Instructions for setting up you CS3319 Virtual Machine

Each database student is going to set up his/her own virtual machine. Every student's virtual machine will have a different number. This machine will be running Linux (Version: RockyLinux9). 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 given your own .pem key file. The name of the .pem file will tell you what your virtual machine number is. I have emailed you your .pem file near the beginning of this course as an email attachment, check your emails from me to find this 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 virtual machine. DO NOT LOSE THIS FILE. You cannot do anything without this .pem file, so make sure you have it handy.

To back up your virtual machine (VM), we will show you how to push your work to a Git repository at https://gitlab.sci.uwo.ca/users/sign_in

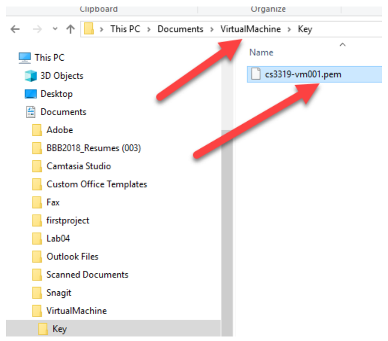
(Please note that we do NOT backup virtual machines. If you screw something up on your virtual machine, we CANNOT get it back, so please make backups of your files, push your files to GitLab regularly and do not change any of the system files unless you are told to in this document OR you are an experienced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

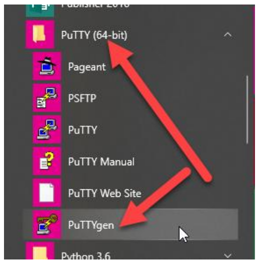
In the upcoming exercise, you will first set up your course virtual machine. Later in this same workshop, we will install and setup Git and practice how to push (upload) your files up to your Western's Git repository as well as pull (download/fetch) those files from the same Git repository onto your virtual machine.

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

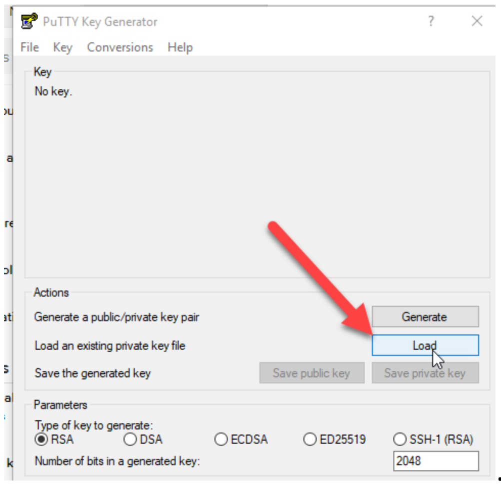
For Windows Users:

1. Install PuTTY on your machine: https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html - just use the default options. We will use PuTTY to set up the Virtual Machine.
2. Save your .pem key given to you for this course (sent as an email from me to you, check your emails from me, it will have an attachment of your .pem file) 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



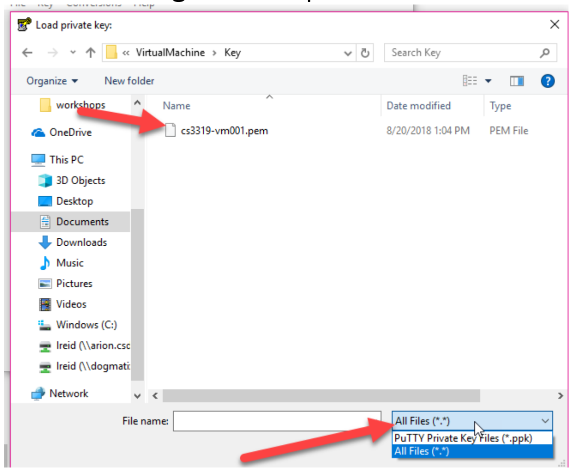


3. From your Windows menu, start PuTTYgen.exe
4. Select the ***Load*** button

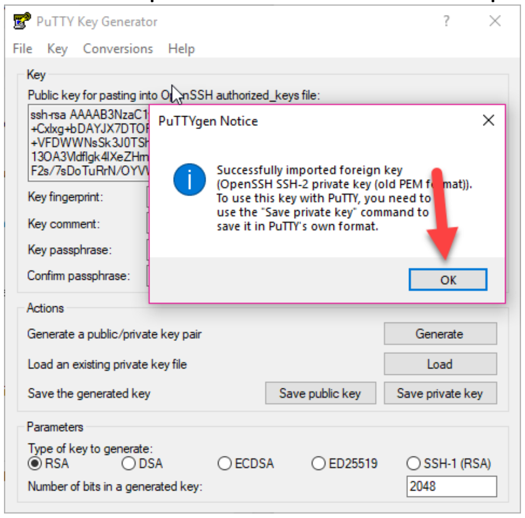


.

5. 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.



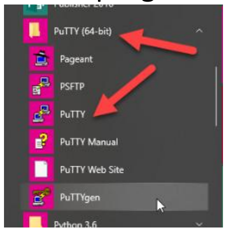
6. Select the .pem file and then select Open. When you get the PuTTYgen Notice, just hit OK:



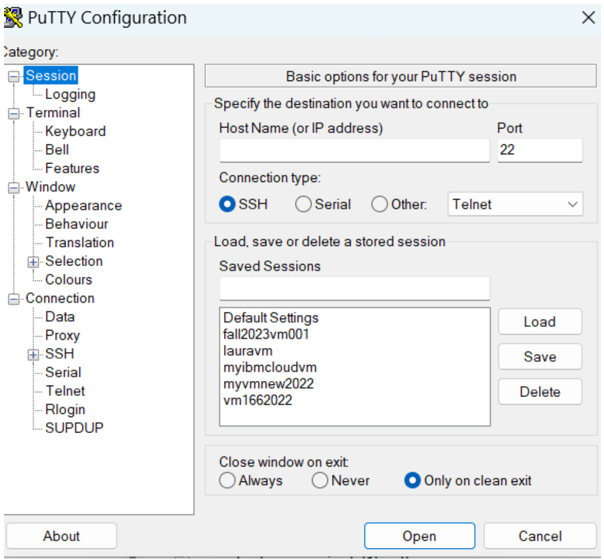
```
ssh-rsa AAAAB3NzaC1
+Cxlxg+bDAYJX7DTOF
+VFDWWNsSk3J0TSH
13OA3Vldflgk4lXeZHm
F2s/7sDoTuRrN/OYV
```

Successfully imported foreign key (OpenSSH SSH-2 private key (old PEM format)). To use this key with PuTTY, you need to use the "Save private key" command to save it in PuTTY's own format.

7. 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.
8. 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
9. Close PuTTYgen (You shouldn’t need this program ever again in this course)



10. Start the program PuTTY.exe now
11. You will see a screen like this:



PuTTY Configuration

Category:

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): Port: 22

Connection type: [✓ SSH] Serial Other: Telnet

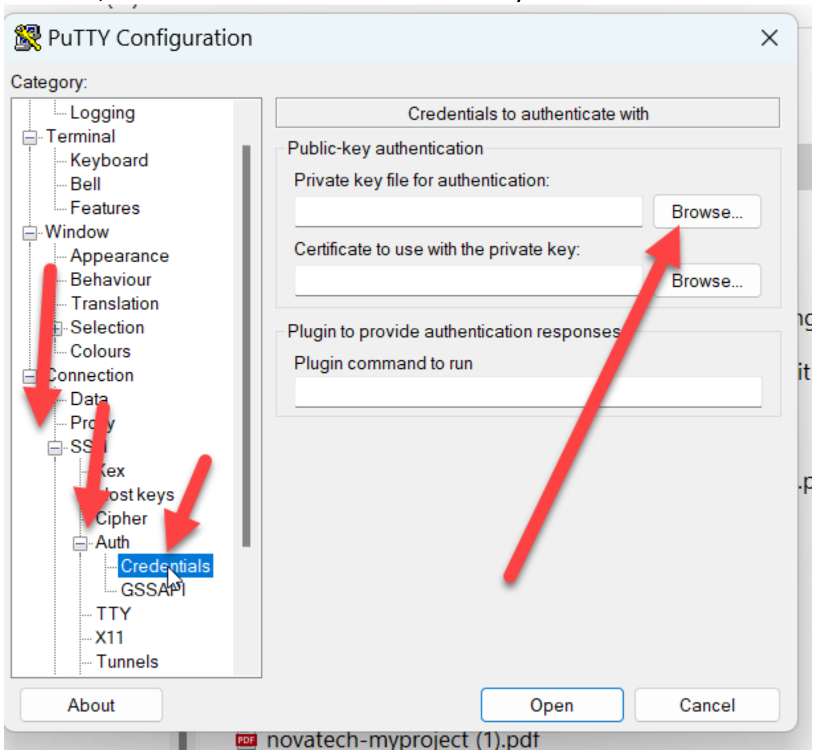
Load, save or delete a stored session

Saved Sessions

Default Settings
fall2023vm001
lauravm
myibmcloudvm
myvmnew2022
vm1662022

Close window on exit: Always Never [✓ Only on clean exit]

12. 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*, and then click on *Credentials* and you will see this:



PuTTY Configuration

Category:

Credentials to authenticate with

Public-key authentication

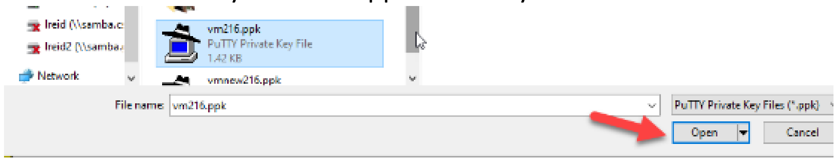
Private key file for authentication:

Certificate to use with the private key:

Plugin to provide authentication responses

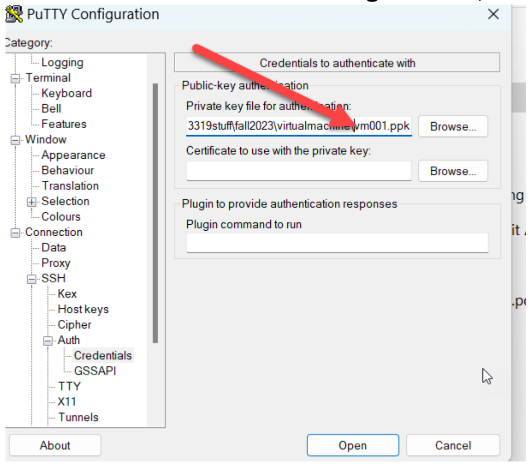
Plugin command to run

13. Click on the Browse button and find the location of you vm00?.ppk file that you created in the previous steps and click on the file and then click on the first Open button.



