

Instructions for setting up you CS3319 Virtual Machine

Each database student is going to set up his/her own virtual machine. This machine will be running Linux (Version: CentOS 7). We have created the machines for each of you, but you will each need to set up a few things to be able to connect to your virtual machine and use it from your laptop. You will each be mailed your own .pem key file. This file is your "private key" to unlock your machine. With this file/key, you do NOT need a password to log onto your machine. You will be emailed this file. DO NOT LOSE THIS FILE. You cannot do anything without this .pem file, so make sure you have it handy.

Please note that we do NOT backup virtual machines. If you screw something up on your virtual machine, we CANNOT get it back, so please make backups of your files and do not change any of the system files unless you are told to in this document OR you are an experienced Linux user!

The first few steps differ slightly depending on if you are a Windows user, a Mac user or a Linux user:

For Windows Users:

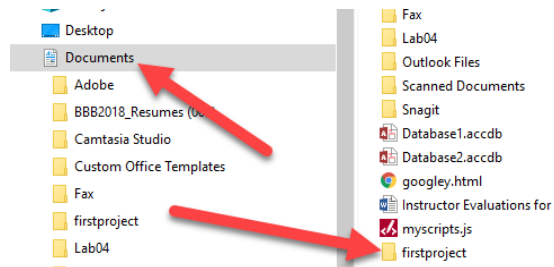
1. Install Git on your machine: <https://gitforwindows.org/> - just use the default options when using the Wizard. You are going to use Git to move our files from your local machine to your virtual machine AND as backup of your files.
2. Install PuTTY on your machine: <https://www.ssh.com/ssh/putty/download> - just use the default options

3. Configure your git as follows:

1. Run git.bash
2. In the command prompt window, type (remember to fill in YOUR name and email)::

```
git config --global user.name "Your Name"  
git config --global user.email "you@youremail.ca"  
git config --global core.editor "notepad.exe"  
git config --global color.ui true
```

3. Create the folder where you want to put your project. We are going to use the **documents/firstproject** folder for this example. So go to your documents folder and create a folder called **firstproject**.



4. Create your repository to hold all your work → Run git bash again and move (cd) to the following directory and do the following commands (git init creates a repository in this directory):

```
cd ~
cd Documents/firstproject
git init
```

```
lreid2@xkcd2 MINGW64 ~
$ cd ~

lreid2@xkcd2 MINGW64 ~
$ cd Documents/firstproject

lreid2@xkcd2 MINGW64 ~/Documents/firstproject
$ git init
Initialized empty Git repository in C:/Users/lreid2/Documents/firstproject/.git/

lreid2@xkcd2 MINGW64 ~/Documents/firstproject (master)
$ |
```

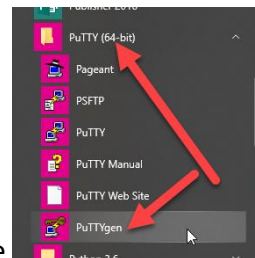
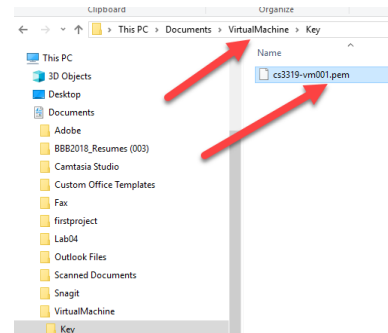
5. Open notepad
6. Save the file as **firstpage.html** to the directory you just created called **firstproject** Put the following html tags into firstpage.html using notepad:

```
<html>
<title>You had me at</title>
<body>
<h1>Hello World</h1>
</body>
</html>
```

7. We will come back to git later on in the workshop to push your local files up to github and then pull those files onto your virtual machine.

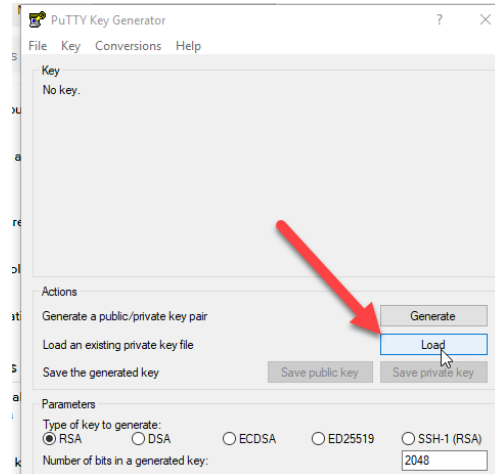
4. We will go back to PuTTY now to set up the Virtual Machine:

