

Boot Camp Homework 1: Development Environment Setup

The goal of this homework is to ensure that you have a functioning Rails development environment with which to complete the remaining Boot Camp homeworks. To accomplish this goal, you must set up a development environment as per the following instructions.

Before I jump into the instructions, I need to clear up a little terminology. In this and future homeworks, you will be using a *virtual machine* (VM). A VM is where one OS is run inside of another OS (rather than directly on a physical machine). The parent OS is called the *host* OS. For example, imagine you have a Windows computer – that’ll be the host OS. You can use the VirtualBox software to install a Linux OS within the host Windows OS and to run a Linux VM as if it were a Windows program.

With that bit out of the way, here are your instructions:

1. You will need to have a Bash shell with SSH client. OSX users have this software by default. Debian/Ubuntu Linux users have the shell, but may need to install the “openssh-client” package (if it’s not already installed by default). Windows users should download and install Git for the Windows platform (<http://git-scm.com/download/win>), which comes with a Bash shell and SSH client.
2. Download and install VirtualBox (<https://www.virtualbox.org/>). You will be using this software to run an Ubuntu Linux virtual machine. This VM will house the majority of your Rails development tools. All students in this course must use an Ubuntu development environment. The rationale is that by having all students standardize on an environment, they will be better able to collaborate and learn from one another.
3. Download and install Vagrant (<https://www.vagrantup.com/>). Vagrant is used to package, distribute, and run custom-configured VMs. I have prepared a Vagrant “box” for you as you will see below.
4. Download and unzip the following file:

<http://www.cs.memphis.edu/~sdf/workspace.zip>

Extracting this file will create a “workspace” folder. (On Windows, you may have one workspace folder within another workspace folder. You only need the inner one.) The workspace folder should contain two files: “Vagrantfile” and (possibly hidden) “.provisioning_script.sh”. This workspace folder is where all your Rails projects for the course will go.

5. Launch a terminal. In OSX, this involves launching the “Terminal” app. In Windows, it involves launching “Git Bash”. I assume that Linux users need no clarification here.
6. In the terminal, change directory (using the “cd” command) to the workspace folder.
7. Run the following command to download and initialize a Vagrant box. Beware that this command (1) may take a long time to complete, (2) downloads a big file (~700MB), and (3) performs at least one processor-intensive compilation (of Ruby).

```
$ vagrant up
```

Once this command completes, you will have a running Ubuntu Linux VM (headless).

8. Run the following command to SSH into the Linux VM:

```
$ vagrant ssh
```

If you're prompted for a passphrase, just hit enter. If you're prompted for a password, enter "vagrant" (without the quotes).

You should see a command prompt that looks like this:

```
[vagrant@vagrant-ubuntu-trusty-64:~]
$
```

If you use the "ls -l" command, you will see a list of files in the current directory. You will see among them a "workspace" folder (actually a symbolic link to the folder "/vagrant").

9. Change directory (using the "cd" command) to the workspace folder.
10. Use Git to download an example project by entering the following command:

```
$ git clone https://github.com/sdflem/sample_app.git
```

You should now see a "sample_app" folder inside the workspace folder. Note that the workspace folder is synced with your host OS, so you should also see sample_app in your host file explorer. The reason for syncing this folder is that it will later enable you to use a GUI code editor to work on your code files.

11. Change directory (using the "cd" command) to the sample_app folder.
12. Run these two commands to set up the web app project (i.e., prepare it to run).

```
$ bundle install --without production
$ rake db:setup
```

The first command may take a while to complete because it downloads and installs a lot of Ruby gems.

13. The project comes with some automated tests. Run this command to execute the tests:

```
$ rake test
```

You should see that all the tests passed.

14. Take a screenshot that shows your desktop and terminal with the notification that all the tests passed. Save this screenshot to be submitted later.

15. Start up the Rails web app server with the following command (those are zeros):

```
$ rails s -b 0.0.0.0
```

You should see that the server has started without error. Note that this command will not "return" like other commands—that is the command prompt will not reappear until you kill the process (covered below).

16. Now open the following URL in a web browser:

<http://localhost:3000>

A "Welcome to the Sample App" web page should appear.

17. In the web app, log in with the following email/password:

example@railstutorial.org
foobar

You will see that it is a twitter-like app.

18. In the app, create a post that contains your university email name (e.g., "sdfmling rocks!").

19. Take a screenshot of your desktop with the web app showing and your post visible. Save this screenshot to be submitted later.

Congratulations! You've completed the main part of this homework.

To shut down everything:

20. Back in the terminal, type Ctrl-C to kill the Rails server.

21. Enter "exit" to logout of the VM.

22. Enter "vagrant halt" to shut down the VM.

Later, when you need to start the VM again, just enter "vagrant up" (should be much faster than last time) and use "vagrant ssh" to log in again.

To get credit for completing this homework, upload your two screenshots to the appropriate eCourseware dropbox before the posted deadline.