File name: vm216.ppk
PuTTY Private Key Files (*.ppk)

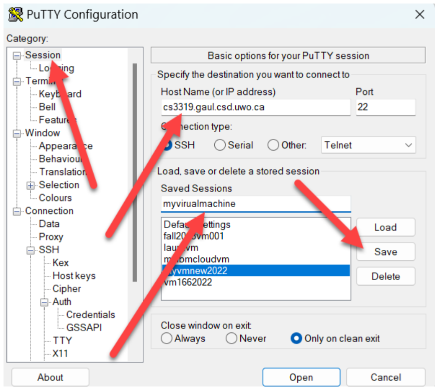
14. You should then see something like this, do NOT press the Open button on this screen yet:



Credentials to authenticate with
Public-key authentication
Private key file for authentication: 3319stuff\fall2023\virtualmachine\vm001.ppk
Certificate to use with the private key:
Plugin to provide authentication responses
Plugin command to run

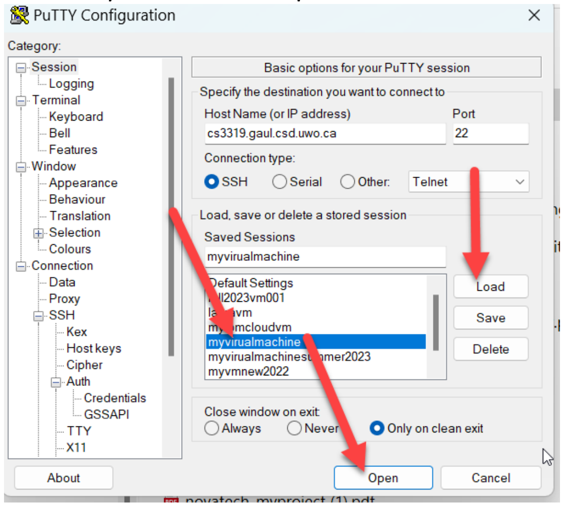
15. Now go back up to the top and click on the *Session* category on the left and fill it in as follows:
    1. Put **cs3319.gaul.csd.uwo.ca** for the Host Name (or IP address)
    2. Put **myvirtualmachine** for the Saved Sessions

    It should look like this:



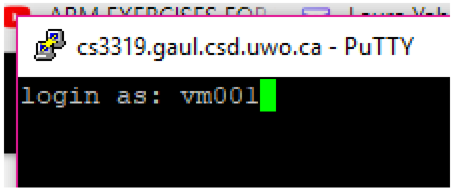
Basic options for your PuTTY session
Specify the destination you want to connect to
Host Name (or IP address): cs3319.gaul.csd.uwo.ca Port: 22
Connection type: [✓ SSH] Serial Other: Telnet
Load, save or delete a stored session
Saved Sessions: myvirualmachine
Default Settings
fall2023vm001
laurvm
mymbmcloudvm
myvmnew2022
vm1662022
Close window on exit: Always Never [✓ Only on clean exit]

    3. Hit the *Save* button so that you can use this session over and over again and not have to redo all the steps above each time you want to use your virtual machine.
    4. Click on *myvirtualmachine* from the list of Saved Sessions and hit the *Load* button and then hit the *Open* button to open a terminal to this virtual machine:



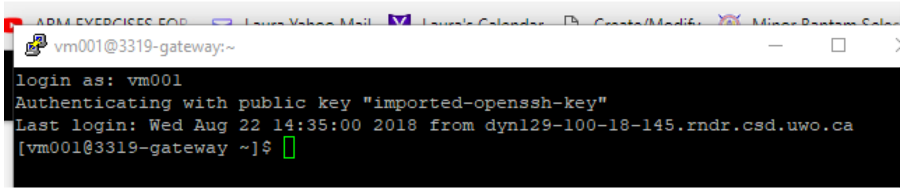
Basic options for your PuTTY session
Specify the destination you want to connect to
Host Name (or IP address): cs3319.gaul.csd.uwo.ca Port: 22
Connection type: [✓ SSH] Serial Other: Telnet
Load, save or delete a stored session
Saved Sessions: myvirualmachine
Default Settings
fall2023vm001
laurvm
mymbmcloudvm
myvirualmachine
myviruallmachinesummer2023
myvmnew2022
Close window on exit: Always Never [✓ Only on clean exit]

5. At the ***login as:*** prompt, type *vm???* Where ??? is the number of the virtual machine you have been given, In this example, the student has vm001 BUT YOU WILL HAVE A DIFFERENT NUMBER (the vm and the number at the end of the filename for YOUR pem key)and then hit Enter



```
login as: vm001
```

6. You should now see something like this (if it doesn't work, close putty and try restarting putty and redoing the 2 previous steps (load, open and login as vm???), it seemed to work for me the second try):

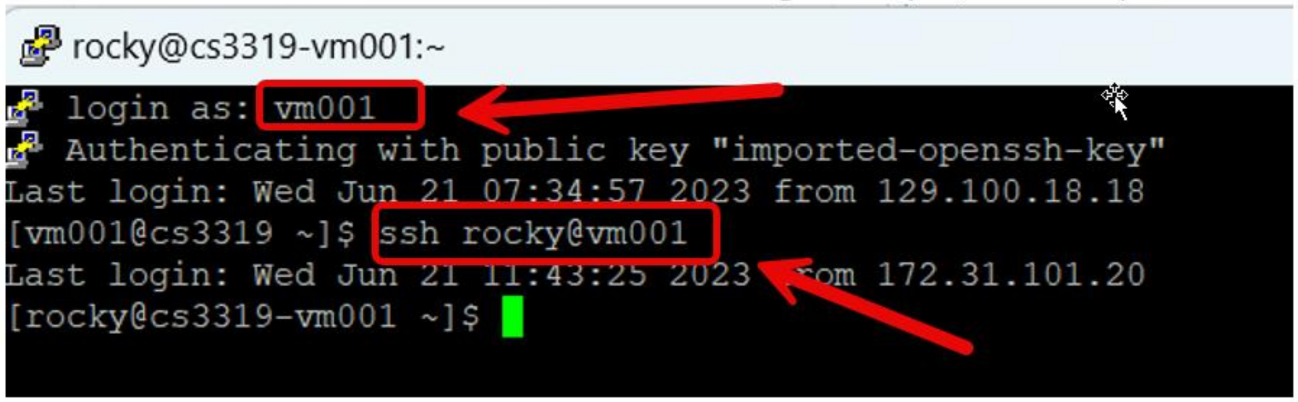


```
login as: vm001
Authenticating with public key "imported-openssh-key"
Last login: Wed Aug 22 14:35:00 2018 from dyn129-100-18-145.rndr.csd.uwo.ca
[vm001@3319-gateway ~]$
```

7. 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 gateway machine nor save anything here (i.e. ALWAYS do the next step of ssh rocky@vm???). All you should ever have to do at this point in order to log onto your virtual machine as follows, at the prompt type:

```
ssh rocky@vm???
```

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



```
login as: vm001
Authenticating with public key "imported-openssh-key"
Last login: Wed Jun 21 07:34:57 2023 from 129.100.18.18
[vm001@cs3319 ~]$ ssh rocky@vm001
Last login: Wed Jun 21 11:43:25 2023 from 172.31.101.20
[rocky@cs3319-vm001 ~]$
```

If you prompted to continue connecting, just type Yes

8. You are now logged into YOUR OWN PERSONAL VIRTUAL MACHINE for this databases course. NOTE: For the rest of this course, you should just need to do 3 things to get onto your virtual machine
   1. Run putty from your laptop, load *myvirtualmachine* from list of saved sessions and hit open
   2. Type in your vm??? number when prompted with **login as:**
   3. Then ALWAYS make sure that your next command is: **ssh rocky@vm???** at the next prompt. DO NOT FORGET THIS STEP (it takes you off the gateway machine that we all share and moves you on to YOUR virtual machine)
9. Skip down to the section that says "*Remaining Instructions are for BOTH Windows & Mac Users"*

For MAC/LINUX users:

1. Open a terminal window and enter the following commands to create your SSH directory and lock it down:

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

2. Copy the .pem file that I emailed to you near the beginning of the course, check your emails from me, it should be an attachment of one of my emails (part of the file name will be the number of the virtual machine assigned to you. We put ??? but yours will have a three-digit number, make sure you put the 3-digit number for ???) into your .ssh directory and lock it down with 600 permissions:

   ```
   cp ~/Downloads/cs3319-vm341.pem ~/.ssh
   chmod 600 ~/.ssh/cs3319-vm341.pem
   ```

3. 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, instead type it in manually)

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

   Make sure that vm??? Is the number assigned to you for your virtual machine and that *cs3319-???.pem* is the name you gave the .pem file you downloaded from your BrightSpace Group Locker.
   This command says:
   a. SSH into a server
   b. Using the identity file (private key) ~/.ssh/cs3319-vm???.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 MIDDLE MACHINE EXCEPT FOR THE following ssh command below because you are not yet on your virtual machine!

4. Now type:

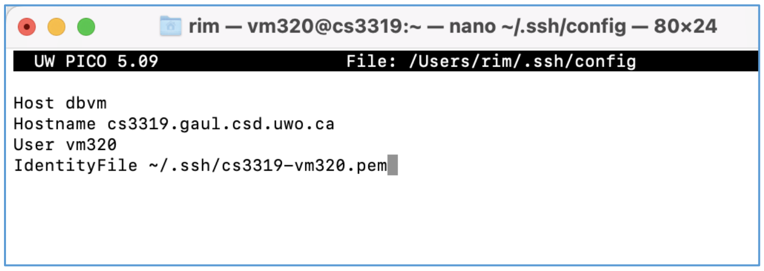
   ```
   ssh rocky@vm???
   ```

   where vm??? is the number for the virtual machine assigned to you. This will put you onto your virtual machine; your username will be ***rocky***.
5. At this point you would normally do your work on the machine but we have one last step to make it easier for you. So type **exit** twice (once to leave your virtual machine and once to leave the gateway machine) to disconnect from the server and go back to terminal (localuser)
6. 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!

      ```
      nano ~/.ssh/config
      ```
      If you don't have nano installed, you can use TextEdit but make sure it is Make Plain Text AND do make sure it doesn't put a file extension on the file name of config.

b. Enter the following in at the end of the file, substituting the correct identify filename and user virtual machine:

```
Host dbvm
Hostname cs3319.gaul.csd.uwo.ca
User vm???
IdentityFile ~/.ssh/cs3319-vm???.pem
```

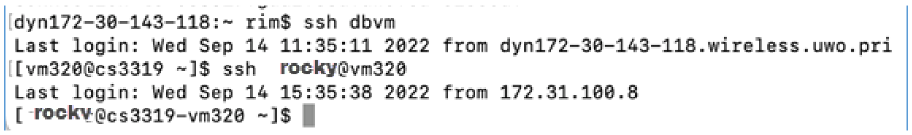


```
rim — vm320@cs3319:~ — nano ~/.ssh/config — 80×24
 UW PICO 5.09                     File: /Users/rim/.ssh/config

Host dbvm
Hostname cs3319.gaul.csd.uwo.ca
User vm320
IdentityFile ~/.ssh/cs3319-vm320.pem
```

c. Save the file (hit CTRL-O and then hit Enter and then hit CTRL-X to save the file and exit) and closer your editor. Then do:
chmod 600 ~/.ssh/config

d. You can now type the following to connect to your instance instead of the long command to type in step 4 and then step 5 above:

```
ssh dbvm
ssh rocky@vm???
```



```
[dyn172-30-143-118:~ rim$ ssh dbvm
Last login: Wed Sep 14 11:35:11 2022 from dyn172-30-143-118.wireless.uwo.pri
[vm320@cs3319 ~]$ ssh  rocky@vm320
Last login: Wed Sep 14 15:35:38 2022 from 172.31.100.8
[ rocky@cs3319-vm320 ~]$ 
```

The remaining instructions are for BOTH Windows & Mac Users:

We will now go through the steps to install all the software you will need for the remainder of this course (make sure you are still in the location [rocky@cs3319-vm3??? ~]$: (**i.e. don't forget that:** ***ssh rocky@vm???*** **step!)**

1. Installing a few updates and software onto your brand new Linux Virtual Machine: Type the four commands in the box to the right. NOTE: the first command may take a long time! Type *yes* or *y* at the prompts (just say *yes* or *y* to any prompts that occur). Again, please note that the: **sudo dnf update** command will take some time to complete!

```
sudo dnf update
sudo dnf install wget
sudo dnf install nano
sudo dnf install emacs
```