1. Save your .pem key given to you for this course to a folder of your choice, for example in your Documents folder, make a folder called VirtualMachine and a subfolder in that called Key so you will have: Documents/VirtualMachine/Key

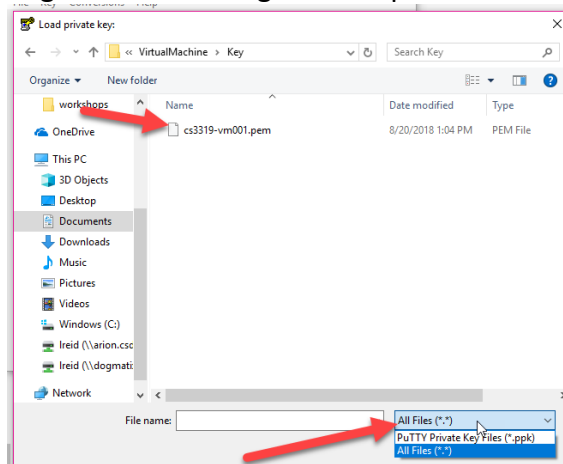


2. From your Windows menu, start up PuTTYgen.exe

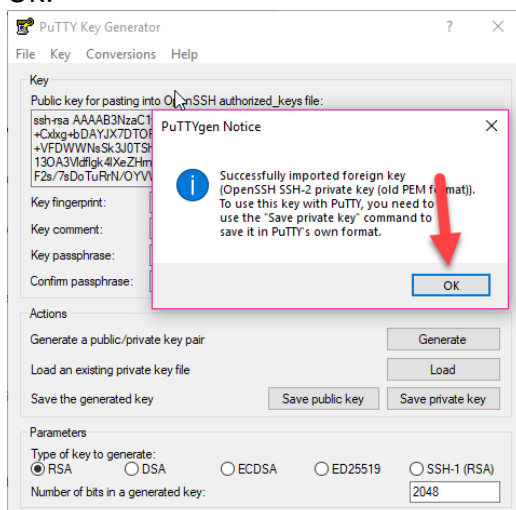
3. Select the **Load** button



4. **Navigate** to the folder you created in step 1 above and find your .pem file. NOTE: you might have to change the dropdown box to All Files (*.*) to see your .pem file.

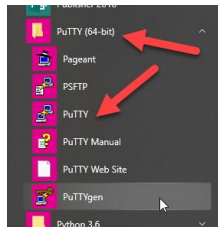


5. Select the .pem file and then select Open. When you get the PuTTYgen Notice, just hit OK:



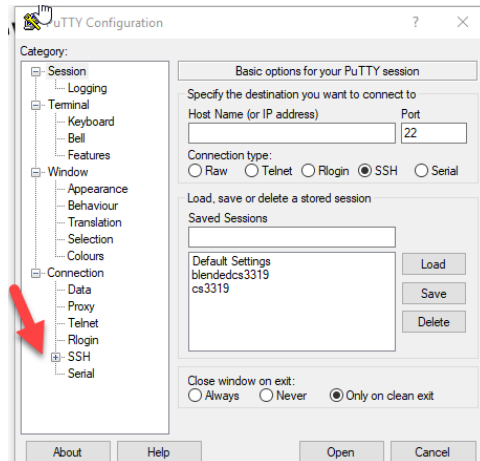
6. Hit the Save private key button and when asked "Are you sure you want to save this key without a passphrase to protect it?" just hit the Yes button.
7. Save the .ppk file in the same location as your .pem file. Give it the name of your virtual machine number, e.g. vm001.ppk

8. Close PuTTYgen

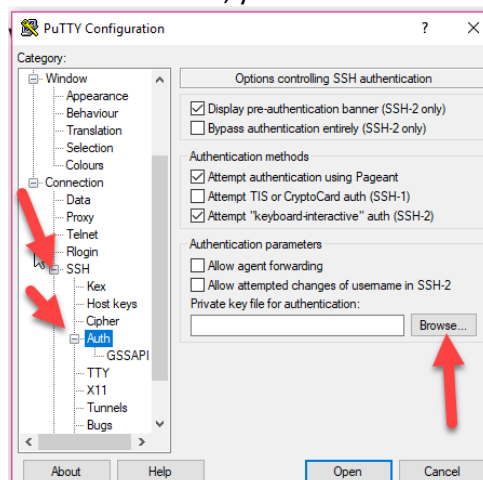


9. Start the program PuTTY now

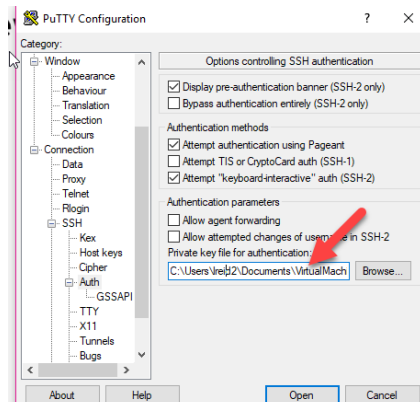
10. You will see a screen like this:



11. Click on the + next to SSH to expand it and then click on the + next to Auth to expand it and then click on Auth, you will see this:



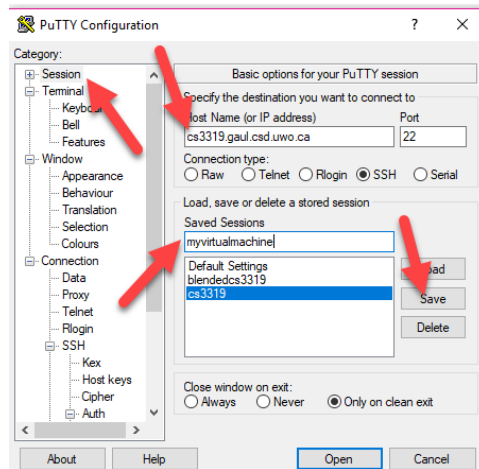
12. Click on the Browse button and find the location of you vm00?.ppk file that you created above and click on the file and then click on the Open button. You should see something like this:



13. Now go back up to the top and click on the Session category on the left and fill it in as follows:

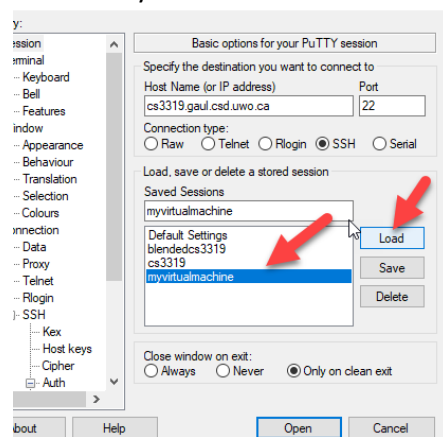
1. Put **cs3319.gaul.csd.uwo.ca** for the Host Name (or IP address)
2. Put **myvirtualmachine** for the Saved Sessions

It should look like this:



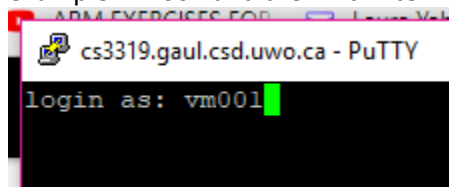
14. Hit the Save button so that you can use this session over and over again.

15. Click on myvirtualmachine from the list of Saved Sessions and hit the Load button:

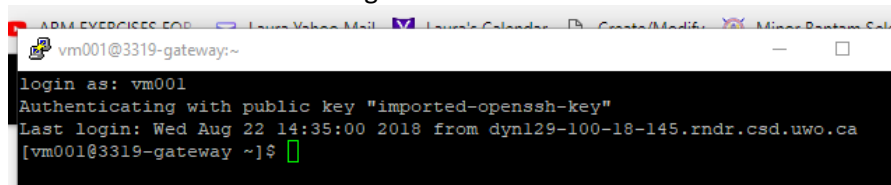


16. Hit the Open button to open a terminal to this virtual machine

17. At the login as: prompt, type the number of the virtual machine you have been given, example: vm001 and then hit Enter



18. You should now see something like this:

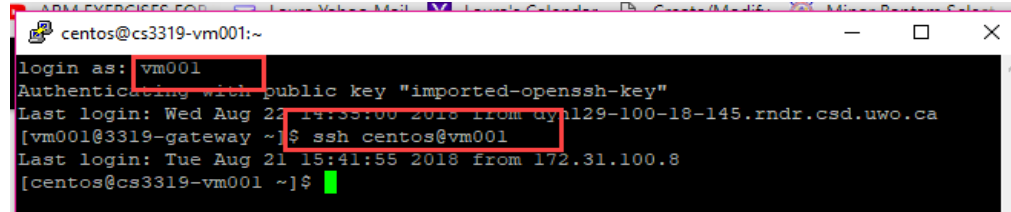


19. NOTE: this is NOT yet YOUR virtual machine, this is the machine that is the GATEWAY to yours AND EVERYONE ELSE'S virtual machine. Never do any work on this machine or save anything here. All you should ever have to do at this point is log onto your virtual machine as follows, at the prompt type:

```
ssh centos@vm???
```

