Name:John Edward Miles D. Espiritu	**Date Performed:01/22/2024**
Course/Section:CPE232-31S1	**Date Submitted:01/23/2024**
Instructor: Sir Jonathan Taylar	**Semester and SY: 2nd sem/2024-2025**

Activity 2: SSH Key-Based Authentication and Setting up Git

1. **Objectives:**

1.1 Configure remote and local machine to connect via SSH using a KEY instead of using a password

1.2 Create a public key and private key

1.3 Verify connectivity

1.4 Setup Git Repository using local and remote repositories

1.5 Configure and Run ad hoc commands from local machine to remote servers

Part 1: Discussion

It is assumed that you are already done with the last Activity (**Activity 1: Configure Network using Virtual Machines).** *Provide screenshots for each task*.

It is also assumed that you have VMs running that you can SSH but requires a password. Our goal is to remotely login through SSH using a key without using a password. In this activity, we create a public and a private key. The private key resides in the local machine while the public key will be pushed to remote machines. Thus, instead of using a password, the local machine can connect automatically using SSH through an authorized key.

What Is ssh-keygen?

Ssh-keygen is a tool for creating new authentication key pairs for SSH. Such key pairs are used for automating logins, single sign-on, and for authenticating hosts.

SSH Keys and Public Key Authentication

The SSH protocol uses public key cryptography for authenticating hosts and users. The authentication keys, called SSH keys, are created using the keygen program.

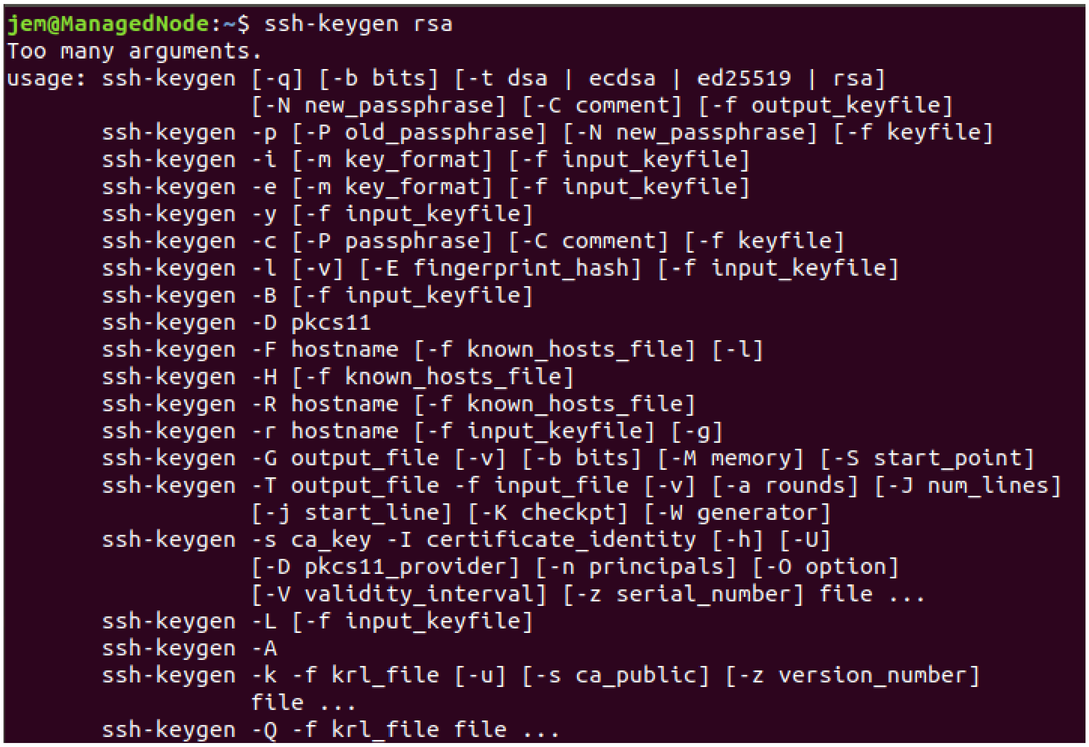
SSH introduced public key authentication as a more secure alternative to the older .rhosts authentication. It improved security by avoiding the need to have password stored in files and eliminated the possibility of a compromised server stealing the user's password.

However, SSH keys are authentication credentials just like passwords. Thus, they must be managed somewhat analogously to usernames and passwords. They should have a proper termination process so that keys are removed when no longer needed.

