

Vagrant VM Installation

Installing the Vagrant VM for Intro to RDB

In Lessons 3 and 5 of **Introduction to Relational Databases**, you'll use a virtual machine (VM) to run a database server and a web app that uses it. The VM is a Linux server system that runs on top of your own computer. You can share files easily between your computer and the VM. [Click here for Vagrant documentation.](#)

Conceptual Overview

We're using the Vagrant software to configure and manage the VM. Here is a conceptual overview of many of the concepts related to virtual machines and vagrant.

<https://www.youtube.com/watch?v=djnqoEO2rLc>

Here are the tools you'll need to install to get it running:

Git

If you don't already have Git installed, [download Git from git-scm.com](#). Install the version for your operating system.

On Windows, Git will provide you with a Unix-style terminal and shell (Git Bash). (On Mac or Linux systems you can use the regular terminal program.)

You will need Git to install the configuration for the VM. If you'd like to learn more about Git, [take a look at our course about Git and Github](#).

VirtualBox

VirtualBox is the software that actually runs the VM. [You can download it from virtualbox.org, here.](#) Install the *platform package* for your operating system. You do not need the extension pack or the SDK. You do not need to launch VirtualBox after installing it.

Windows Users - VirtualBox

If you run into difficulties using VirtualBox 5 or higher then we recommend installing an earlier version (4.3.0).

Mac Users

If you encounter a problem running the command "vagrant up" on a Mac (found later in these instructions) the issue may be with the version of VirtualBox installed. Uninstall both VirtualBox and Vagrant and use the latest test build of VirtualBox for Mac found here — <https://www.virtualbox.org/wiki/Testbuilds> — and then install Vagrant as per the instructions below.

Error message on VirtualBox: "Failed to load VMMR0.r0 (VERR_VMM_SMAP_BUT_AC_CLEAR)."

Error message on Vagrant: "The guest machine entered an invalid state while waiting for it to boot. Valid states are 'starting, running'. The machine is in the 'poweroff' state. Please verify everything is configured properly and try again."

Ubuntu 14.04 Users:

If you are running Ubuntu 14.04, install VirtualBox using the Ubuntu Software Center, not the [virtualbox.org](#) web site. Due to a [reported bug](#), installing VirtualBox from the site may uninstall other software you need.

Vagrant

Vagrant is the software that configures the VM and lets you share files between your host computer and the VM's filesystem. [You can download it from vagrantup.com.](#) Install the version for your operating system.

Windows Note: The Installer may ask you to grant network permissions to Vagrant or make a firewall exception. Be sure to allow this.

Use Git/GitHub to fetch the VM configuration

Windows: Use the Git Bash program (installed with Git) to get a Unix-style terminal. Make sure to run as administrator.

Other systems: Use your favorite terminal program.

We have provided a GitHub repository that contains all of the code you will need for this course (and if you are enrolled in it, for the entire Full-Stack Nanodegree program).

1. Go here: <https://github.com/udacity/fullstack-nanodegree-vm>
2. Fork the repository (Click **Fork** in the top-right corner)
3. Now we want to clone the newly forked repository to your computer. We will want to use the HTTPS method of cloning. Click the clipboard icon (see image below) to copy the repo's path.

Clone with HTTPS ?

[Use SSH](#)

Use Git or checkout with SVN using the web URL.

`https://github.com/udacity/fullstack-nanod`



[Open in Desktop](#)

[Download ZIP](#)