(NOTE: vm??? Should be the virtual machine number assigned to you)

NOTE: vm??? Should be the machine number assigned to you, for example:



```
centos@cs3319-vm001:~  
login as: vm001  
Authenticating with public key "imported-openssh-key"  
Last login: Wed Aug 22 14:33:00 2018 from dyl129-100-18-145.rndr.csd.uwo.ca  
[vm001@3319-gateway ~]$ ssh centos@vm001  
Last login: Tue Aug 21 15:41:55 2018 from 172.31.100.8  
[centos@cs3319-vm001 ~]$
```

20. You are now logged into YOUR PERSONAL VIRTUAL MACHINE for this databases courses.
21. Skip down to the section that says "Remaining Instructions are for Everyone"

For MAC/LINUX users:

1. Install Git on your machine (sorry, I don't have a Mac so I am not sure how to install it but should be the same as other software). Then follow all the parts of Step 3 (highlighted in yellow) in the Windows instructions above (but don't go past step4) to set up your Git stuff.
2. Open a terminal window and enter the following commands to create your SSH directory and lock it down:

```
mkdir -p ~/.ssh  
chmod 700 ~/.ssh
```

3. Copy the .pem file that was emailed to you by the computer science department into your .ssh directory and lock it down with 600 permissions:

```
cp ~/Downloads/jshantz4.pem ~/.ssh  
chmod 600 ~/.ssh/jshantz4.pem
```

4. Connect to your instance via SSH, substituting the appropriate filename and hostname below (this should be the hostname you created above) DO NOT CUT AND PASTE THIS COMMAND (it screws up the hyphen after the ssh -i if you cut and paste, type it in manually)

```
ssh -i ~/.ssh/jshantz4.pem vm???@cs3319.gaul.csd.uwo.ca
```

Make sure that vm??? Is the number assigned to you for your virtual machine and that jshantz4.pem is the name you gave the .pem file emailed to you

This command says:

- a. SSH into a server
- b. Using the identity file (private key) ~/.ssh/jshantz4.pem
- c. The username should be vm???
- d. And the hostname of the server is cs3319.gaul.uwo.ca

NOTE: this just brings you to a middle machine, NEVER EVER DO ANYTHING ON THIS MACHINE EXCEPT FOR THE ssh command below, you are not yet on your virtual machine!

5. Now type:

```
ssh centos@vm???
```

where vm??? is the number for the virtual machine assigned to you. This will put you onto your virtual machine; your username will be **centos**.

6. Type **exit** to disconnect from the server.
7. This method works fine, but it's a bit of a pain to type in such a long command to connect to the server. Fortunately, we can do better using the SSH configuration file.
 - a. Edit the file **~/.ssh/config** NOTE: This file will likely not already exist on your system. Make sure this file is called **config** and is NOT called **config.txt** or it won't work!
 - b. Enter the following in the file, substituting the correct identify filename and hostname:
Host dbvm
Hostname cs3319.gaul.csd.uwo.ca
User vm???
IdentityFile ~/.ssh/jshantz4.pem
 - c. Save the file and closer your editor. Then do:
chmod 600 config
 - d. You can now type the following to connect to your instance:
ssh dbvm
ssh centos@vm???

The Remaining Instructions are FOR EVERYONE:

5. We will now go through the steps to install all the software you will need for the remainder of this course:
 1. Installing a few updates onto your Linux Virtual Machine:
 1. Type the following at the prompt (just say Yes to any prompts that occur):

```
sudo yum update
sudo yum install wget
sudo yum install nano
sudo yum install emacs
```

2. Installing MySQL onto your Linux Virtual Machine:
 1. Type the following at the prompt - just say Yes to any prompts that occur: *(NOTE: If you have issues with the commands in the box below, the latest instructions for installing MySQL on centos (linux operating system) can be found here:*
<https://www.linode.com/docs/databases/mysql/how-to-install-mysql-on-centos-7/>
*ALSO NOTE: command is a lowercase L not a one: **..-el7-5** not **...e17-5***

```
wget http://repo.mysql.com/mysql-community-release-el7-5.noarch.rpm
sudo rpm -ivh mysql-community-release-el7-5.noarch.rpm
sudo yum update
sudo yum install mysql-server
sudo systemctl start mysqld
```