Task 1: Create an SSH Key Pair for User Authentication

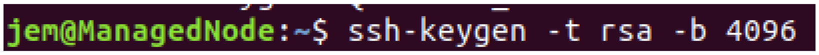
1. The simplest way to generate a key pair is to run *ssh-keygen* without arguments. In this case, it will prompt for the file in which to store keys. First,

the tool asked where to save the file. SSH keys for user authentication are usually stored in the users .ssh directory under the home directory. However, in enterprise environments, the location is often different. The default key file name depends on the algorithm, in this case *id_rsa* when using the default RSA algorithm. It could also be, for example, *id_dsa* or *id_ecdsa*.



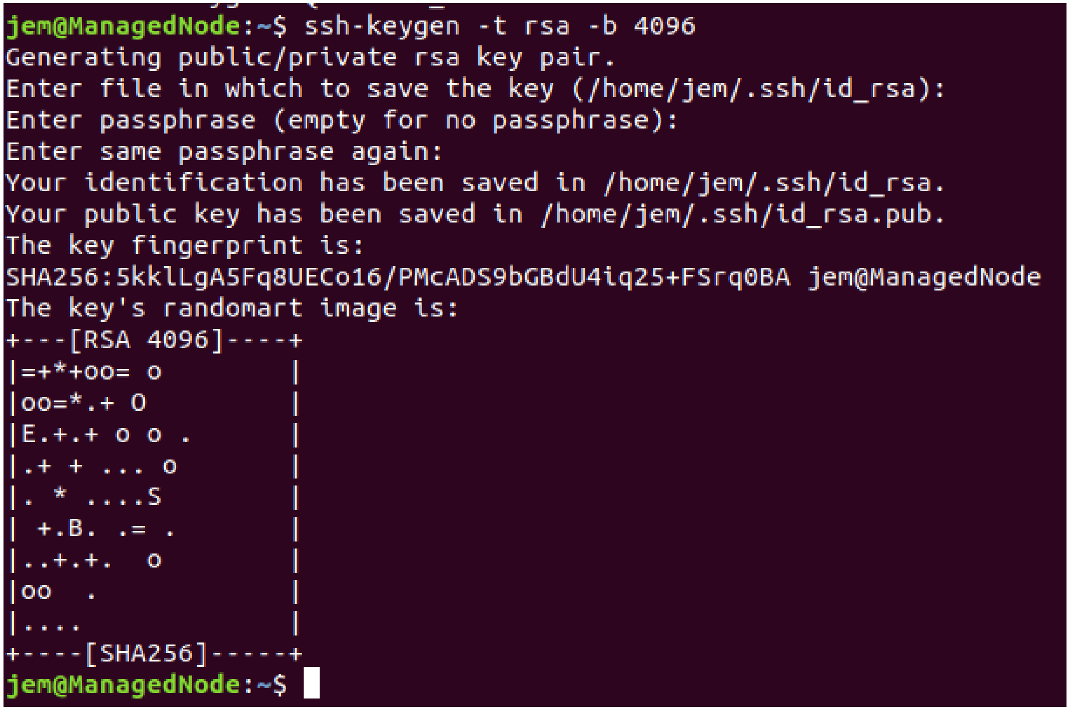
```
jem@ManagedNode:~$ ssh-keygen rsa
Too many arguments.
usage: ssh-keygen [-q] [-b bits] [-t dsa | ecdsa | ed25519 | rsa]
                  [-N new_passphrase] [-C comment] [-f output_keyfile]
       ssh-keygen -p [-P old_passphrase] [-N new_passphrase] [-f keyfile]
       ssh-keygen -i [-m key_format] [-f input_keyfile]
       ssh-keygen -e [-m key_format] [-f input_keyfile]
       ssh-keygen -y [-f input_keyfile]
       ssh-keygen -c [-P passphrase] [-C comment] [-f keyfile]
       ssh-keygen -l [-v] [-E fingerprint_hash] [-f input_keyfile]
       ssh-keygen -B [-f input_keyfile]
       ssh-keygen -D pkcs11
       ssh-keygen -F hostname [-f known_hosts_file] [-l]
       ssh-keygen -H [-f known_hosts_file]
       ssh-keygen -R hostname [-f known_hosts_file]
       ssh-keygen -r hostname [-f input_keyfile] [-g]
       ssh-keygen -G output_file [-v] [-b bits] [-M memory] [-S start_point]
       ssh-keygen -T output_file -f input_file [-v] [-a rounds] [-J num_lines]
                  [-j start_line] [-K checkpt] [-W generator]
       ssh-keygen -s ca_key -I certificate_identity [-h] [-U]
                  [-D pkcs11_provider] [-n principals] [-O option]
                  [-V validity_interval] [-z serial_number] file ...
       ssh-keygen -L [-f input_keyfile]
       ssh-keygen -A
       ssh-keygen -k -f krl_file [-u] [-s ca_public] [-z version_number]
                  file ...
       ssh-keygen -Q -f krl_file file ...
```

2. Issue the command *ssh-keygen -t rsa -b 4096.* The algorithm is selected using the -t option and key size using the -b option.



