Chemeketa CS Dev Environment Setup Instructions

These instructions describe how to set up the official Chemeketa CS C++ development environment. They will make a virtual machine on your computer that runs Linux and has the tools needed to build and debug C++ code.

Contents

Requirements

1. Approximately 20 gigs hard drive space.
2. Fast internet or a lot of time (you will need to download ~5gigs of data). If you do not have fast internet at home, I recommend doing this from school or a public library.
3. Processor that supports 64bit operation and has hardware virtualization - standard on every machine. **Make sure it is active.** Sometimes hardware virtualization is disabled by a PC manufacturer.

   To check if it is active on your PC, you can follow the instructions here:
   https://stackoverflow.com/questions/49005791/how-to-check-if-intel-virtualization-is-enabled-without-going-to-bios-in-windows
   (It should always come enabled on macs)

   If it is NOT enabled, you will need to access the BIOS to enable it:
   https://helpdeskgeek.com/how-to/enable-virtualization-in-the-bios/
   If you can't figure out what key to press while your computer boots to access the BIOS, try googling something like "HP laptop access BIOS" (but replace HP with your computer brand).

Download and install:

1. VirtualBox: https://www.virtualbox.org/wiki/Downloads (Program for running virtual machines)
   Pick the operating system you are currently running on your computer (Windows, Mac etc…)

   On Windows, If you have Hyper-V enabled, you will be asked to disable it during install. You must do so. Hyper-V is like VirtualBox - a platform for running VMs - that comes build into Windows. You can't use both at the same time so Hyper-V needs to be turned off. You can use Hyper-V as the VM provider for Vagrant instead of VirtualBox - you may want to do that if you are actively using Hyper-V for other virtual machines. Doing so is not insanely hard, but is not covered in this document. (Use google and check here: https://blogs.technet.microsoft.com/virtualization/2017/07/06/vagrant-and-hyper-v-tips-and-tricks/)

2. Vagrant: https://www.vagrantup.com/downloads.html (Program for automating setup and maintenance of virtual machines)
   Again, pick the version for the operating system you are currently using.

Setup:

1. Make a directory where you want to store your work. You can call it anything you want, but the example assumes you call the folder **ChemeketaDevEnvironment**.

   This folder should be on your computer's filesystem - do not put it in a One Drive folder or iCloud folder.

2. Download the file below and save it to that directory.
   http://computerscience.chemeketa.edu/CSResources/Vagrant/Vagrantfile

   The file should simply be called **Vagrantfile** without any extension (no .txt or .dms). It is the instructions vagrant will use to setup your virtual machine. It is a plain text file, you can check it out if you like (just make sure not to by accident save it with an extension .txt).

3. Open up a command prompt window (PC) or terminal window (Mac). Navigate to the folder you created. Type:
   vagrant up
   The first time you use this it will take a while. It is going to download a Linux disk image, install it into the virtual machine and then download and configure some other software.

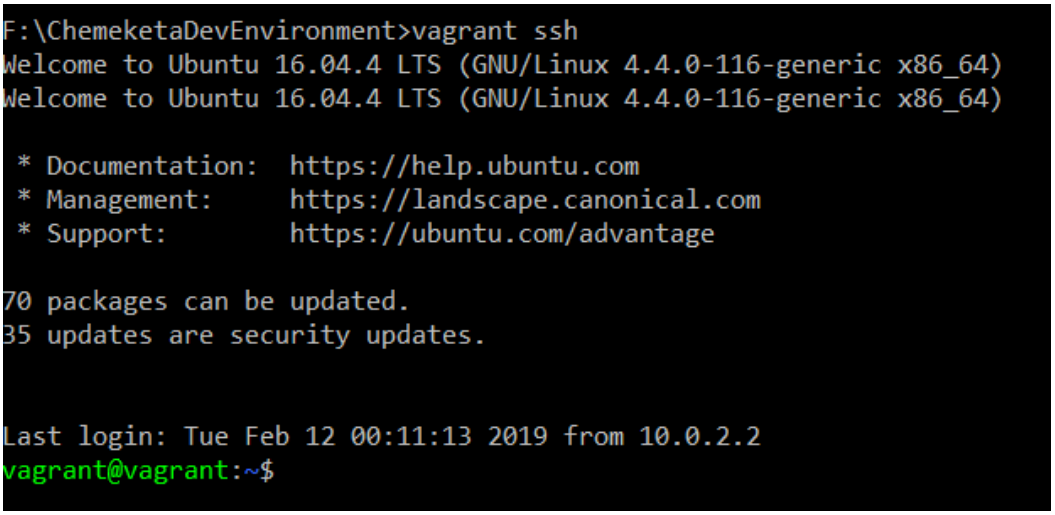
   If this causes an error or times out, the most likely cause is that you do not have Hardware Virtualization turned on in your BIOS. ***Go back and check Requirements step 3.***