2. Next, type the command:
sudo mysql_secure_installation
3. Hit enter when prompted for a password as we have not set one yet.
4. You will now be prompted for a password, WE CANNOT RETRIEVE THE PASSWORD FOR YOU IF YOU FORGET IT, so pick something you wont forget, e.g. **cs3319**
5. For all the remaining prompts that you get, answer with Y (yes)

3. Installing git (to move files back and forth from our local machine to our virtual machine) onto your Linux Virtual Machine:

1. Type the following commands at the prompt (say Yes when prompted, remember to put in YOUR name and email and you might want emacs instead of nano)

```
sudo yum install git
git --version
git config --global user.name "Your Name"
git config --global user.email "you@youremail.ca"
git config --global core.editor "nano"
```

4. Installing Apache (a service that will allow us to host webpages):

```
sudo yum install httpd
sudo systemctl enable httpd.service
sudo systemctl restart httpd.service
cd /var
pwd
sudo chown -R centos www
```

5. Installing PHP (a programming language that allows us to talk to a MySQL database)

1. Type the following commands (answer Yes when prompted):

```
sudo yum install https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
sudo yum install http://rpms.remirepo.net/enterprise/remi-release-7.rpm
sudo yum install yum-utils
sudo yum-config-manager --enable remi-php70
sudo yum install php php-mcrypt php-cli php-gd php-curl php-mysql php-ldap php-zip php-fileinfo
php -v
sudo systemctl restart httpd.service
```

2. To test our php out, we will make testing directory and put some simple php and html into a webpage there and make sure it displays. Do the following commands (this will create a webpage file called junk.php):

```
cd /var/www/html
mkdir testing
cd testing
nano junk.php
```


3. The last command above runs the editor to add some code. We have installed Emacs and nano editors. My instructions will use nano but if you would rather use emacs, that is fine. While inside of nano, type the following:

```
<!DOCTYPE html>
<html><head>
<meta charset="utf-8">
<title>My first webpage</title>
</head>
<body>
<?php
include 'bigmistake.php';
?>
<h1>MyName's Virtual Machine WORKS!</h1>
<h2>Hurray</h2>
</body>
</html>
```

4. Hit CTRL o to output your file, hit ENTER and then hit CTRL x to exit the editor.
5. Open a browser and go to the following location:
<http://cs3319.gaul.csd.uwo.ca/vm???.testing>
where ??? is the number of your virtual machine
6. You should see your file you just created there, click on it to open your webpage.
7. There is one intentional mistake inside of *junk.php*. The php code says to include another php file called bigmistake.php BUT we haven't created that file so it is missing. I intentionally put that in because I want you to see that you are NOT receiving any errors or warnings in the browser about mistakes BUT we want to be able to see our coding mistakes! In order to be able to view our coding errors, we need to modify the configuration file for php. It is called php.ini BE VERY CAREFUL WHEN YOU MODIFY THIS FILE, you do NOT want to screw it up! (We will make a backup of it just to be on the safeside!)

6. Modifying the php.ini file using nano so that our errors are displayed in the browser.

1. Type the following commands:

```
cd /etc
sudo cp php.ini phpbackup.ini
sudo nano php.ini
```

2. Once inside the nano editor for php.ini, press ALT / to get to the very end of the php.ini file

3. Add the following 3 lines to the bottom of the file but do NOT cut and paste them, rather type them EXACTLY AS SHOWN BELOW because the ~ character doesn't copy correctly:

```
error_reporting = E_ALL & ~E_DEPRECATED & ~E_NOTICE
display_errors = On
display_startup_errors = On
```

4. Hit CTRL o and then ENTER to save the php.ini file and then hit CTRL x to exit from the editor.

5. Type the following command to restart Apache so that the error messages are displayed:

```
sudo systemctl restart httpd.service
```

6. Go back to your browser and refresh your page. You should now see the PHP error displayed on your screen!

7. You will NOT need to modify the php.ini file ever again UNLESS, for some reason, you need to reinstall PHP again, for example: if you completely screw up your virtual machine and we have to kill it and recreate it.

6. Creating a GitHub account. If you don't have a GitHub account, you will need to create one here: <https://github.com> NOTE: you maybe prompted for your github user id and password for some of these steps.