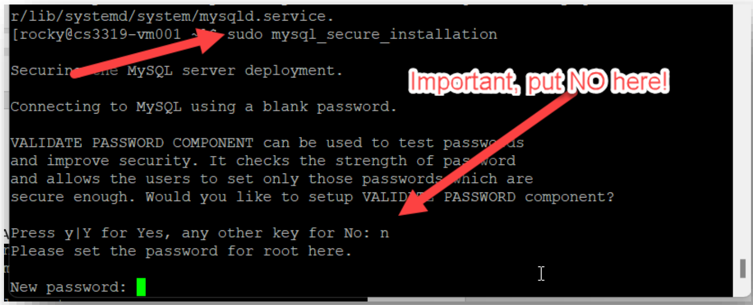
2. Installing MySQL onto your Linux Virtual Machine:
   - Type the following at the prompts - just say *yes* or *y* to any prompts that occur:

```
sudo dnf install mysql-server
sudo systemctl start mysqld.service
q
```

     NOTE: → If you have issues with the commands in the box below, the latest instructions for installing MySQL on rocky (linux operating system) can be found here: https://www.digitalocean.com/community/tutorials/how-to-install-mysql-on-rocky-linux-9
     *→The q command (for **q**uit) is just because the second command put me into the vi editor. q should allow you to exit the editor but if you don't get put into the editor, don't worry about typing the q.*

   - **STOP AND READ NEXT PART SLOWLY!**

     READ THIS RED PART FIRST BEFORE DOING THE 2 COMMANDS BELOW because this is where most students screw up: When you see this screen (look at screen to the right), **ANSWER NO** to the question: ***Would you like to setup VALIDATE PASSWORD component****?* so just put ***n*** then read step 3 below about what to put for the password. Type 2 commands below but remember to put N and remember to put password listed below!



```
r/lib/systemd/system/mysqld.service.
[rocky@cs3319-vm001 ~]$ sudo mysql_secure_installation

Securing the MySQL server deployment.

Connecting to MySQL using a blank password.

VALIDATE PASSWORD COMPONENT can be used to test passwords
and improve security. It checks the strength of password
and allows the users to set only those passwords which are
secure enough. Would you like to setup VALIDATE PASSWORD component?

Press y|Y for Yes, any other key for No: n
Please set the password for root here.

New password:
```

```
sudo systemctl enable mysqld
sudo mysql_secure_installation
```

NOTE: only read this box if you screwed up on Step 2 above and hit Y instead of N, where I tell you to hit N above (the red and yellow part above). If you DID SCREW UP THAT STEP, do the following commands BUT you only should do these next 5 commands if you didn't read the big STOP part in Step 2 above!

```
sudo systemctl stop mysqld
sudo /usr/bin/rm -R /var/lib/mysql/*
sudo systemctl start mysqld
sudo systemctl enable mysqld
sudo mysql_secure_installation
```

And make sure you hit NO when it prompts for: Would you like to setup Validate password component, then continue with step 3 at the bottom of page 8.

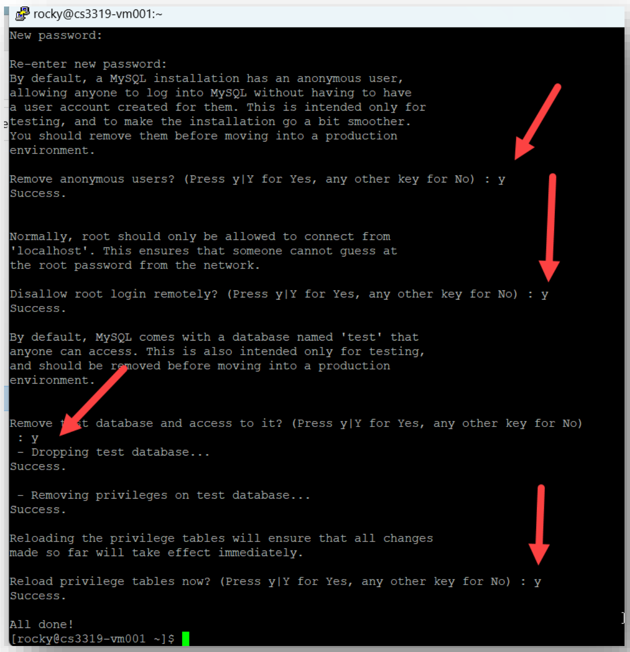
   - You will now be prompted if you want to set a new password. IMPORTANT: WE CANNOT RETRIEVE THE PASSWORD FOR YOU SO WE ARE ALL GOING TO USE THE EXACT SAME PASSWORD (don't worry, it doesn't matter since other students don't have access to your virtual machine, no one can get to your account unless they have your .pem file). Set the password to:

     cs3319

     IMPORTANT: if you do not use *cs3319* as your password (but I highly recommend you do this for mysql while taking this course) do NOT pick a password for mysql that contains a semicolon ; because it will screw up your assignment 2 later on. Also, if you do not make the password cs3319, later on the instructions in the flipped classroom and assignment 3 might be confusing. Thus PLEASE pick *cs3319* as your password here!

     You will be prompted to Re-enter new password, type ***cs3319*** again.

- For all the remaining prompts that you get, answer with Y (yes)



```
New password:

Re-enter new password:
By default, a MySQL installation has an anonymous user,
allowing anyone to log into MySQL without having to have
a user account created for them. This is intended only for
testing, and to make the installation go a bit smoother.
You should remove them before moving into a production
environment.

Remove anonymous users? (Press y|Y for Yes, any other key for No) : y
Success.


Normally, root should only be allowed to connect from
'localhost'. This ensures that someone cannot guess at
the root password from the network.

Disallow root login remotely? (Press y|Y for Yes, any other key for No) : y
Success.

By default, MySQL comes with a database named 'test' that
anyone can access. This is also intended only for testing,
and should be removed before moving into a production
environment.


Remove test database and access to it? (Press y|Y for Yes, any other key for No)
 : y
 - Dropping test database...
Success.

 - Removing privileges on test database...
Success.

Reloading the privilege tables will ensure that all changes
made so far will take effect immediately.

Reload privilege tables now? (Press y|Y for Yes, any other key for No) : y
Success.

All done!
[rocky@cs3319-vm001 ~]$
```

3. Next, we will setup **Apache** (a service that will allow us to host webpages) on your VM. For installing and setting the owner of folders for Apache, type the following commands, and again type *y* or *yes* for any prompts):

```
sudo dnf install httpd httpd-tools
sudo systemctl enable httpd.service
sudo systemctl restart httpd.service
cd /var
pwd
sudo chown -R rocky www
```

4. Next, we will setup **PHP** (a programming language that allows us to talk to a MySQL database) on your VM:
   - For installing PHP, type the following commands (type *y* or *yes* when prompted):

```
sudo dnf install php php-cli php-common
php -v
sudo dnf install php-mysqli
sudo systemctl restart httpd.service
```

   - 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 four commands in the box to the right (this will create a webpage file called junk.php in the directory `/var/www/html/testing`):

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

   - The last command above runs the text editor *nano* 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 (changing the red to your name):

```
<!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>
```

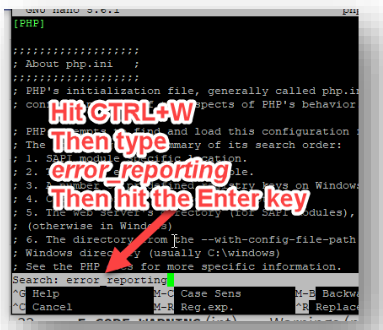
- Hit CTRL+O or Control+O to output (i.e. save) your file, hit ENTER and then hit CTRL+X or Control+X to exit the editor.
- Open a browser and go to the following location:
  http://cs3319.gaul.csd.uwo.ca/vm???/testing/
  where ??? is the number of your virtual machine. MAKE SURE YOU REMEMBER THE / at the very end of the URL, it didn't work for me without the /
- You should see your file you just created there, click on it to open your webpage.
- 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 mistake 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. The configuration file for php 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!)

  a. Go back to your virtual machine and type the 3 commands in the box to create a backup of the file(you will first create a backup called *phpbackup.ini* incase something wrong occurs with the modified *php.ini)*:

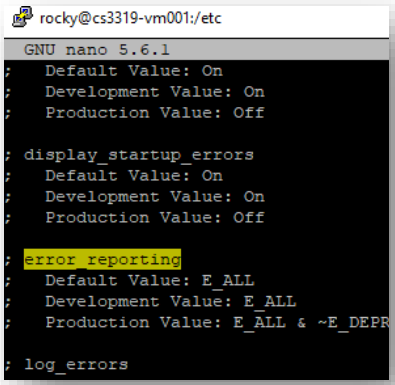
```
cd /etc
sudo cp php.ini phpbackup.ini
sudo nano php.ini
```

  b. Once inside the nano editor for *php.ini*, hold down the CTRL+W keys (W stands for ***W****here*, it is similar to ctrl+F on a windows machine to Find something) and type ***error_reporting*** and hit the Enter key.



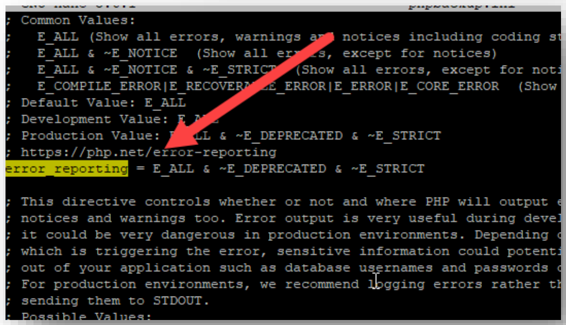
```
  GNU nano 5.6.1                                              php
[PHP]

;;;;;;;;;;;;;;;;;;;
; About php.ini   ;
;;;;;;;;;;;;;;;;;;;
; PHP's initialization file, generally called php.i
; con                      spects of PHP's behavior

; PHP                 find and load this configuration
; The                  mmary of its search order:
; 1. SAPI module specific location.
; 2.                        ble.
; 3. A number of predefined registry keys on Window
; 4.
; 5. The web server's directory (for SAPI modules),
; (otherwise in Windows)
; 6. The directory from the --with-config-file-path
; Windows directory (usually C:\windows)
; See the PHP      for more specific information.
Search: error_reporting
^G Help              M-C Case Sens        M-B Backw
^C Cancel            M-R Reg.exp.         ^R Replac
```

You will find the first occurrence of it, it will look like this →



```
rocky@cs3319-vm001:/etc
  GNU nano 5.6.1
;    Default Value: On
;    Development Value: On
;    Production Value: Off

; display_startup_errors
;    Default Value: On
;    Development Value: On
;    Production Value: Off

; error_reporting
;    Default Value: E_ALL
;    Development Value: E_ALL
;    Production Value: E_ALL & ~E_DEPR

; log_errors
```

then hit CTRL+W again and hit Enter to find the SECOND occurrence of it, then it should look like this (you want the line that does NOT start with a **;**) →