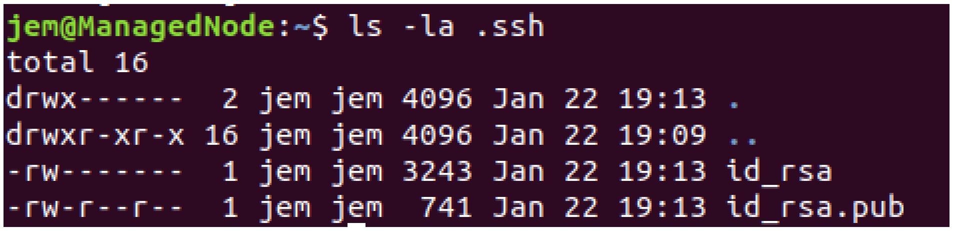
```
jem@ManagedNode:~$ ssh-keygen -t rsa -b 4096
```

3. When asked for a passphrase, just press enter. The passphrase is used for encrypting the key, so that it cannot be used even if someone obtains the private key file. The passphrase should be cryptographically strong.



```
jem@ManagedNode:~$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/jem/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/jem/.ssh/id_rsa.
Your public key has been saved in /home/jem/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:5kklLgA5Fq8UECo16/PMcADS9bGBdU4iq25+FSrq0BA jem@ManagedNode
The key's randomart image is:
+---[RSA 4096]----+
|=+*+oo= o        |
|oo=*.+ O         |
|E.+.+ o o .      |
|.+ + ... o       |
|. * ....S        |
| +.B. .= .       |
|..+.+.  o        |
|oo  .            |
|....             |
+----[SHA256]-----+
jem@ManagedNode:~$
```

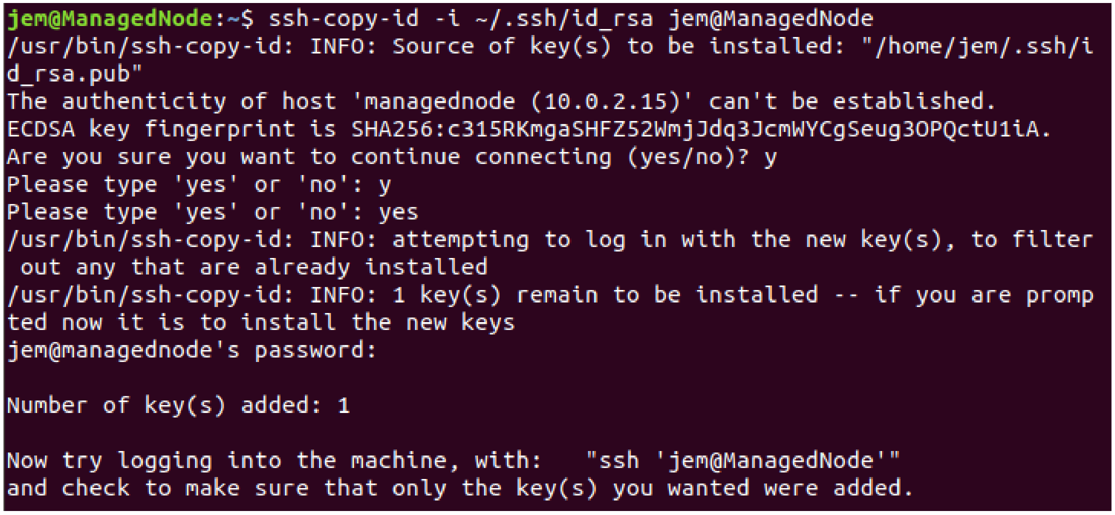
4. Verify that you have created the key by issuing the command *ls -la .ssh.* The command should show the .ssh directory containing a pair of keys. For example, id_rsa.pub and id_rsa.



```
jem@ManagedNode:~$ ls -la .ssh
total 16
drwx------  2 jem jem 4096 Jan 22 19:13 .
drwxr-xr-x 16 jem jem 4096 Jan 22 19:09 ..
-rw-------  1 jem jem 3243 Jan 22 19:13 id_rsa
-rw-r--r--  1 jem jem  741 Jan 22 19:13 id_rsa.pub
```

