.vm.provider "virtualbox" do |v|  v.customize ["modifyvm", :id, "--name", "astaxie", "--memory", "512"]  end |

这行设置的意思是调用VBoxManage的modifyvm的命令，设置VM的名称为astaxie，内存为512MB。你可以类似的通过定制其它VM属性来定制你自己的VM。

**② 网络设置**

Vagrant有两种方式来进行网络连接，一种是host-only(主机模式)，意思是主机和虚拟机之间的网络互访，而不是虚拟机访问internet的技术，也就是只有你一個人自High，其他人访问不到你的虚拟机。另一种是Bridge(桥接模式)，该模式下的VM就像是局域网中的一台独立的主机，也就是说需要VM到你的路由器要IP，这样的话局域网里面其他机器就可以访问它了，一般我们设置虚拟机都是自high为主，所以我们的设置一般如下：

config.vm.network :private\_network, ip: "11.11.11.11"

这里我们虚拟机设置为hostonly，并且指定了一个IP，IP的话建议最好不要用192.168..这个网段，因为很有可能和你局域网里面的其它机器IP冲突，所以最好使用类似11.11..这样的IP地址。

选择网络模式，在Vagrantfile配置文件把前面的#号去掉。

|  |  |
| --- | --- |
| config.vm.network :forwarded\_port, guest: 80, host: 8080 | guest: 80 表示虚拟机中的80端口， host: 8080 表示映射到宿主机的8080端口。  较为常用是端口映射，就是将虚拟机中的端口映射到宿主机对应的端口直接使用 |
| config.vm.network :private\_network, ip: "192.168.1.104" | 如果需要自己自由的访问虚拟机，但是别人不需要访问虚拟机，可以使用private\_network，并为虚拟机设置IP。  192.168.1.104 表示虚拟机的IP，多台虚拟机的话需要互相访问的话，设置在相同网段即可 |
| config.vm.network :public\_network | 如果需要将虚拟机作为当前局域网中的一台计算机，由局域网进行DHCP |

**③ hostname设置**

hostname的设置非常简单，Vagrantfile中加入下面这行就可以了：

|  |
| --- |
| config.vm.hostname = "go-app" |

设置hostname非常重要，因为当我们有很多台虚拟服务器的时候，都是依靠hostname來做识别的，例如Puppet或是Chef，都是通过hostname来做识别的，既然设置那么简单，所以我们就別偷懒，设置一个。

**④ 同步目录**

默认情况下，当前的工作目录，会被映射到虚拟机的 /vagrant 目录，当前目录下的文件可以直接在 /vagrant 下进行访问，当然也可以在通过 ln 创建软连接，如

|  |
| --- |
| ln -fs /vagrant/wwwroot /var/www |

我们上面介绍过/vagrant目录默认就是当前的开发目录，这是在虚拟机开启的时候默认挂载同步的。我们还可以通过配置来设置额外的同步目录：

|  |
| --- |
| config.vm.synced\_folder "/Users/astaxie/data", "/vagrant\_data" |

在win7如：

|  |
| --- |
| config.vm.synced\_folder "D:/www", "/home/wwwroot/default" |

上面这个设定，第一个参数是主机的目录，第二个参数是虚拟机挂载的目录

**⑤ 端口转发**

config.vm.network :forwarded\_port, guest: 80, host: 8080

上面这句配置可厉害了，这一行的意思是把对host机器上8080端口的访问请求forward到虚拟机的80端口的服务上，例如你在你的虚拟机上使用nginx跑了一个Go应用，那么你在host机器上的浏览器中打开http://localhost:8080时，Vagrant就会把这个请求转发到VM里面跑在80端口的nginx服务上，因此我们可以通过这个设置来帮助我们去设定host和VM之间，或是VM和VM之间的信息交互。

安装web服务器时，如果无法访问，检查下linux开没开启80端口

|  |
| --- |
| vi /etc/sysconfig/iptables  -A INPUT -m state –state NEW -m tcp -p tcp –dport 80 -j ACCEPT（允许80端口通过防火墙）  -A INPUT -m state –state NEW -m tcp -p tcp –dport 3306 -j ACCEPT（允许3306端口通过防火墙）  特别提示：很多网友把这两条规则添加到防火墙配置的最后一行，导致防火墙启动失败，正确的应该是添加到默认的22端口这条规则的下面 |

修改完Vagrantfile的配置后，记得要用vagrant reload命令来重启VM之后才能使用VM更新后的配置

# 2 vagrant命令说明

|  |  |
| --- | --- |
| vagrant box add 名称 box路径 | 添加box的操作 |
| vagrant init 名称 | 初始化box的操作 |
| vagrant up | 启动虚拟机的操作 |
| vagrant ssh | 登录拟机的操作，也可以指定ID号登陆 |
| vagrant box list | 显示当前已经添加的box列表 |
| vagrant box remove 名称 | 删除相应的box |
| vagrant destroy | 停止当前正在运行的虚拟机并销毁所有创建的资源 |
| vagrant halt | 关机 |
| vagrant package | 打包命令，可以把当前的运行的虚拟机环境进行打包 |
| vagrant plugin | 用于安装卸载插件 |
| vagrant reload | 重新启动虚拟机，主要用于重新载入配置文件 |
| vagrant ssh-config | 输出用于ssh连接的一些信息 |
| vagrant status | 获取当前虚拟机的状态 |
| vagrant suspend | 挂起当前的虚拟机 |
| vagrant resume | 恢复前面被挂起的状态 |
| vagrant global-status | 查看所有虚拟机的ID号 |

# 3 打包输出

启动虚拟机

**vagrant up**

进入虚拟机

**vagrant ssh**

删除一个文件(可能名字不一样)

**rm /etc/udev/rules.d/75-persistent-net-generator.rules**

退出虚拟机

**exit**

打包虚拟机

**vagrant package**