



安装 Vagrant 和 Debian

Vagrant 是一个管理虚拟机 (VM) 的程序。基于名为 `a` 的配置文件 `Vagrantfile`，它可以下载和配置磁盘映像（称为框），并调用其他程序来运行它们。Vagrant 本身不会运行虚拟机，因此您需要另一个程序（例如 `virtualbox`）来运行它。

在您自己的机器上安装

要在您自己的计算机上使用 vagrant（推荐），请按照以下步骤操作：

- 访问<https://www.vagrantup.com/downloads>并下载适合您操作系统的 vagrant 版本。支持 Windows、Mac OS 和常见版本的 Linux。
- 从<https://www.virtualbox.org/wiki/Downloads>下载并安装 Virtualbox。
- 重新启动您的机器。

如果您使用的是 Linux，您当然也可以从发行版存储库安装程序。Vagrant 的开发人员实际上建议不要这样做，因为他们声称某些发行版打包了过时的版本，但这是你的选择。

配置盒子

接下来，您将使用 Debian linux 配置虚拟机，这是我们将在本单元中使用的 Linux 发行版。

- 在某处创建一个空文件夹。
- 在该文件夹中，创建一个名为 `Vagrantfile` 的文件（大写，无扩展名）并向其中添加以下行 - 或者直接从[此处](#)下载该文件：

```
Vagrant.configure("2") do |config|
  config.vm.box = "generic/debian12"
  config.vm.synced_folder ".", "/vagrant"

  config.vm.provision "shell", inline: <<-SHELL
    echo "Post-provision installs go here"
  SHELL
end
```

这个配置文件实际上是 ruby 编程语言中的一个脚本，但你不需要学习它来使用 vagrant。让我们看看它做了什么。

- `config.vm.box` 选择要使用的虚拟机映像，或者用 vagrant 的话说就是框。您可以在<https://app.vagrantup.com/boxes/search>上查看可用列表。

- `config.vm.synced_folder` 在来宾（虚拟机）和主机（您的计算机）之间设置共享文件夹。
- 首次下载并安装该盒子时，会 `config.vm.provision` 运行配置命令。这些命令在虚拟机上以 `root` 身份运行，在本例中，我们使用 `apt` 包管理器（我们将在稍后讨论）来安装包 `git`。
- 该 `<<-SHELL` 构造称为“此处文档”，是某些编程语言中编写多行字符串的一种方式。它告诉 `ruby` 将结束关键字 `SHELL`（任意）之前的所有内容视为字符串，该字符串可以包含多行。

奔跑的流浪者

- 在包含 `Vagrantfile` 的文件夹中打开终端。如果您使用的是 Windows，则 Windows CMD 和 Linux 终端的 Windows 子系统都可以同样有效地实现此目的。
- 运行命令 `vagrant up`。这将启动在当前文件夹中配置的虚拟机，如果尚未下载和配置虚拟机（就像您 `up` 第一次运行时的情况一样），那么它也会为您执行此操作。
- 当 `vagrant` 告诉您机器正在运行时，运行 `vagrant ssh` 以登录到您的虚拟机。如果它要求您输入密码，请使用 `vagrant`。
- 您现在应该看到虚拟机提示符 `vagrant@debian12:~$`。尝试该命令 `ls /` 并检查顶级文件夹中是否有一个名为 `vagrant` 的文件夹，以及名称类似于 `usr` 和 `bin` 的系统文件夹。

在此期间您可能会遇到两种错误 `vagrant up`：

- 如果 `vagrant` 抱怨它找不到提供程序，那么您可能没有安装 `virtualbox`，或者安装后没有重新启动。
- 如果您收到有关虚拟机管理程序的一些奇怪的崩溃或错误消息，请参阅页面 <https://www.vagrantup.com/docs/installation> 的说明，运行多个虚拟机管理程序部分。基本上，当另一个程序已经在使用您的处理器的虚拟化子系统时，您无法运行 `vagrant`，并且该页面给出了如何关闭另一个程序的说明。