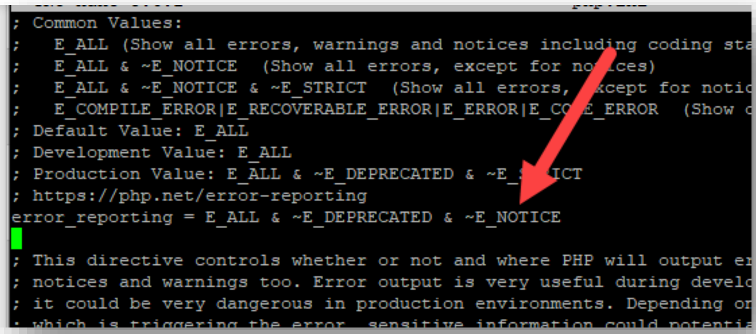


```
; Common Values:
;   E_ALL (Show all errors, warnings and notices including coding st
;   E_ALL & ~E_NOTICE  (Show all errors, except for notices)
;   E_ALL & ~E_NOTICE & ~E_STRICT  (Show all errors, except for noti
;   E_COMPILE_ERROR|E_RECOVERABLE_ERROR|E_ERROR|E_CORE_ERROR  (Show
; Default Value: E_ALL
; Development Value: E_ALL
; Production Value: E_ALL & ~E_DEPRECATED & ~E_STRICT
; https://php.net/error-reporting
error_reporting = E_ALL & ~E_DEPRECATED & ~E_STRICT

; This directive controls whether or not and where PHP will output e
; notices and warnings too. Error output is very useful during devel
; it could be very dangerous in production environments. Depending o
; which is triggering the error, sensitive information could potenti
; out of your application such as database usernames and passwords o
; For production environments, we recommend logging errors rather th
; sending them to STDOUT.
; Possible Values:
```

This is the line you want to CAREFULLY edit. Use your right arrow key to get to the end of the line and change the ~E_STRICT at the end of the line to be:

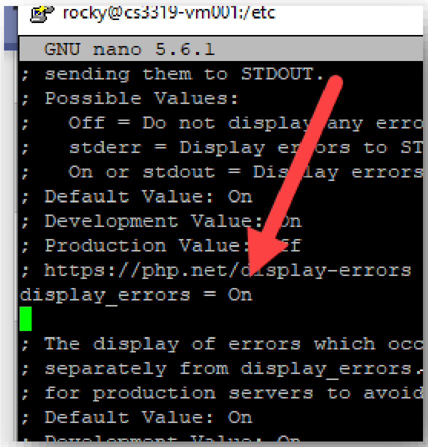
~E_NOTICE

So that is now looks like this →



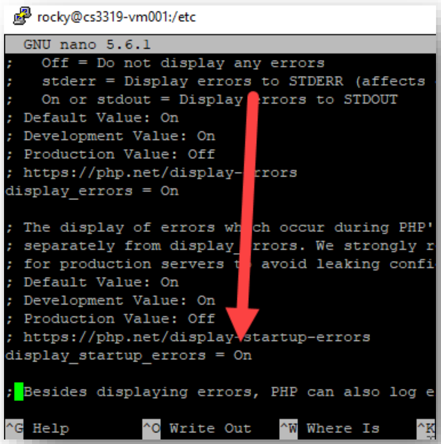
```
; Common Values:
;   E_ALL (Show all errors, warnings and notices including coding sta
;   E_ALL & ~E_NOTICE  (Show all errors, except for notices)
;   E_ALL & ~E_NOTICE & ~E_STRICT  (Show all errors, except for notic
;   E_COMPILE_ERROR|E_RECOVERABLE_ERROR|E_ERROR|E_CORE_ERROR  (Show c
; Default Value: E_ALL
; Development Value: E_ALL
; Production Value: E_ALL & ~E_DEPRECATED & ~E_STRICT
; https://php.net/error-reporting
error_reporting = E_ALL & ~E_DEPRECATED & ~E_NOTICE

; This directive controls whether or not and where PHP will output er
; notices and warnings too. Error output is very useful during develo
; it could be very dangerous in production environments. Depending or
; which is triggering the error, sensitive information could potentia
```

c. Now hit CTRL+W again and search for: ***display_errors***
Make sure that you are editing the line that does NOT start with ; (you might have to do CTRL+W twice)
The line you are looking for is this:
display_errors = Off
and change the word **Off** to
On
So it should look like this →



```
rocky@cs3319-vm001:/etc
  GNU nano 5.6.1
; sending them to STDOUT.
; Possible Values:
;   Off = Do not display any erro
;   stderr = Display errors to ST
;   On or stdout = Display errors
; Default Value: On
; Development Value: On
; Production Value: Off
; https://php.net/display-errors
display_errors = On

; The display of errors which occ
; separately from display_errors.
; for production servers to avoid
; Default Value: On
; Development Value: On
```

d. Now hit CTRL+W again and search for:
   display_startup_errors
   Make sure that you are editing the line that does NOT start with ; (you might have to do CTRL+W twice)
   The line you are looking for is this:
   display_startup_errors = Off
   and change the word **Off** to
   On
   So it should look like this →



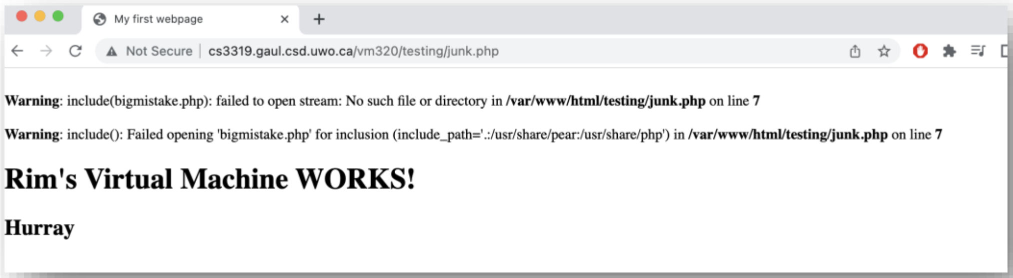
```
rocky@cs3319-vm001:/etc
  GNU nano 5.6.1
;   Off = Do not display any errors
;   stderr = Display errors to STDERR (affects
;   On or stdout = Display errors to STDOUT
; Default Value: On
; Development Value: On
; Production Value: Off
; https://php.net/display-errors
display_errors = On

; The display of errors which occur during PHP'
; separately from display_errors. We strongly r
; for production servers to avoid leaking confi
; Default Value: On
; Development Value: On
; Production Value: Off
; https://php.net/display-startup-errors
display_startup_errors = On

;Besides displaying errors, PHP can also log e
^G Help        ^O Write Out   ^W Where Is    ^K
```

e. Hit CTRL+O or Control+O and then ENTER to save the php.ini file and then hit CTRL+X or Control+X to exit from the editor.

f. Type the following commands to update everything and to restart Apache so that the error messages are displayed:

```
sudo systemctl restart php-fpm.service
```

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



My first webpage

Not Secure | cs3319.gaul.csd.uwo.ca/vm320/testing/junk.php

```
Warning: include(bigmistake.php): failed to open stream: No such file or directory in /var/www/html/testing/junk.php on line 7

Warning: include(): Failed opening 'bigmistake.php' for inclusion (include_path='.:/usr/share/pear:/usr/share/php') in /var/www/html/testing/junk.php on line 7
```

Rim's Virtual Machine WORKS!

Hurray

h. 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.

5. Now we will set our editor so that you can see line numbers when you are using the Nano editor.
   - First do this to create a file to set the properties of the Nano editor :

     ```
     nano ~/.nanorc
     ```

   - Then type this line into that file:

     ```
     set linenumbers
     ```

   - Then hit Ctrl-O to save the file and hit Ctrl-X to exit the file.

6. Next, you will install a zipping software so that you can zip files together for assignment 3:

```
sudo dnf install -y zip
```

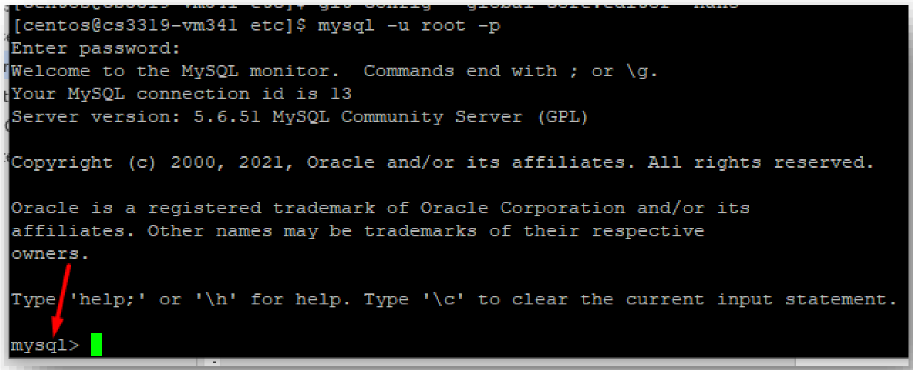
7. Next, you will install Git on your Virtual Machine. Type the following commands at the prompt (type *yes* or *y* when prompted, remember to put in YOUR name and email and you MIGHT want **emacs** below instead of **nano** - it depends on which editor you like better, I prefer to use nano). Hit Y when prompted.

```
sudo dnf install git
git --version
git config --global user.name "Your Name"
git config --global user.email "yourwesternuserid@uwo.ca"
git config --global core.editor "nano"
```

8. Finally, you will be testing *MySQL* to make sure that it was installed correctly. We will get into it, try the help command and then exit from it.
   Go to your virtual machine and, at the command prompt type:

```
cd ~
mysql -u root -p
```

9. Type in your password that you set up when prompted (it will likely be ***cs3319*** if you set it to the name I recommended above).
10. Make sure you see the mysql prompts like so:



```
[centos@cs3319-vm341 etc]$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 13
Server version: 5.6.51 MySQL Community Server (GPL)

Copyright (c) 2000, 2021, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

11. Type:
    help
    and make sure it shows you help for the mysql commands
12. Type:
    quit
    to exit/quit from mysql

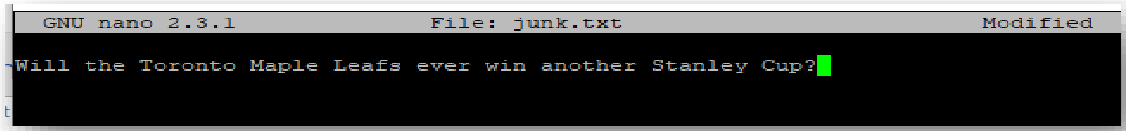
Using Git and Western's Git Repository to Regularly Backup Your Work.

In this section, you will practice how to push (upload) your files up to your Western's Git repository and pull (download/fetch) those files from your Git repository onto your created VM. Git is a code repository for backing up your work.

1. If you have logged out of your virtual machine, then log onto your virtual machine as you did before using PuTTY (if you, as a windows user, closed it) or via your terminal window (if you, as a mac or linux user, closed it), don't forget the:
   ssh rocky@vm???
   part and type these commands to create folders to put your assignments:

```
cd ~
mkdir assignments
cd assignments
mkdir assignment2
cd assignment2
nano junk.txt
```

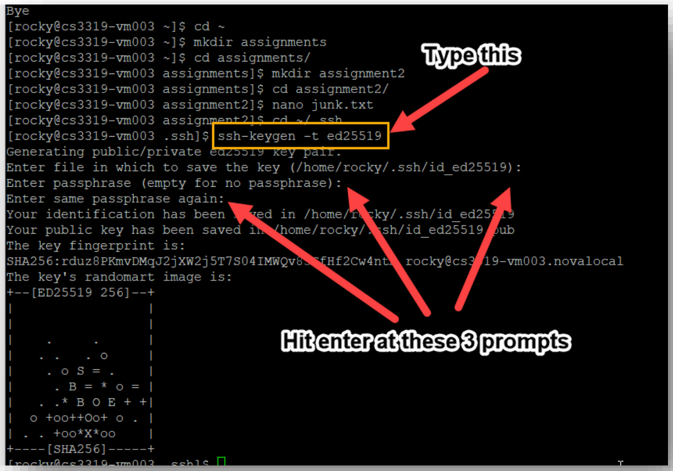
2. While in nano editor, type this text: "*Will the Toronto Maple Leafs ever win another Stanley Cup*?"