Use:

1. Open up a terminal/command prompt window. Navigate to the folder you created. If you have not just finished setup, you may need to type this to bring the virtual machine up:
   vagrant up

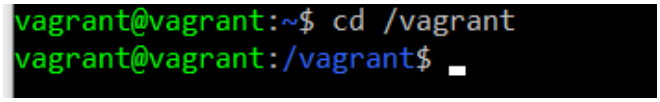
2. Now type:
   vagrant ssh
   This will connect to your virtual machine using the ssh network protocol.
   You should not be asked for a password, but if you are, use:
   vagrant
   For security you will not be shown anything as you type the password.

3. At this point your command prompt should change and tell you welcome to ubuntu:



   Initially you are in the directory **/home/vagrant** on the Linux system. This is your home directory and is represented by the shorthand **~**. You can return to the home directory at any point with:
   cd ~

4. Navigate to the directory /vagrant by typing:
   cd /vagrant
   The prompt should change to show that you are in the /vagrant directory:



   / is the "root" of the filesystem. It is the top level folder in Linux. So /vagrant is a folder called vagrant that is in the top level folder.

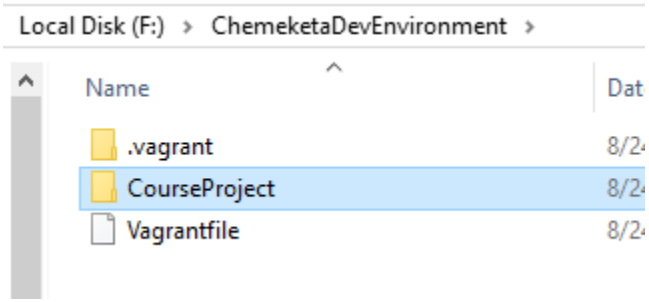
5. You can also check what directory you are in with the present working directory command. Type:
   pwd

6. Now list the files in that directory:
   ls -la
   Those are both lower-case L . The command is just **ls**, the **-la** gives a longer listing and shows all files (including hidden ones).
   You should see a list of the files in that directory:

```
vagrant@ubuntu-16:/vagrant$ ls -la
total 8
drwxr-xr-x  2 vagrant vagrant    0 Aug 24 18:33 .
drwxr-xr-x 25 root    root    4096 Aug 24 19:36 ..
drwxr-xr-x  2 vagrant vagrant    0 Aug 24 18:34 .vagrant
-rwxr-xr-x  1 vagrant vagrant 3818 Aug 24 19:19 Vagrantfile
vagrant@ubuntu-16:/vagrant$
```

   Note that it has a file called Vagrantfile and a directory called .vagrant. Those are the files on your host machine (real operating system) in the folder you set up (ChemeketaDevEnvironment).

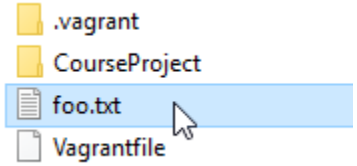
7. Add some more files from your host system. I added a QTCreator project folder called CourseProject. You can use any folder you like to test with.