4. Then from the terminal, run:

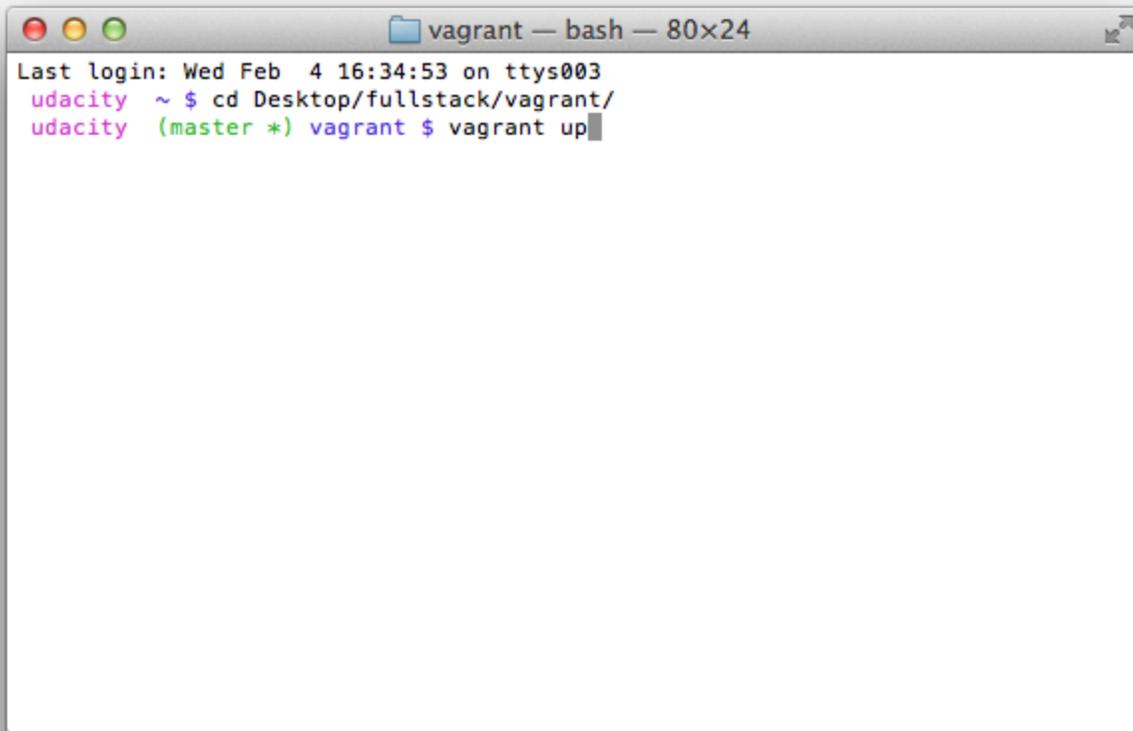
```
git clone PASTE_PATH_TO_REPO_HERE fullstack
```

This will give you a directory named **fullstack**. **Note:** you will want to paste the path you copied from step 3 into `PASTE_PATH_TO_REPO_HERE`.

Run the virtual machine!