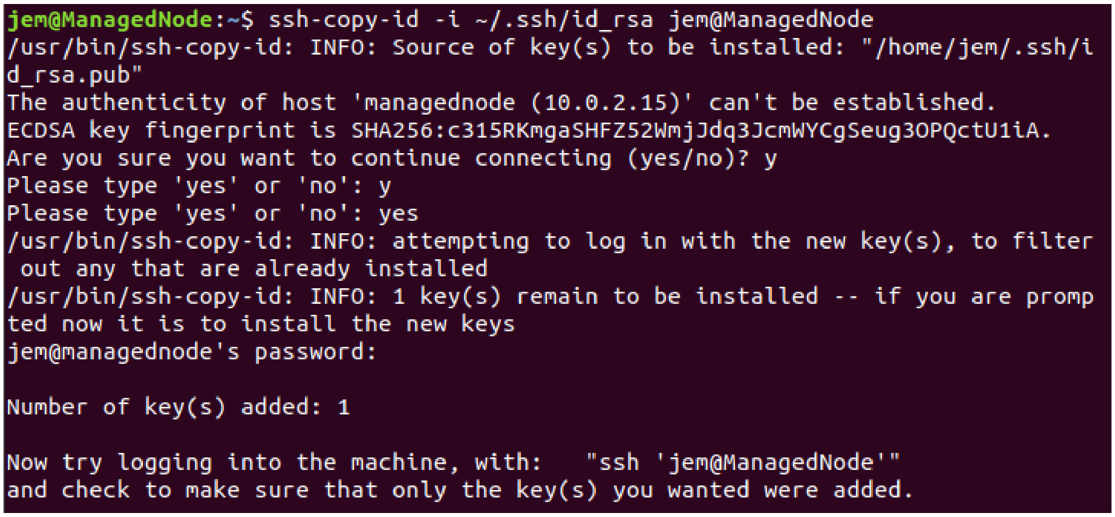
Task 2: Copying the Public Key to the remote servers

1. To use public key authentication, the public key must be copied to a server and installed in an *authorized_keys* file. This can be conveniently done using the *ssh-copy-id* tool.



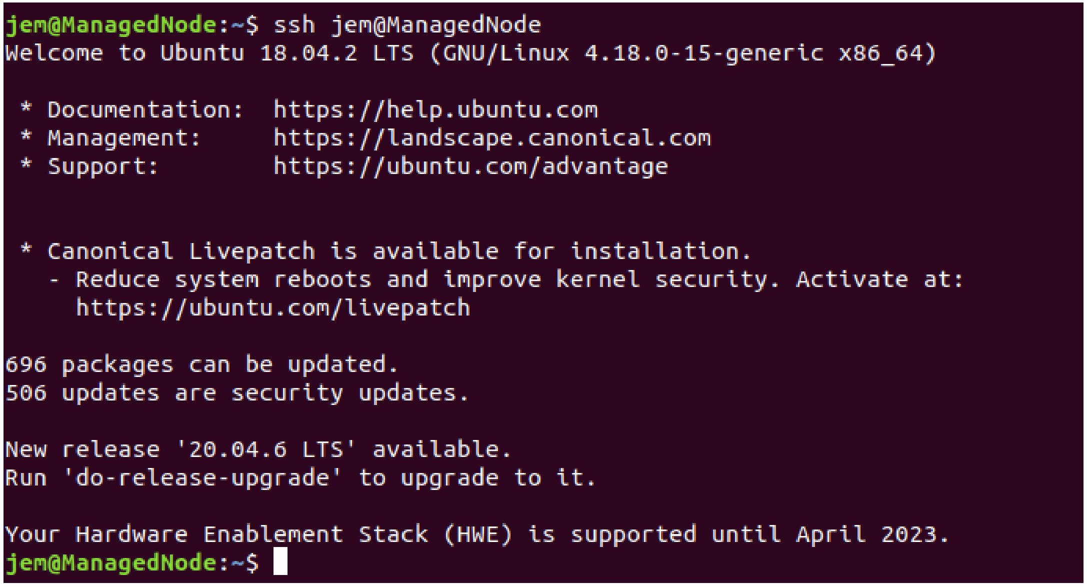
```
jem@ManagedNode:~$ ssh-copy-id -i ~/.ssh/id_rsa jem@ManagedNode
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/jem/.ssh/i
d_rsa.pub"
The authenticity of host 'managednode (10.0.2.15)' can't be established.
ECDSA key fingerprint is SHA256:c315RKmgaSHFZ52WmjJdq3JcmWYCgSeug3OPQctU1iA.
Are you sure you want to continue connecting (yes/no)? y
Please type 'yes' or 'no': y
Please type 'yes' or 'no': yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter
 out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are promp
ted now it is to install the new keys
jem@managednode's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'jem@ManagedNode'"
and check to make sure that only the key(s) you wanted were added.
```