   If I list the files in the directory /vagrant on the virtual machine, I will see it:

```
vagrant@ubuntu-16:/vagrant$ ls -la
total 12
drwxr-xr-x  2 vagrant vagrant 4096 Aug 24 20:46 .
drwxr-xr-x 25 root    root    4096 Aug 24 19:36 ..
drwxr-xr-x  2 vagrant vagrant    0 Aug 24 20:46 CourseProject
drwxr-xr-x  2 vagrant vagrant    0 Aug 24 18:34 .vagrant
-rwxr-xr-x  1 vagrant vagrant 3818 Aug 24 19:19 Vagrantfile
vagrant@ubuntu-16:/vagrant$
```

8. Anything you do to the files in the guest Linux operating system will show up in your host operating system. Type:
   touch foo.txt
   This should make a blank text file called foo.txt. Check in your host operating system and you should see it appear:



   You can of course see it in Linux by typing "**ls -la**".

9. Navigate to the folder you just copied. For me that would be
   cd CourseProject
   As CourseProject is the name of the folder. Then do "**ls -la**" to see the names of the files in it.

```
vagrant@ubuntu-16:/vagrant/CourseProject$ ls -la
total 38
drwxr-xr-x 2 vagrant vagrant 4096 Aug 24 20:46 .
drwxr-xr-x 2 vagrant vagrant 4096 Aug 24 20:51 ..
-rwxr-xr-x 1 vagrant vagrant 1011 Apr  4  2016 Course.cpp
-rwxr-xr-x 1 vagrant vagrant  585 Apr  4  2016 Course.h
-rwxr-xr-x 1 vagrant vagrant 2197 May 12  2016 CourseList.cpp
-rwxr-xr-x 1 vagrant vagrant  826 May 12  2016 CourseList.h
-rwxr-xr-x 1 vagrant vagrant  664 Apr  4  2016 CourseProject.pro
-rwxr-xr-x 1 vagrant vagrant  187 Apr  4  2016 Courses.txt
-rwxr-xr-x 1 vagrant vagrant 1371 Apr  4  2016 main.cpp
-rwxr-xr-x 1 vagrant vagrant 2870 May 12  2016 Student.cpp
-rwxr-xr-x 1 vagrant vagrant  843 May 12  2016 Student.h
vagrant@ubuntu-16:/vagrant/CourseProject$
```

10. Finally, do:
    cd ..
    This will take you up one folder level - out of your project folder (CourseProject) and back to the **/vagrant** directory.

11. Building, running and testing code are covered in a separate document.

Shutdown:

If you close the command prompt, your "network" connection to the virtual machine will close, but the machine will still be running. To reconnect, you can open a new command prompt, navigate to the folder you created (**ChemeketaDevEnvironment**) and then do:
vagrant ssh
To reconnect.

Normally when done working, you want to do:

1. In Linux, type:
   exit

2. To turn off the virtual machine so it is not consuming system resources (memory mostly), do:
   vagrant halt
   Once halted, you need to do **vagrant up** again to restart the machine before you can **vagrant ssh** to connect to it and use it.

3. If you are not sure if the machine is active or not, do:
   vagrant status
   If that shows a status of "running", the machine is not halted.

```
F:\ChemeketaDevEnvironment>vagrant status
Current machine states:

default                   running (virtualbox)

The VM is running. To stop this VM, you can run `vagrant halt` to
shut it down forcefully, or you can run `vagrant suspend` to simply
suspend the virtual machine. In either case, to restart it again,
simply run `vagrant up`.
```

4. If you want to wipe out your existing virtual machine and start with a fresh one, you can do:
   vagrant destroy
   This will not affect any "normal" files on your host operating system - everything in the **ChemeketaDevEnvironment** folder will still be there.