```
  GNU nano 2.3.1              File: junk.txt                         Modified

Will the Toronto Maple Leafs ever win another Stanley Cup?
```

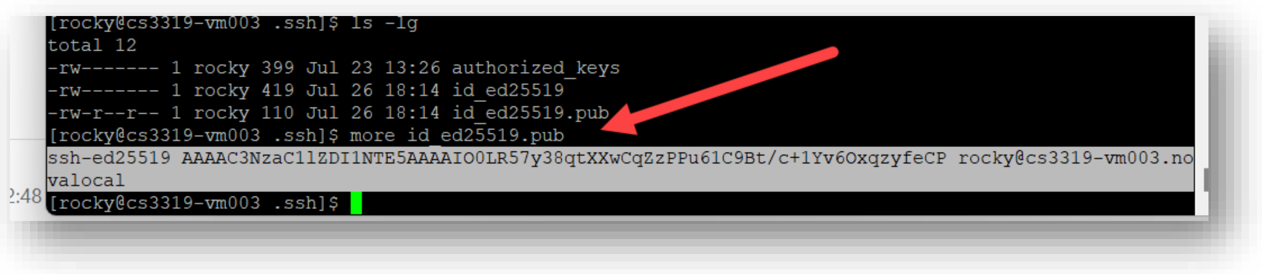
3. Hit CTRL+O or Control+O to save your file and CTRL+X or Control+X to exit nano
4. Now, we need to move to your .ssh folder. In your .ssh folder, you need to generate a private key and public key on your Virtual Machine. Then you will put your VM's public key into gitlab.sci.uwo.ca for your userid (It is your own little private repository just for this databases course). This will allow your VM to write to your repository. Type the following commands on your VM (Hit Enter three times when prompted about which file to save the key and about the passphrase):

```
cd ~/.ssh

ssh-keygen -t ed25519
ls -lg
more id_ed25519.pub
```



```
Bye
[rocky@cs3319-vm003 ~]$ cd ~
[rocky@cs3319-vm003 ~]$ mkdir assignments
[rocky@cs3319-vm003 ~]$ cd assignments/
[rocky@cs3319-vm003 assignments]$ mkdir assignment2
[rocky@cs3319-vm003 assignments]$ cd assignment2/
[rocky@cs3319-vm003 assignment2]$ nano junk.txt
[rocky@cs3319-vm003 assignment2]$ cd ~/.ssh
[rocky@cs3319-vm003 .ssh]$ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/rocky/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/rocky/.ssh/id_ed25519
Your public key has been saved in /home/rocky/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:rduz8PKmvDMqJ2jXW2j5T7S04IMWQv8[illegible]fHf2Cw4nt[illegible] rocky@cs3319-vm003.novalocal
The key's randomart image is:
+--[ED25519 256]--+
|                 |
|                 |
|     .     .     |
|    . .    . o   |
|     . o S = .   |
|      . B = * o =|
|    . .* B O E + +|
|   o +oo++Oo+ o .|
|  . . +oo*X*oo   |
+----[SHA256]-----+
[rocky@cs3319-vm003 .ssh]$
```

5. The *more* command will display the contents of the file you just created to the screen (the fille called id_ed25529.pub) Copy everything displayed into the clipboard (just highlight the text with your mouse, that is all you need to do to put it in the clipboard). Make sure you don't start your highlighting to soon or too late, it should look like this:



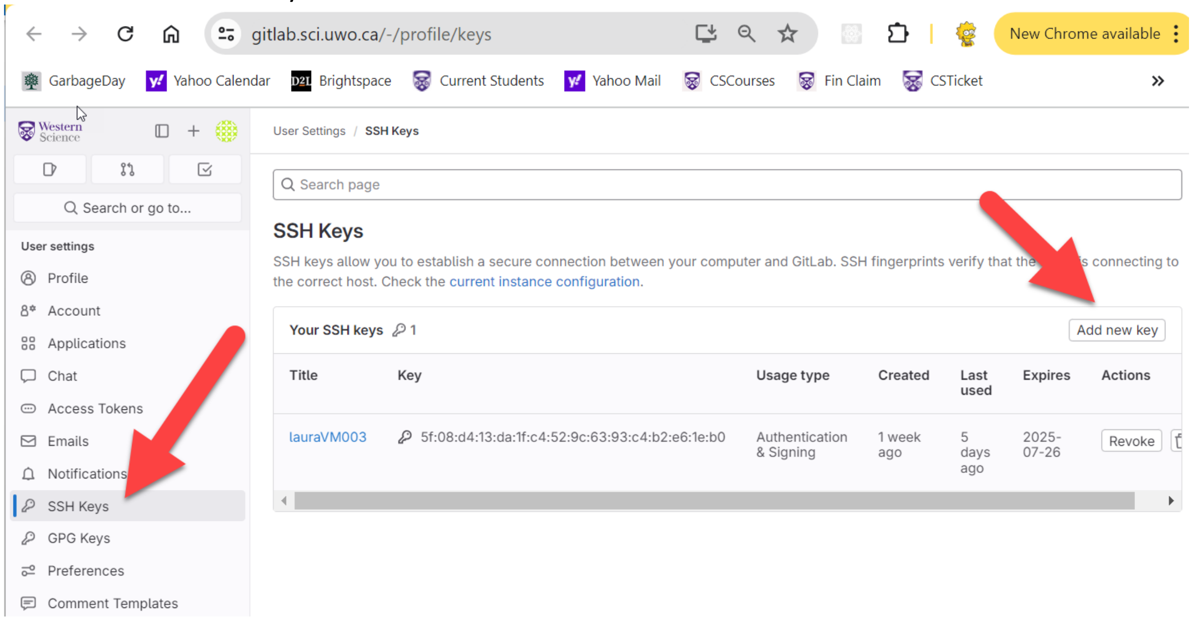
```
[rocky@cs3319-vm003 .ssh]$ ls -lg
total 12
-rw------- 1 rocky 399 Jul 23 13:26 authorized_keys
-rw------- 1 rocky 419 Jul 26 18:14 id_ed25519
-rw-r--r-- 1 rocky 110 Jul 26 18:14 id_ed25519.pub
[rocky@cs3319-vm003 .ssh]$ more id_ed25519.pub
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIO0LR57y38qtXXwCqZzPPu61C9Bt/c+1Yv6OxqzyfeCP rocky@cs3319-vm003.no
valocal
[rocky@cs3319-vm003 .ssh]$
```

6. Then type this so that you are in the correct directory to set up your repository:

```
cd ~/assignments
```

7. Leaving your virtual machine open in the background, open a browser and go to Western's Git area: https://gitlab.sci.uwo.ca and log in with your Western Id.

8. Click on this link https://gitlab.sci.uwo.ca/-/user_settings/ssh_keys and select SSH Keys and then click on the Add new key button:



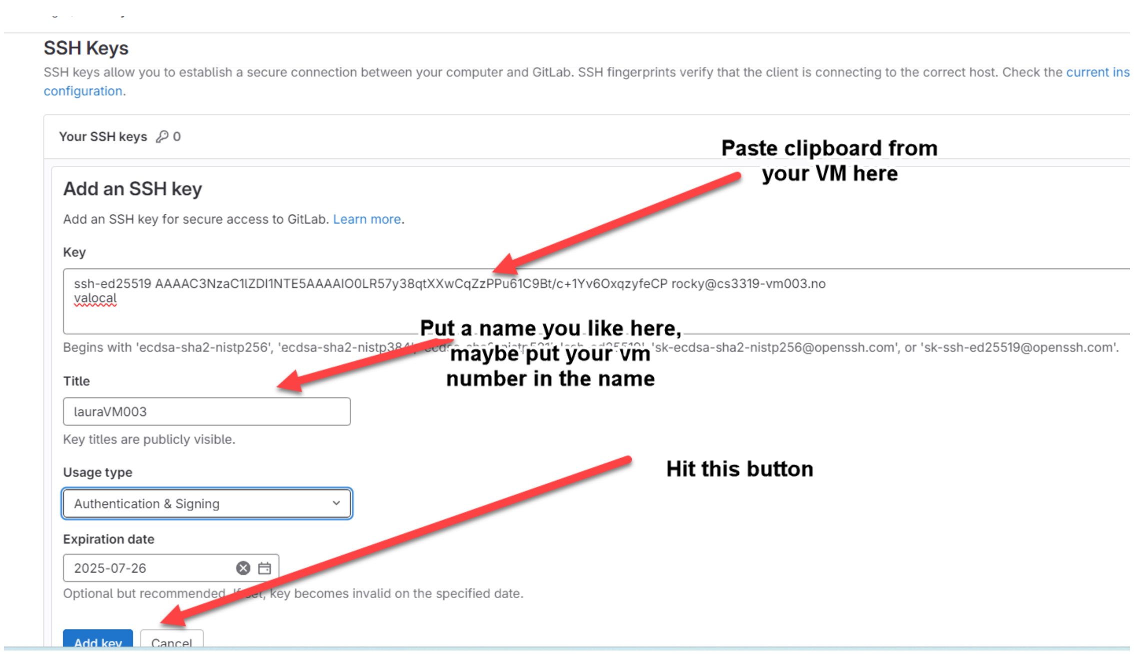
SSH Keys

SSH keys allow you to establish a secure connection between your computer and GitLab. SSH fingerprints verify that the s connecting to the correct host. Check the current instance configuration.

Your SSH keys 1

Title	Key	Usage type	Created	Last used	Expires	Actions
lauraVM003	5f:08:d4:13:da:1f:c4:52:9c:63:93:c4:b2:e6:1e:b0	Authentication & Signing	1 week ago	5 days ago	2025-07-26	Revoke

9. Paste the entire contents that you just copied from the clipboard (from *myvmkey.pub* file) into the box and give your key a title (in my example I put lauraVM003 but you can pick any name you want but don't put spaces in it), make sure the expiration date is past the end of the term (or you could leave it blank) and then hit the **Add key** button.


Paste clipboard from your VM here
Put a name you like here, maybe put your vm number in the name
Hit this button

SSH Keys

SSH keys allow you to establish a secure connection between your computer and GitLab. SSH fingerprints verify that the client is connecting to the correct host. Check the current ins configuration.

Your SSH keys 0

Add an SSH key

Add an SSH key for secure access to GitLab. Learn more.

Key