干净地关闭

要退出虚拟机，请键入 `exit` 这将使您返回到主机上的 `shell`。在主机上，`vagrant halt` 彻底关闭虚拟机。

向您自己保证，如果您一直在使用 `Vagrant`，那么在关闭计算机之前您将始终执行此操作！

在实验室机器上运行

`Vagrant` 已安装在 MVB 2.11 的实验室机器上，因此您可以远程登录并从那里启动盒子。这将为您的提供与在自己的计算机上运行它时完全相同的 `Debian` 环境，每个人都应该尝试一下。如果由于某种原因你无法在你的机器上运行 `Vagrant`，那么只要你有互联网连接，你仍然应该能够在实验室机器上运行它。

ssh lab 首先，我们连接到实验室机器：打开终端并运行您在上一个练习中在 SSH 上配置的命令。

在实验室机器上，我们需要创建一个文件夹并加载 Vagrantfile，如上所示，但让我们从单元网页下载 Vagrantfile，而不是输入它。运行以下 shell 命令（第三个开始的命令 wget 必须全部在一行上，即使您的 Web 浏览器已添加换行符）：

```
mkdir softwaretools
cd softwaretools
wget https://raw.githubusercontent.com/cs-
uob/COMSM0085/master/exercises/part1/src/resources/Vagrantfile
```

您可以将顶部文件夹（软件工具）命名为您喜欢的任何名称，并将其放在您想要的任何位置。

vagrant up 您现在可以从该文件夹内运行 vagrant ssh。

Advanced note

当您执行时 vagrant up，vagrant 会在内部将客户机上的端口 22（sshd 客户机上正在侦听的端口）连接到主机上的端口 2222。当您配置 vagrant 计算机时，这会在主机上创建密钥对并将公钥加载到来宾中。私钥实际上位于 .vagrant/machines/default/virtualbox/private_key 主机上的文件中，而公钥位于 /home/vagrant/.ssh/authorized_keys 来宾上。那么 vagrant ssh 所做的就是启动 ssh -i KEYFILE vagrant@localhost -p 2222。

关于实验室机器的警告 - 仔细阅读！

实验室计算机上主目录中的文件存储在网络文件夹中，因此无论登录哪台实验室计算机，您都可以看到相同的文件；它们也会自动备份。

如果许多学生在他们的主文件夹中创建大量虚拟机，这将占用大量空间，而且速度会很慢：通过网络共享运行操作系统会导致带宽和延迟问题。

相反，IT 部门在实验室计算机上配置了 vagrant，将虚拟机存储在 /tmp 每台计算机本地的文件夹中。这意味着：

- 如果您登录到另一台实验室计算机，您的虚拟机将会消失。
- 如果您登录到同一台实验室计算机，但自您上次登录以来该计算机已重新启动，您的虚拟机将会消失。
- 您的虚拟机以及您存储在虚拟机本身中的任何文件都不会被备份。

这并不像看上去那么严重，因为虚拟机就是这样工作的：如果虚拟机不可用，vagrant 就会下载并配置它。因此，对于您想要在实验室计算机中的虚拟机上安装的任何软件，您应该将安装命令写入 Vagrantfile 中的配置脚本中，以便下次 Vagrant 设置虚拟机时重新安装该软件。我们很快就会学会如何做到这一点。

但是，这仍然会留下您在虚拟机本身上创建的文件，例如您为本单元中的练习创建的文件。基本警告是，当重建虚拟机时，主目录中的所有文件都将丢失。这就是为什么我们设置了一个共享文件

夹，您可以像 `/vagrant` 在虚拟机上一样访问该文件夹，该文件夹映射到主机上包含 Vagrantfile 的文件夹。由于它存储在实验室计算机上的主文件夹下，因此它位于网络文件存储中，因此可以从所有实验室计算机进行备份和使用。

因此，每当您登录到实验室计算机上的虚拟机进行某些工作时，您 `cd /vagrant` 都应该使用它而不是您的主文件夹来保存您不想丢失的任何文件。如果你在自己的计算机上运行 vagrant，那么虚拟机中的任何内容都不会被删除，除非你给 Vagrant 一个命令来自行销毁或重建虚拟机。