Penn Engineering
ONLINE LEARNING

Project 0: Setting Up Your Development Environment

In the class, we will be using a Vagrant Virtual Machine to build and test our projects. Please follow the directions below to download and install the virtual machine on your host computer as soon as possible.

Please go through our TA demos on Vagrant and Git before starting this project. Refer to the Vagrant FAQs if you need more help. If you are still having trouble setting up your development environment, please post on Piazza proactively with any concerns or problems because you will not be able to start work on the next two projects until it is set up!

Your code will be validated by a custom-designed autograder.

Setting up the Virtual Machine (VM)

- The first part of this assignment is to set up the virtual machine (VM) you will use for the rest of the course.
- All grading will be done on the VM so we **highly recommend** you use the provided VM and you **must** test your code on the virtual machine before project submissions.
- The VagrantFile can be downloaded from Coursera under Resources → Project 0 VagrantFile.
- Note: The install process requires the use of a command line interface (CLI) from your host operating system. You will be using a CLI very frequently as part of your work in this class, so we recommend finding a setup that you are comfortable with.

Step 1: Install Vagrant

- Vagrant is a tool for automatically configuring a VM using instructions given in a single “Vagrantfile.” You need to install Vagrant using the correct download link for your computer here: https://www.vagrantup.com/downloads.html.

Step 2: Install VirtualBox

- VirtualBox is a VM provider (hypervisor).

- You need to install VirtualBox using the correct download link for your computer here: https://www.virtualbox.org/wiki/Downloads. The links are under the heading "VirtualBox X.X.X platform packages."

Step 3: Provision virtual machine using Vagrant

- Run the command `vagrant up` in your target install directory to start the VM and provision it according to the Vagrantfile.

Vagrant Commands

Note: The following commands will allow you to stop the VM at any point (such as when you are done working on an assignment for the day):

- `vagrant suspend` will save the state of the VM and stop it.
- `vagrant halt` will gracefully shutdown the VM operating system and power down the VM.
- `vagrant destroy` will remove all traces of the VM from your system. If you have important files saved on the VM (like your assignment solutions) **DO NOT** use this command.

Step 4: Test SSH to VPN

- Run `vagrant ssh` from your terminal. This is the command you will use every time you want to access the VM. If it works, your terminal prompt will change to `vagrant@cit595Dev:~$`. All further commands will execute on the VM.

Shared Directory

- From the home directory denoted as ($HOME) or the tilde, ~, you can run `cd ../../vagrant` to get to the course directory that's shared between your regular OS and the VM. This location serves as the root of your shared directory.
- Vagrant is especially useful because of this shared directory structure. You don't need to copy files to and from the VM. Any file or directory in the host OS's directory

where the `Vagrantfile` is located is automatically shared between your computer and the virtual machine.

Exiting Vagrant

- The command `logout` will stop the SSH connection at any point and return you to your terminal.

Clone Git Repository

- Each of you will be assigned a git repository. To get the repository, you need to create a **new Github account** using your seas.upenn.edu email. You can skip this step if you already have a Github account using your seas.upenn.edu email. To standardize, we will not use Github accounts from non-Penn emails.
- Log into your Vagrant instance.
- The command “git clone https://github.com/CIS380/20sp-cit595-githubID will clone the files onto your machine. Replace “githubID” with your GitHub ID. It should prompt you for your user name and password, use your Github username and password then when you push and pull it will ask you do this again.
- If you want to setup an ssh key so you don't have to keep typing in your credentials you should check out this link and post on Piazza, if necessary.

To-Do List:

1. Create a project directory “project0”.
2. Within this directory, create a README describing in one paragraph what you hope to learn from this class and your experiences setting up the Git repository and Vagrant.
3. Push this README in the project0 directory to Github.

No Coursera submission is necessary. **You need to both ‘commit’ and ‘push’ your README file to the project0 directory.**

To validate that you have pushed successfully, go to github.com, log in using our seas.upenn.edu email, locate your code repository, and verify that the README file in the project0 directory.

This assignment is worth 5% of your total grade for this class.

Penn Engineering
ONLINE LEARNING

Grading Rubric:

In this project, we will be checking that you have:

1. Successfully cloned your Git repository to your local machine.
2. Created a ‘project0’ directory.
3. Written a README file inside the ‘project0’ directory.
4. Pushed the new directory and README into their code repository.

The autograder will assign points based on the following rubric:

- 5 points: README file is in the ‘project0’ directory.
- 3 points: Either the README file or ‘project0’ directory was created but not both, or the README file is not inside the ‘project0’ directory.
- 0 points: Neither the ‘project0’ directory nor README file were found.

Note: We will not be validating that you are cloning your repository within your Vagrant development environment. However, we strongly encourage you to do so now since the rest of the semester will depend on your Vagrant setup.