```
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIO0LR57y38qtXXwCqZzPPu61C9Bt/c+1Yv6OxqzyfeCP rocky@cs3319-vm003.no
valocal
```

Begins with 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384', ... 'sk-ecdsa-sha2-nistp256@openssh.com', or 'sk-ssh-ed25519@openssh.com'.

Title: lauraVM003

Key titles are publicly visible.

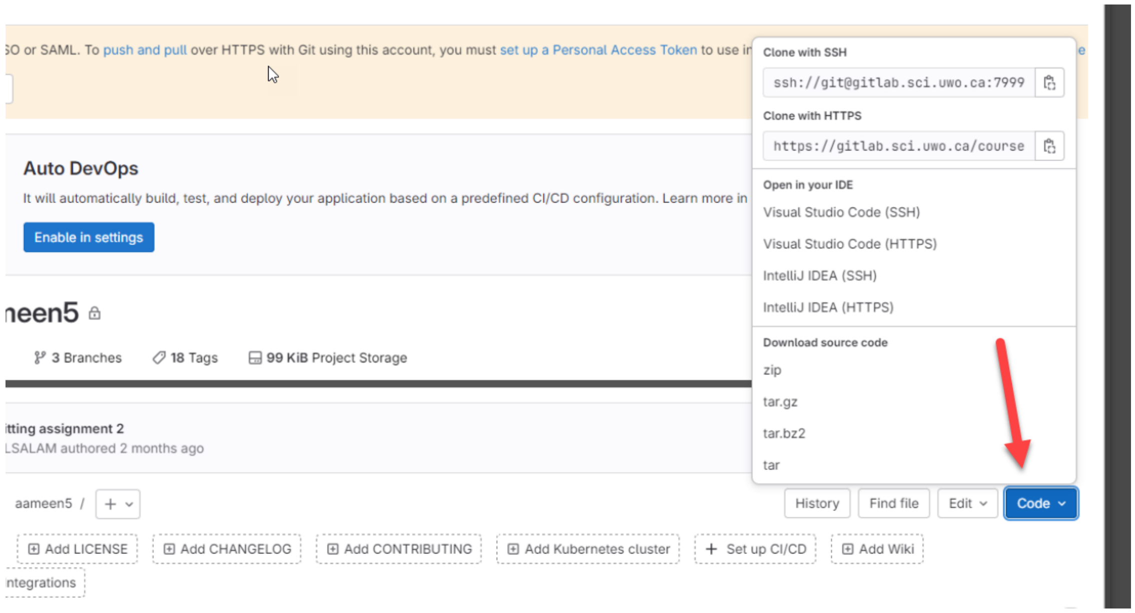
Usage type: Authentication & Signing

Expiration date: 2025-07-26

Optional but recommended. If set, key becomes invalid on the specified date.

Add key | Cancel

10. In the browser in gitlab.sci.uwo.ca, click on the *Search or go to…* button and search for your repository (it will likely be your username). Click on the blue button called Code

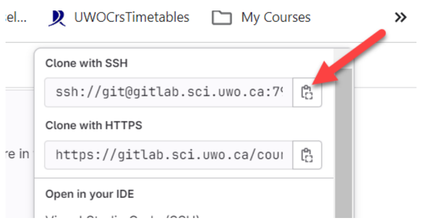


SO or SAML. To push and pull over HTTPS with Git using this account, you must set up a Personal Access Token to use in

Auto DevOps

It will automatically build, test, and deploy your application based on a predefined CI/CD configuration. Learn more in

11. Click on the button next to the ssh URL to copy this URL **(MAKE SURE YOU COPY THE ssh url NOT the https url):**



12. Now you want to type in the commands into your Virtual Machine to set everything up. Make sure you are in your assignments directory on your virtual machine (just type ***cd ~/assignments)***

    Here are the commands you will need to type on your VM, so go back to your VM terminal and I just put the first cd commands to make sure you are in the correct directory (put ***yes*** if prompted). MAKE SURE YOU CHANGE red and yellow part to be what you copied into your clipboard when you pressed the button above.

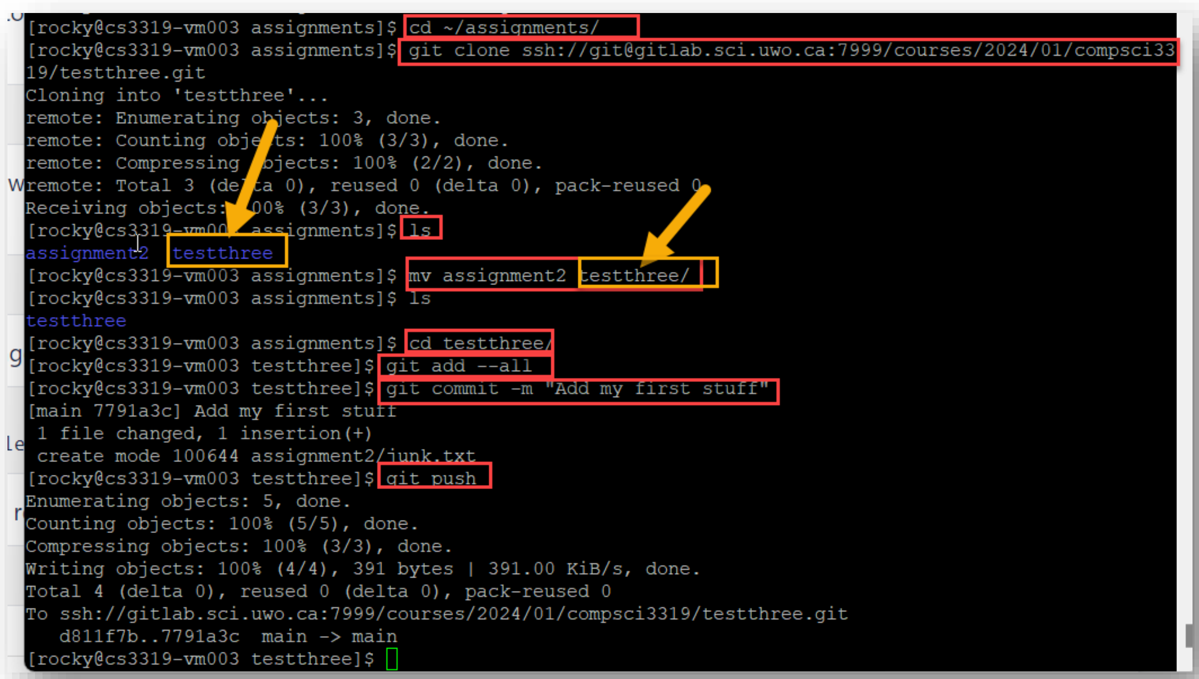
This yellow & red part will be specific for YOUR repo account, that you just copied by pressing the button next to Clone with SSH

```
cd ~/assignments
git clone ssh://git@gitlab.sci.uwo.ca:7999/courses/2024/01/compsci3319/testthree.git


ls
mv assignment2 yourreponame/
cd yourreponame


git add --all
git commit -m "Add my first stuff"
git push
```

Do the ls command so that you see that your repo is there and what its name is. Mine looks slightly different than yours because I am an instructor . Then move your assignment2 folder into your repo name (it will likely be your Western userid), then change directory (cd) and go into your repository.

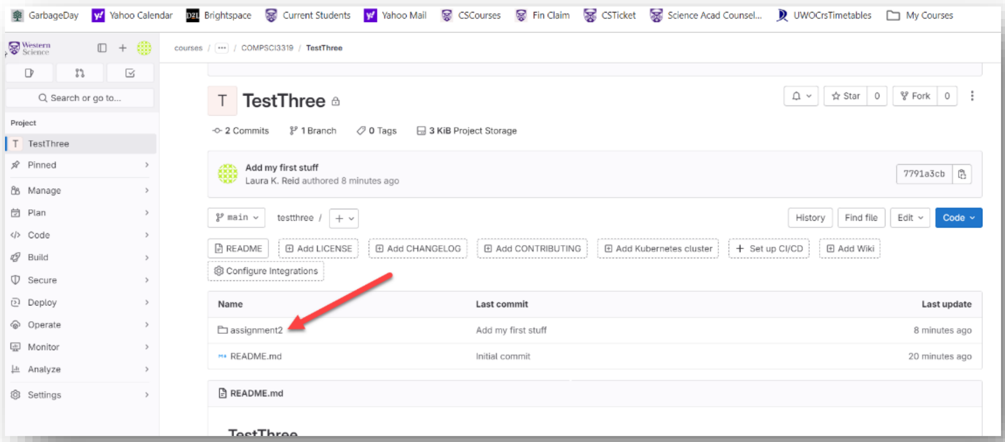
Push everything to the repository in gitlab.

13. If everything worked, you should see something like this (the part in yellow will be different for you, it will be your repo name (likely your Western userid):



```
[rocky@cs3319-vm003 assignments]$ cd ~/assignments/
[rocky@cs3319-vm003 assignments]$ git clone ssh://git@gitlab.sci.uwo.ca:7999/courses/2024/01/compsci33
19/testthree.git
Cloning into 'testthree'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
[rocky@cs3319-vm003 assignments]$ ls
assignment2  testthree
[rocky@cs3319-vm003 assignments]$ mv assignment2 testthree/
[rocky@cs3319-vm003 assignments]$ ls
testthree
[rocky@cs3319-vm003 assignments]$ cd testthree/
[rocky@cs3319-vm003 testthree]$ git add --all
[rocky@cs3319-vm003 testthree]$ git commit -m "Add my first stuff"
[main 7791a3c] Add my first stuff
 1 file changed, 1 insertion(+)
 create mode 100644 assignment2/junk.txt
[rocky@cs3319-vm003 testthree]$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 391 bytes | 391.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To ssh://gitlab.sci.uwo.ca:7999/courses/2024/01/compsci3319/testthree.git
   d811f7b..7791a3c  main -> main
[rocky@cs3319-vm003 testthree]$
```

14. Then go back to your browser and hit the refresh button in the gitlab.sci.uwo.ca tab and now you should see your code from your Virtual Machine inside your gitlab.sci.uwo.ca area (notice the red arrow is pointing to a copy of the directory we created on our VM:



TestThree

2 Commits 1 Branch 0 Tags 3 KiB Project Storage

Add my first stuff
Laura K. Reid authored 8 minutes ago 7791a3cb

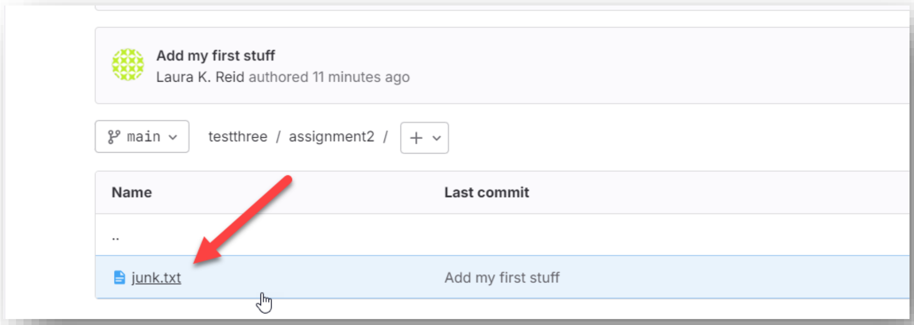
main testthree /

Name	Last commit	Last update
assignment2	Add my first stuff	8 minutes ago
README.md	Initial commit	20 minutes ago

README.md

TestThree

Then click on the assignment 2 directory and you should see the junk.txt file that we put in that directory:

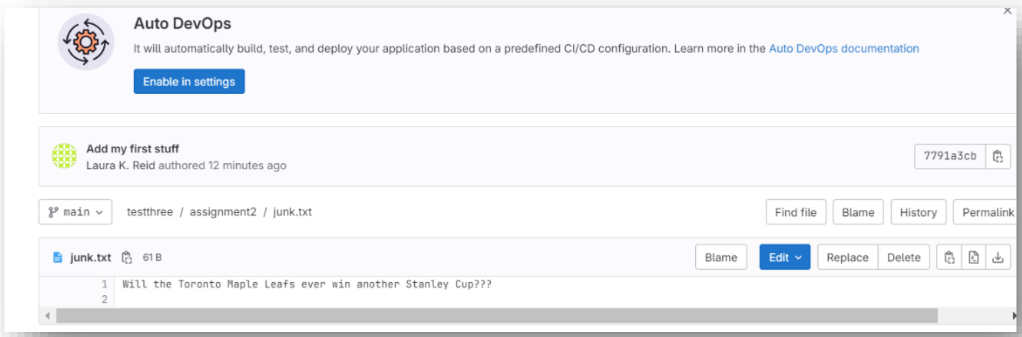


Add my first stuff
Laura K. Reid authored 11 minutes ago

main testthree / assignment2 /

Name	Last commit
..	
junk.txt	Add my first stuff

15. Now, in the browser, click into the junk.txt folder, you should see something like this:



Auto DevOps

It will automatically build, test, and deploy your application based on a predefined CI/CD configuration. Learn more in the Auto DevOps documentation

Add my first stuff
Laura K. Reid authored 12 minutes ago 7791a3cb

main testthree / assignment2 / junk.txt

junk.txt 61 B

```
Will the Toronto Maple Leafs ever win another Stanley Cup???
```

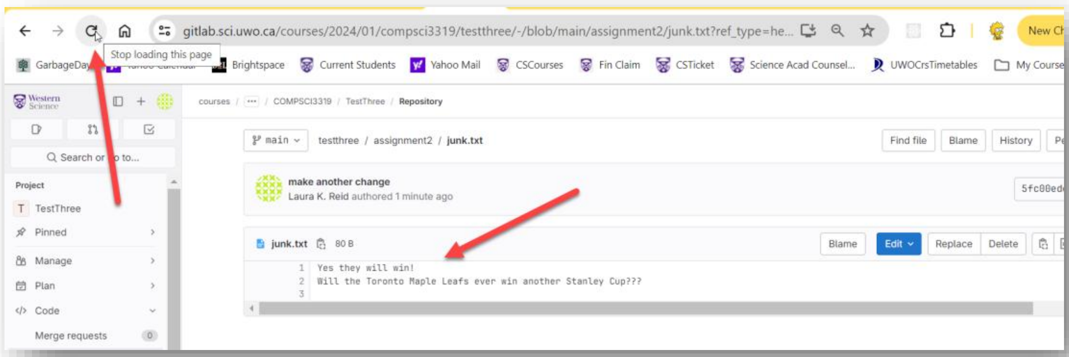
16. Go back to the Virtual Machine terminal and type something like this to change that junk.txt file:

```
cd ~/assignments/yourreponame/assignment2
nano junk.txt

Modify the junk.txt file so that it says:

 "YES THEY WILL WIN!"
(or type Hello World, or Laura is Evil or anything you want!)