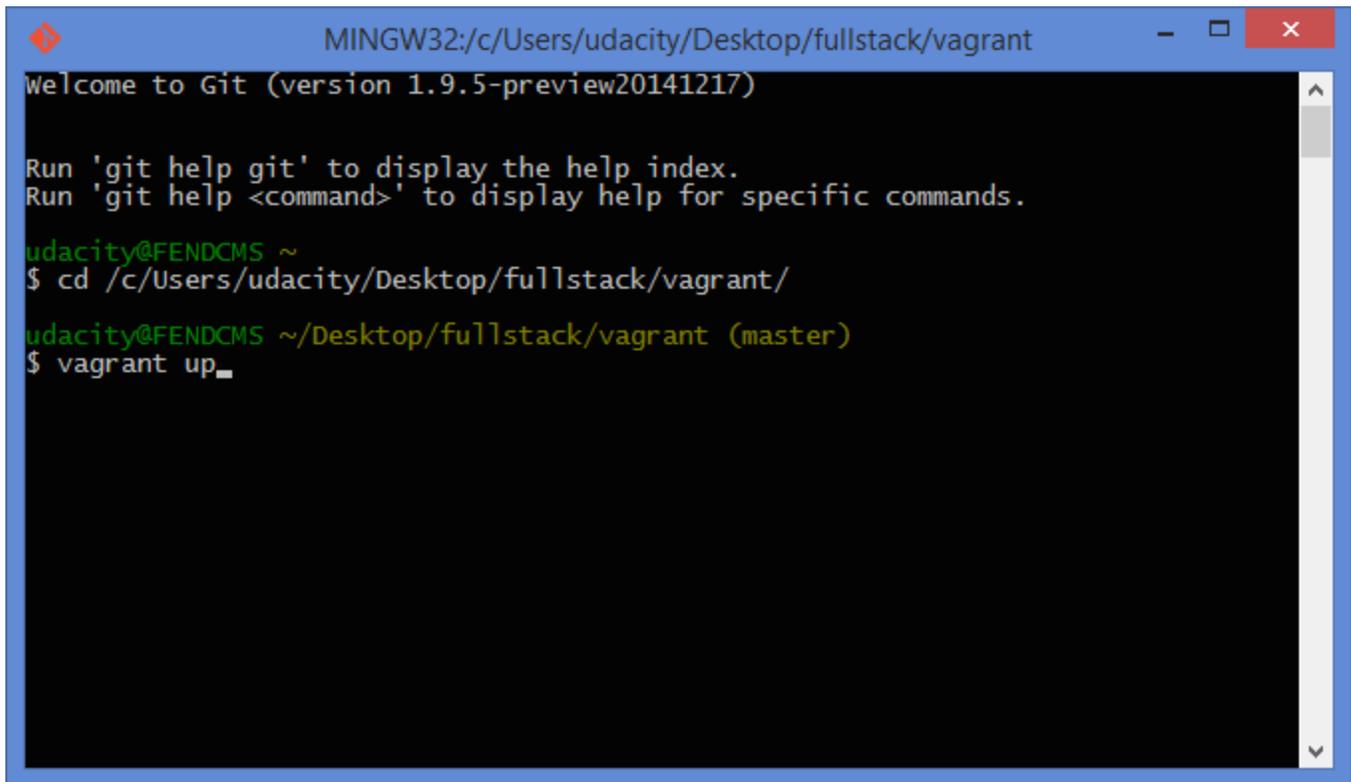
Using the terminal, change directory to `fullstack/vagrant` (`cd fullstack/vagrant`), then type `vagrant up` to launch your virtual machine.

OS X Example:



```
vagrant — bash — 80x24
Last login: Wed Feb  4 16:34:53 on ttys003
udacity ~ $ cd Desktop/fullstack/vagrant/
udacity (master *) vagrant $ vagrant up
```

Windows Example:

A screenshot of a Windows terminal window titled "MINGW32:/c/Users/udacity/Desktop/fullstack/vagrant". The terminal shows the following text: "Welcome to Git (version 1.9.5-preview20141217)", "Run 'git help git' to display the help index.", "Run 'git help <command>' to display help for specific commands.", "udacity@FENDCMS ~", "\$ cd /c/Users/udacity/Desktop/fullstack/vagrant/", "udacity@FENDCMS ~/Desktop/fullstack/vagrant (master)", "\$ vagrant up_".

```
MINGW32:/c/Users/udacity/Desktop/fullstack/vagrant
Welcome to Git (version 1.9.5-preview20141217)

Run 'git help git' to display the help index.
Run 'git help <command>' to display help for specific commands.

udacity@FENDCMS ~
$ cd /c/Users/udacity/Desktop/fullstack/vagrant/
udacity@FENDCMS ~/Desktop/fullstack/vagrant (master)
$ vagrant up_
```

Once it is up and running, type **vagrant ssh** to log into it. This will log your terminal in to the virtual machine, and you'll get a Linux shell prompt. When you want to log out, type **exit** at the shell prompt. To turn the virtual machine off (without deleting anything), type **vagrant halt**. If you do this, you'll need to run **vagrant up** again before you can log into it.

Files and commands we'll be using (Relational Databases)

Files installed for this class are located in the **/vagrant** directory inside the virtual machine. Everything here is automatically shared with the **vagrant** directory inside the **fullstack** directory on your computer. Any code files you save into that directory from your favorite text editor will be automatically available in the VM.

If you'd like to see what was installed in the VM, look in **/vagrant/pg_config.sh**.

In this class you will mostly be running your work in Python from the command line. In addition you'll use the **psql** program to interact with the PostgreSQL database.

To connect **psql** to the forum database for Lesson 3, type **psql forum** at the command line. To exit psql, type **\q** or Control-D (^D).