1. Once logged in, click on the small + in the top right corner of any screen in github and create a new repository called **firstcs3319**. Give it the description "Trying out github" and leave it public. Then hit the **Create repository** button

2. The next screen is important, do NOT close it. You should now see a screen titled 'Quick setup – if you've done this kind of thing before'. Make sure the that https button is pushed. Copy the link in the input right beneath this title, it should look like this:

<https://github.com/lreid/firstcs3319.git>

3. Go to your git bash window ON YOUR LOCAL MACHINES and cd to your folder that you created at the beginning of this workshop:

```
cd ~/Documents/firstproject
```

4. In the local git bash terminal window, type the following command:
git remote add origin [copied web address], for example:
and hit enter.

```
git remote add origin https://github.com/lreid/firstcs3319.git
```

5. Go into local git bash and type the following commands. These 2 commands will add the file you created in your firstproject folder (remember it was called firstpage.html) to your local repository AND commit it to the local repository.

```
git add .  
git commit -m "putting my first file in local repo"
```

6. Now we are going to PUSH that file up to github so that our virtual machine can pull it down. In your local git bash window type the following:

```
git push origin master
```

7. Now go back to your github window in the browser and do refresh and you should now see your firstpage.html file there.

7. Now we are going to pull files from github ONTO our virtual machine.

1. First we have make a copy of our local repository on our virtual machine. We can put it anywhere we want, we will put in the area where webpages show up (/var/www/html). Do the following commands:

```
cd /var/www/html/testing  
git clone https://github.com/lreid/firstcs3319.git  
ls  
cd firstcs3319
```

2. As you can see it created a new directory called firstcs3319 (the name of your repository that you created in github) and we have moved to it.
3. Now we are going to pull all the files that we just pushed up to github ONTO our virtual machine by typing the following command:

```
git pull origin master
```

4. Now, using nano or emacs, while in the virtual machine (make sure you are still in the firstcs3319 directory), create a file with a silly sentence in it and call that file ***silly.txt***

5. Then commit those files to the local repository on the virtual machine and then push them up to github as follows:

```
nano silly.txt (type in some silly text to a .txt file in firstcs3319 and save it)
ls
git add .
git commit -m "puush file from my vm to my local machine"
git push origin master
```

6. Now go back to your local machine, into git bash and PULL those files from your virtual machine ONTO your local machine as follows:

```
pwd (make sure you are in your firstproject folder that is your
git repository folder, if not, do the following cd command:)
cd ~/Documents/firstproject
git pull origin master
ls
```

7. Now your silly.txt file should be in your firstproject folder.
8. RECAP: Remember the process for moving files:
1. Create a directory on your local machine that will hold all your work
 2. Using git bash, cd to that directory
 3. Inside that directory in git bash, do **git init** to create a git repository on your local machine to hold the project you are working on.
 4. On your local machine, create your local files, such as .txt files, using the editor of your choice INSIDE the directory you created in step 1
 5. Using a browser, go to github.com and create a new repository (with the + button, top right) to hold your files and copy the https address you see when it is created
 6. Go back to your local machine to git bash and cd to your directory and type this command:
git remote add origin https://github.com/lreid/firstcs3319.git
 7. On your local machine, still in git bash, add and commit your files with **git add .** and **git commit -m "some message"**
 8. Then push the files to github using **git push origin master**
 9. Then go to your virtual machine and use the following command to create the repository that will hold your files:
git clone https://github.com/lreid/firstcs3319.git
 10. Then pull all the files into that directory on your virtual machine by cd ing to the directory and using the command:
git pull origin master
ls
the ls command is to make sure they got there.
 11. Then anytime you want to push files up to github use **git add .** **git commit -m "message"** and **git push origin master** and anytime you want to pull files use **git pull origin master**
 12. NOTE: for the steps in GREEN above, you should only have to do ONCE at the beginning when setting everything up, and never again.

9. This is one way to move files back and forth between your local machine and your virtual machine AND the great thing about this method is that it backs up all your files for you on github. You could also sftp your files back and forth instead of using github but keep in mind that you don't have backups that way and that we have NO way to recover your virtual machine if you lose your files.

WE DO NOT DO BACKUPS OF YOUR VIRTUAL MACHINES, IF YOU ACCIDENTLY DELETE YOUR FILES, WE CANNOT GET THEM BACK!

8. Finally we are going to make sure that mysql was installed correctly.
 1. Go to your virtual machine window:
 1. At the command prompt type:
`mysql -u root -p`
 2. Type in your password that you set up when prompted (it might be cs3319)
 3. Make sure you see the mysql prompts
 4. Hit \q to quit