

Virtual Machine Setup Instructions

Spring 2019

Overview

Linux is the required development environment for this course. Linux is an open source, Unix-like operating system that is free and commonly used in industry and academia. It is a complete operating system, meaning you *could* install it alongside, or in place of, Windows or MacOS on your laptop. However, to minimize this course's impact on your machine, we expect you to install Linux on a *virtual machine* (VM). A VM simulates another computer, a *guest machine* with its own operating system, running on your laptop from your usual operating system, the *host*.

The virtualization software we recommend for this course is the free, open source combination of VirtualBox and Vagrant. VirtualBox is the software capable of running guest machines on your host laptop. It is highly capable and configurable, perhaps to a fault. To avoid the lengthy manual process of configuring a VM and installing its operating system, we will use Vagrant to automate and abstract away the complexities. These tools are commonly used by teams in industry to ensure consistency between software engineers. Similarly, this course uses these tools to ensure your development environment is consistent with everyone else's and the course staff's.

To prepare yourself for success in this course, please complete the following steps whose detailed instructions follow:

1. Install VirtualBox and Vagrant
2. Clone and **vagrant** up the Course VM Project
3. Generate SSH keys for your VM and add them to GitHub account
4. Exit and Halt the VM

1. Installing VirtualBox and Vagrant

Please follow the instructions specific to your laptop's host operating system below.

Windows

- Install git with Bash
 - Go to the downloads page <https://git-scm.com/downloads>
 - Download the latest version for windows
 - Follow its install instructions accepting defaults
- Install VirtualBox 6.0+
 - Go to the downloads page <https://www.virtualbox.org/wiki/Downloads>
 - Select the Windows hosts link
 - Open the downloaded .exe file and follow its instructions accepting defaults
- Install Vagrant 2.2+
 - Go to the downloads page <https://www.vagrantup.com/downloads.html>
 - Select the Windows 64-bit link
 - Open the downloaded .msi file and follow its instructions accepting defaults
 - Reboot your machine

MacOS

MacOS will, by default, try to block your installing of an app not on the app store for security purposes. You will need to follow the instructions necessary during the process to allow an exception in your Mac's security preferences to continue forward with the installs.

- Install VirtualBox 6.0+
 - Go to the downloads page <https://www.virtualbox.org/wiki/Downloads>
 - Select the OS X hosts link
 - Open the downloaded .dmg file and double click the installer. If you are given instructions about the install not working due to security settings, please follow the instructions provided or do a search for your specific version of MacOS's instructions for getting around the security restriction.
- Install Vagrant 2.2+
 - Go to the downloads page <https://www.vagrantup.com/downloads.html>
 - Select the MacOS 64-bit link
 - Open the downloaded .dmg file and follow its instructions accepting defaults
 - Reboot your machine

Linux

Are you already running Linux on your laptop? Great! We will still encourage you to use VirtualBox and Vagrant like your peers on Windows and Mac. Why? Primarily to help you avoid accidentally mucking up your personal Linux host operating systems for our purposes. We expect an Ubuntu 18.04 Server environment and your running it virtualized will help

our course staff better serve you by ensuring they're familiar with your setup, too. Finally, the ability to manage a virtualized environment is a valuable skill to have in your repertoire.

For Ubuntu 18.04, Debian Sid, and newer versions of either, running `sudo apt install vagrant virtualbox` should be enough. This should also work for older releases of those distributions, but be aware that we cannot support problems encountered when using the accompanying older versions of Vagrant and Virtualbox.

2. Clone and `vagrant` up the Course VM Project

From Mac or Linux, open a Terminal, from Windows, start a new Git Bash shell. Clone the following git repository in the directory of your choosing. Your home directory, that your terminal originally opens in, is a perfectly fine place:

```
git clone https://github.com/comp524-s19/unc-comp-vm.git
```

Once the repository has cloned, continue with the following commands that will change directories into the project and bring *up* the VM. The second command will take some time to complete as it must download the virtual machine image, which is ~2.1 gigabytes, and initialize it.

```
cd unc-comp-vm
vagrant up
```

Once this has completed, your class virtual machine is running. You can login to the virtual machine with the command:

```
vagrant ssh
```

Upon doing so, your terminal will now be inside the virtual machine and you're ready to continue on to the next section.

3. Initialize git and SSH keys for your VM

While still logged in to your VM from the steps above, you should initialize your VM's git information so that when you make commits in git your name and email address are associated with them. You should go ahead and register for GitHub if you have not already. The email address you will establish in the next steps should match your email address on GitHub.

```
git config --global user.name "FirstName LastName"
git config --global user.email "the-email-@address-you-use-on-github.com"
```

[Optional] Key Authentication with GitHub

Optionally, you can also enable key-based authentication with GitHub. While the default password-approach should be sufficient for the purposes of this class, many companies only permit key-based authentication. To set this up, you will want to generate a key pair that will authenticate you with GitHub. The following command will ask you a series of three

questions, just press enter three times for now. The default location for the key is best and you do not need a passphrase to protect the key for now.

```
ssh-keygen -t rsa -b 4096 -C "the-email-address-you-use-on-github.com"
```

Finally, you will need to add the public key you just generated to GitHub in order for authentication to work. To do so, run the following command to print it to your screen. Select the output of this command, copy it (on Windows' Git Bash copy happens every time you select text with the mouse, on Mac's terminal press Command C), and continue with the next instructions:

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

Now, log in to GitHub and navigate to your user's settings. Select "SSH and GPG keys" in the sidebar. Click the green "New SSH key" button. Title the key "UNC CS VM" and paste in your public key. It should begin with **ssh-rsa**.

4. Exit and Halt the VM

When you are done working on course related projects, you should halt the virtual machine correctly. Simply closing your terminal will not halt the virtual machine, it will continue running in the background until you do.

```
exit
```

To halt the virtual machine, first you'll need to **exit** it if you are logged into it. Simply issuing the **exit** command will bring you back to your host machine's directory that you setup the course VM in. Then, you'll need to run the following vagrant command:

```
vagrant halt
```

This powers down the virtual machine so that is not using any resources other than storage space for its data. To get started with course work again, you'll simply need to open a terminal, change your directory to be where you setup the VM, and type **vagrant up**. The second and subsequent times you issue the **vagrant up** command will go much faster than the first time because it is already setup and simply needs to be booted.