Hit Ctrl-O, enter, Ctrl-X to save the file and exit
```

17. Then, type the following 3 commands to PUSH your changes that you just made to the junk.txt file in your Virtual Machine up to your gitlab Code Repository:

```
git add .
git commit -m "made another change"
git push
```

18. Now go back to your browsor and hit refresh again, and you show now see something like this:



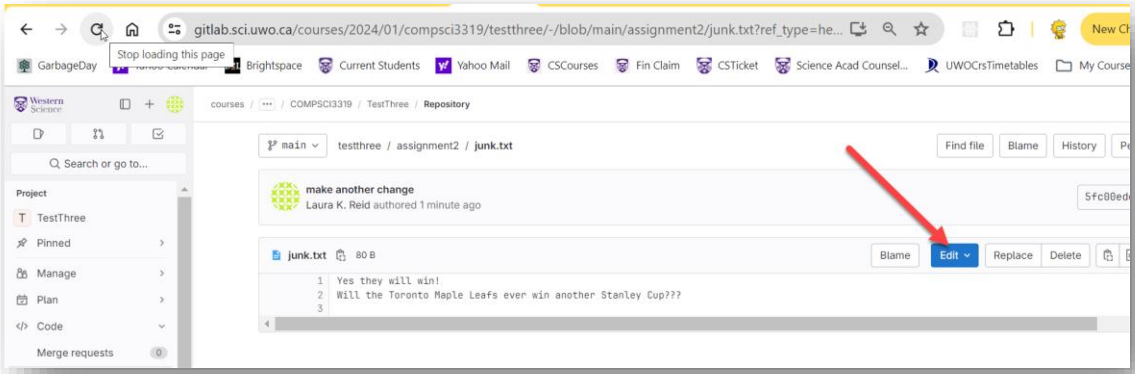
make another change
Laura K. Reid authored 1 minute ago
5fc80ed

```
Yes they will win!
Will the Toronto Maple Leafs ever win another Stanley Cup???
```

19. Let's recap → Everytime you want to do a backup of your files on your VM (maybe do it every 2 or 3 hours, or when you log off), make sure you are in your folder with the files you have been working on and then you need to only type these 3 commands to PUSH the files from your VM up to the code repository in gitlab.sci.uwo.ca. These are the ONLY 3 commands you will likely need from now on to get your code from your VM to your BitBucket

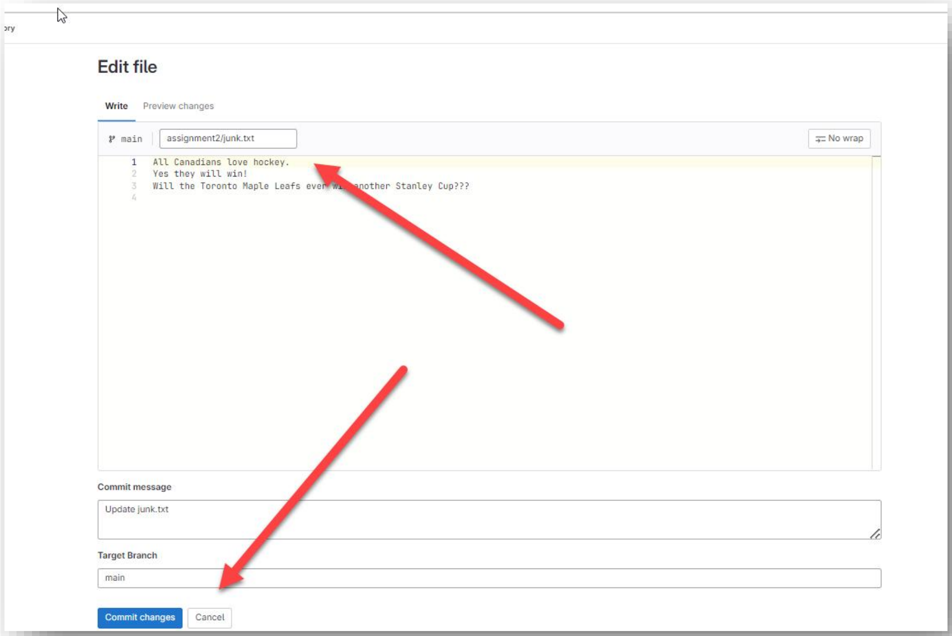
```
git add .
git commit -m "some message"
git push
```

20. If you don't like using the nano editor, you could use the gitlab.csd.uwo.ca editor instead and **PULL** files FROM the gitlab.sci.uwo.ca onto your VM (rather than pushing the way we were doing above). These next steps will show you how to do that.
21. In your browser, make sure you are clicked into your junk.txt file. Then click on the Edit button:



```
Yes they will win!
Will the Toronto Maple Leafs ever win another Stanley Cup???
```

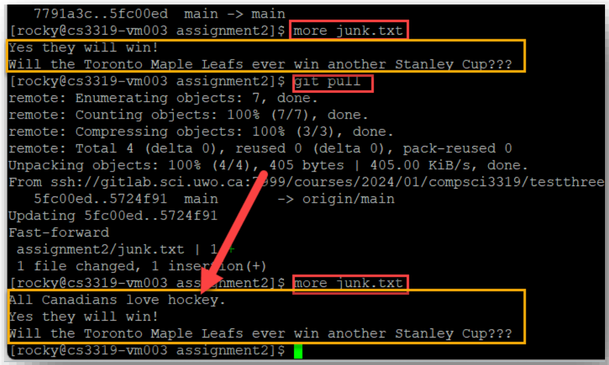
22. Select edit single file only. Now you can edit the junk.txt file here. Add another sentence to the file and then click on the Commit Changes button:



```
All Canadians love hockey.
Yes they will win!
Will the Toronto Maple Leafs ever win another Stanley Cup???
```

23. Now the code is the way you want it in your code repository but it is not yet on your VM. So now you must PULL it onto your VM by typing the following *git pull* command on your VM (the *more* commands below are just so that you can see the file before and after the *git pull* command):

```
more junk.txt
git pull
more junk.txt
```

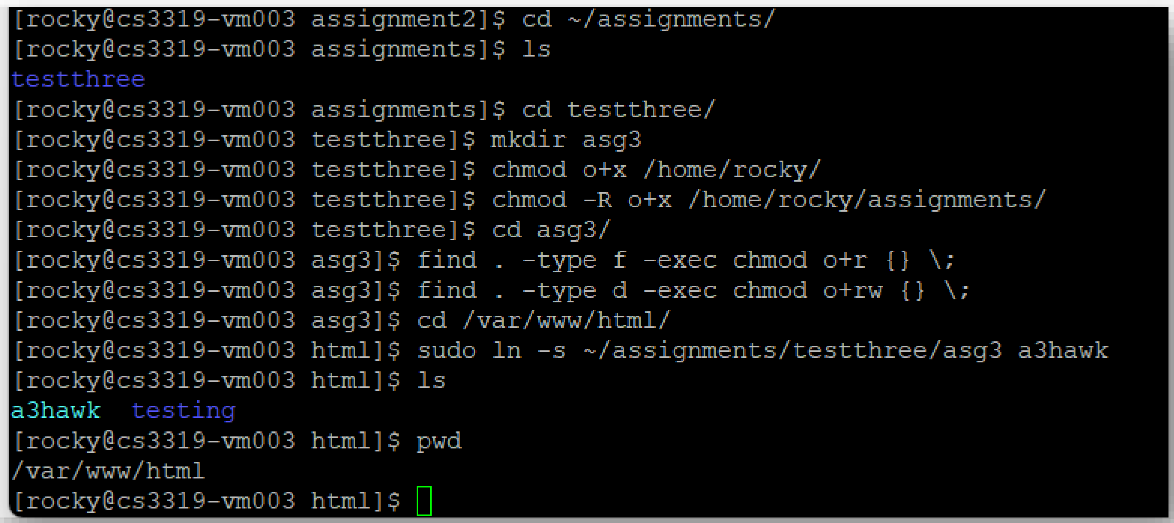


```
   7791a3c..5fc00ed  main -> main
[rocky@cs3319-vm003 assignment2]$ more junk.txt
Yes they will win!
Will the Toronto Maple Leafs ever win another Stanley Cup???
[rocky@cs3319-vm003 assignment2]$ git pull
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (4/4), 405 bytes | 405.00 KiB/s, done.
From ssh://gitlab.sci.uwo.ca:7999/courses/2024/01/compsci3319/testthree
   5fc00ed..5724f91  main       -> origin/main
Updating 5fc00ed..5724f91
Fast-forward
 assignment2/junk.txt | 1 +
 1 file changed, 1 insertion(+)
[rocky@cs3319-vm003 assignment2]$ more junk.txt
All Canadians love hockey.
Yes they will win!
Will the Toronto Maple Leafs ever win another Stanley Cup???
[rocky@cs3319-vm003 assignment2]$
```

24. Again, let's recap → if you do your changes on the VM, then you need to do the 3 commands of ***git add ., git commit -m "message", git push*** in order to PUSH your changes *from* your VM *to* your repository BUT if you do your changes on your repository, then you need to do ***git pull*** to PULL your changes *from* your repo *onto* your VM.
25. WARNINGS:
    1. you do not want your VM to get out of sync with your repository so always do *git pull* when you first log into your VM at the beginning of a new day and start working on your assignment. That way you will always have the latest version on your gitlab.sci.uwo.ca area before you make any changes.
    2. To hand in assignment 3, you need to zip all your files together and then push that .zip file to your gitlab area. Then you can get the .zip file to hand into kritik.io along with your URL.
26. There is one last thing we need to do. For assignment 3, you need to put your .php files in the folder called ***/var/www/html*** in order to test and run your program in a browser. So we need to be able to push and pull from that folder as well as the assignments folder so that you don't lose your code. It gets a bit messy if we just keep putting (cloning) your respostory all over the place, so we are going to do a little trick to avoid having to put the whole repository into /var/www/html. We are going to make a directory under your ~/assignments/yourrepo folder called asg3 and then symlink it (i.e. create a shortcut) to it in /var/www/html. That way we can modify files, add files, make directories and do all that within ~/assignments/yourrepo/asg3 where the repository is located BUT all the changes we make will be visible on our shortcut inside of /var/www/html (i.e. visible on the internet within a browser). The other issue is that we don't want students to be able to see each others assignments for assignment 3. Thus we are going to symlink your asg3 folder to a folder in /var/www/html called a3yourfavouriteanimal. The reason we are calling it a3yourfavouriteanimal rather than asg3 is that we want to make the folder name hard to guess so that your classmates can't go your URL and just type asg3 as the end of the URL for your machine number to see your all your hard work on your virtual machine and then copy your .html and css code (your lovely user interface that you worked hard on!). They will never be able to copy your .php code because it is always hidden but they could copy your html/css code, thus we are going to make it hard for them to guess at it by putting your favourite animal in the folder name. Then we will push the new folder asg3 into the repository (remember: git add . , git commit, git push). First we need to create the directories and set the permissions so that Apache can navigate the folders. So type the following commands (make sure you put YOUR favourite animal after a3 in the last command below, no spaces, such as a3chicken) : NOTICE that you now have a directory called *a3yourfavanimal* in the */var/www/html* Make sure you change the yellow parts below to be your gitlab repository name (likely your Western UserID) and the green part to be your favourite animal after a3.

```
cd ~/assignments/yourrepo
mkdir asg3
chmod o+x /home/rocky/
chmod -R o+x /home/rocky/assignments/
cd asg3
find . -type f -exec chmod o+r {} \;
find . -type d -exec chmod o+rx {} \;
cd /var/www/html/
sudo ln -s ~/assignments/yourrepo/asg3 a3yourfavouriteanimal
ls
pwd
```

Here is what it looks like on my machine:



```
[rocky@cs3319-vm003 assignment2]$ cd ~/assignments/
[rocky@cs3319-vm003 assignments]$ ls
testthree
[rocky@cs3319-vm003 assignments]$ cd testthree/
[rocky@cs3319-vm003 testthree]$ mkdir asg3
[rocky@cs3319-vm003 testthree]$ chmod o+x /home/rocky/
[rocky@cs3319-vm003 testthree]$ chmod -R o+x /home/rocky/assignments/
[rocky@cs3319-vm003 testthree]$ cd asg3/
[rocky@cs3319-vm003 asg3]$ find . -type f -exec chmod o+r {} \;
[rocky@cs3319-vm003 asg3]$ find . -type d -exec chmod o+rw {} \;
[rocky@cs3319-vm003 asg3]$ cd /var/www/html/
[rocky@cs3319-vm003 html]$ sudo ln -s ~/assignments/testthree/asg3 a3hawk
[rocky@cs3319-vm003 html]$ ls
a3hawk  testing
[rocky@cs3319-vm003 html]$ pwd
/var/www/html
[rocky@cs3319-vm003 html]$ 
```

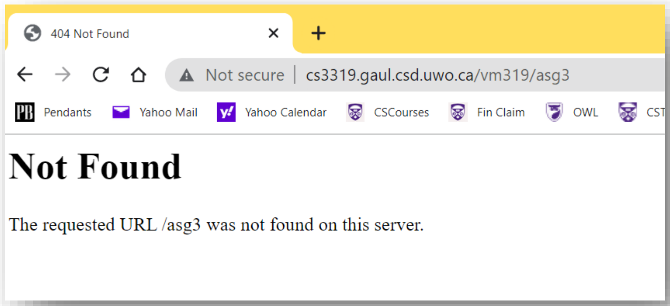
27. Now go back to your asg3 file and add a little test file as follows:

```
cd ~/assignments/yourrepo/asg3
nano testing.html
```

and then type the following html code into the testing.html file and then hit CTRL-O, hit Enter and then hit CTRL-X to save your file (change the yellow to your first name):

```
<html>
<head>
<title>
This is a test
</title>
</head>
<body>
Laura is testing the repository
</body>
</html>
```

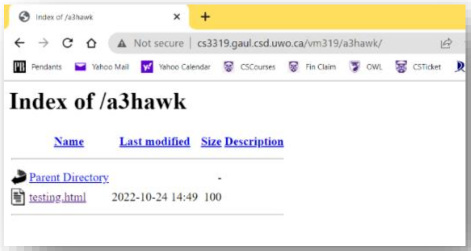
28. Now we are going to first make sure this little webpage is visible. In a browser, go to:
    https://cs3319.gaul.csd.uwo.ca/vm???/asg3/
    where ??? is your vm number. Make sure you put the / at the end of the URL or the url won't work. NOTICE that it is not found. That is because /var/www/html thinks the folder name is a3yourfavanimal NOT asg3.



Not Found

The requested URL /asg3 was not found on this server.

    Now try going to:
    https://cs3319.gaul.csd.uwo.ca/vm???/a3yourfavanimal/
    like I have here (again don't forget the / at the end) and just click on your testing.html file to make sure it shows up.



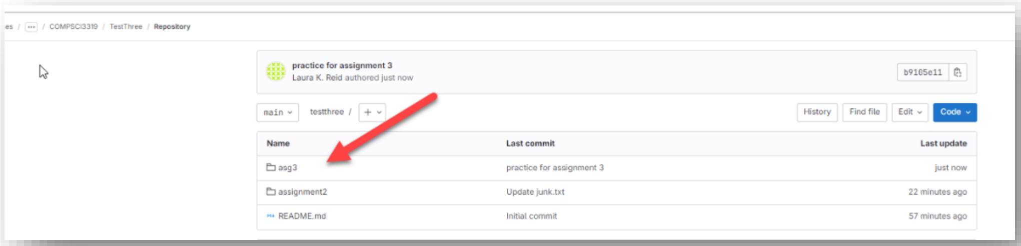
Index of /a3hawk

```
      Name              Last modified     Size Description
Parent Directory                            -
testing.html          2022-10-24 14:49  100
```

29. When you are ready to start assignment 3, you are going to put ALL of your assignment 3 files in that asg3 folder and then you are going to check if the files are working by going to: **https://cs3319.gaul.csd.uwo.ca/vm???/a3yourfavanimal/**
30. Now let's make sure that we can push that *testing.html* file we just made in your asg3 folder, up to your repository in *gitlab.csd.uwo.ca*. Since the *asg3* folder is inside the *assignments* folder (where we put our repository), our commands are quite simple. Go to your virtual machine and type the following commands (I am just doing the second command below, the *git pull*, to make sure that everything is in sync (the VM matches the gitlab.sci.uwo.ca), it is probably not necessary but just in case, we will do it also):

```
cd ~/assignments/yourrepo/asg3
ls
git pull
git add .
git commit -m "practice for assignment 3"
git push
```

31. Now go back to the browser to gitlab.sci.uwo.ca and do a Refresh and make sure that the asg3 folder exists:

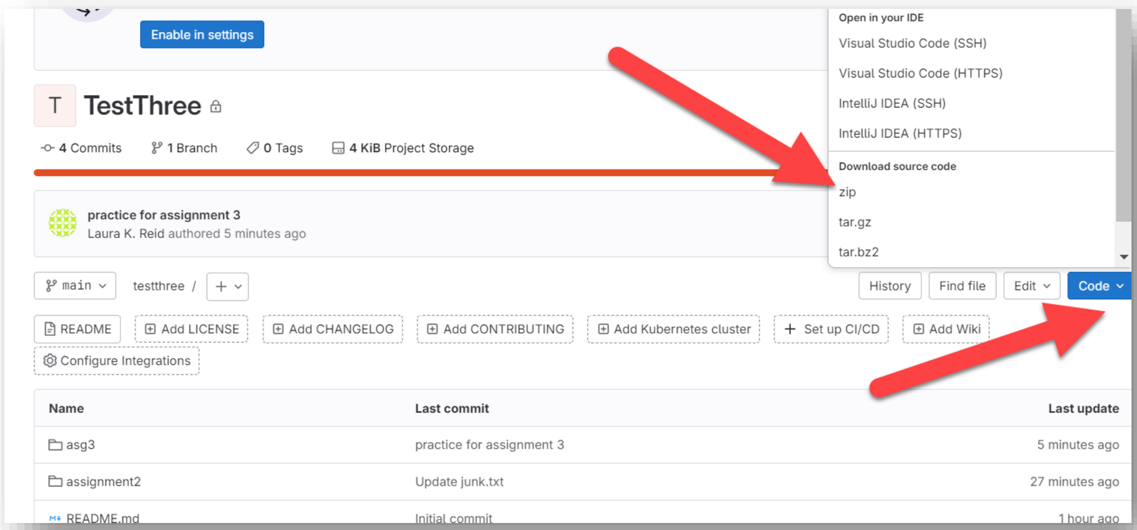


practice for assignment 3
Laura K. Reid authored just now
b9105e11

Name	Last commit	Last update
asg3	practice for assignment 3	just now
assignment2	Update junk.txt	22 minutes ago
README.md	Initial commit	57 minutes ago

32. Whew, that is all you will have to do in order to set up your assignment 3 directory and make sure it is part of your repository. Now, when you start working on assignment 3, all you will have to do is put your assignment 3 files in ~/assignments/yourepo/asg3 and then you can now push and pull files into the asg3 folder BUT asg3 is still hidden from your classmates (unless of course, they know your favourite animal and your virtual machine number …I REALLY love hawks… Hence the ***a3hawk*** folder name for me!)

33. One last thing: When you need to hand in your files for assignment 2 and 3, do NOT right click on the files in gitlab.sci.uwo.ca and do Save Link As. That will NOT download the file properly, it will be corrupt. In order to get the files correctly from gitlab.sci.uwo.ca, you MUST click on the blue Code button again, and then select Download source code .zip as shown in the image below. So for assignments 2 and 3, you will need to do this:



practice for assignment 3
Laura K. Reid authored 5 minutes ago

Name	Last commit	Last update
asg3	practice for assignment 3	5 minutes ago
assignment2	Update junk.txt	27 minutes ago
README.md	Initial commit	1 hour ago

34. This is one way to move files back and forth between your repository and your virtual machine AND the great thing about this method is that it backs up all your files for you on gitlab. Remember:

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

35. Now go to *Brightspace>Assessments>Assignments>Setting Up Your Virtual Machine* and submit a link to the testing folder you created above on Page 12 – step 4.7.g, i.e. a link to: http://cs3319.gaul.csd.uwo.ca/vm???/testing/junk.php where ??? is YOUR virtual machine number and when you click on junk.php, it displays your name. You get 1% for submitting this workshop to Owl>Assessments>Assignments>Setting Up Your Virtual Machine.

36. **CONGRATULATIONS →** you have now done the following in your own Linux Virtual Machine:
    1. Logged onto a clean Virtual Machine and set it up
    2. Installed all the lastest packages
    3. Installed a web server
    4. Installed two editors
    5. Installed PHP
    6. Installed a database management system called MySQL
    7. Installed a code repository system
    8. Pushed and pulled to a code repository
    9. Created and posted a simple webpage
    10. Edited a configuration file to display errors
    11. Checked the webpage and checked that errors are displaying
    12. Installed zip and unzip software on your Virtual Machine
    13. Tested MySQL on your Virtual Machine
    14. Set up a public/private key pair on your Virual Machine so that you can push and pull files and folders to and from the gitlab repository that was created for you at Western.
    15. Set up your folder for assignment 3 and put it into your gitlab Repository.
    16. Handed in a link to your virtual machine to prove everything is working.
    17. And always remember… in case of fire: Git add, Git commit, Git push and then Git out of the building!

EY KARUMBA…that was a lot of work!
WELL DONE ☺