2. Issue the command similar to this: *ssh-copy-id -i ~/.ssh/id_rsa user@host*



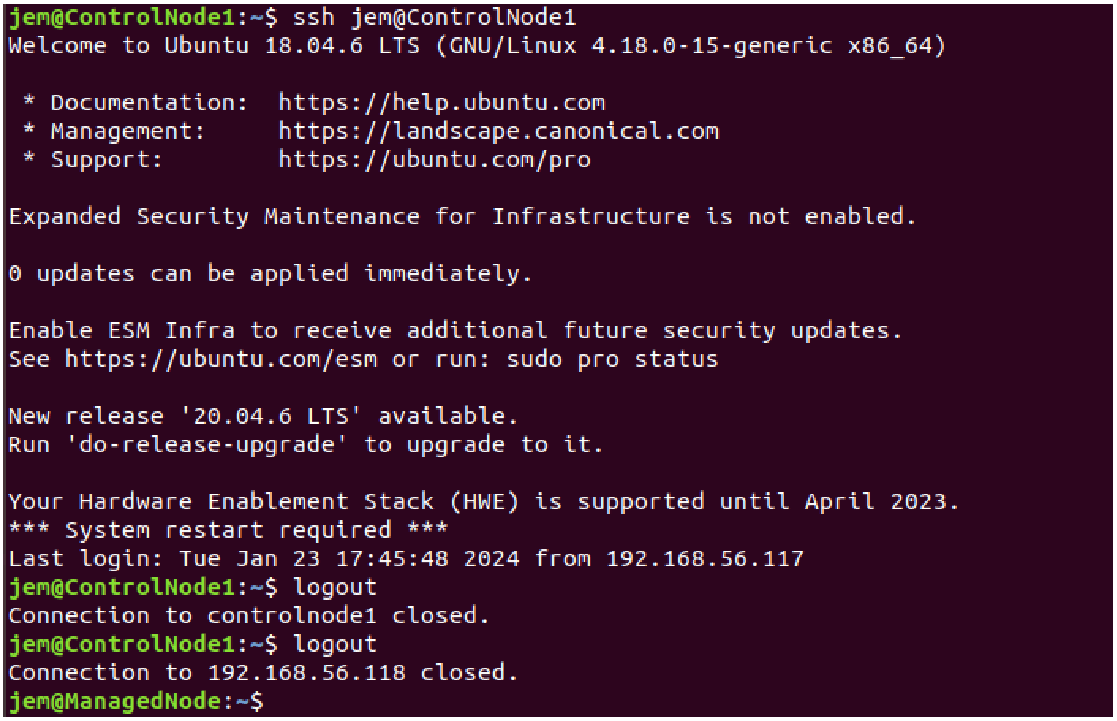
```
jem@ManagedNode:~$ ssh-copy-id -i ~/.ssh/id_rsa jem@ManagedNode
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/jem/.ssh/i
d_rsa.pub"
The authenticity of host 'managednode (10.0.2.15)' can't be established.
ECDSA key fingerprint is SHA256:c315RKmgaSHFZ52WmjJdq3JcmWYCgSeug3OPQctU1iA.
Are you sure you want to continue connecting (yes/no)? y
Please type 'yes' or 'no': y
Please type 'yes' or 'no': yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter
 out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are promp
ted now it is to install the new keys
jem@managednode's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'jem@ManagedNode'"
and check to make sure that only the key(s) you wanted were added.
```

3. Once the public key has been configured on the server, the server will allow any connecting user that has the private key to log in. During the login process, the client proves possession of the private key by digitally signing the key exchange.



```
jem@ManagedNode:~$ ssh jem@ManagedNode
Welcome to Ubuntu 18.04.2 LTS (GNU/Linux 4.18.0-15-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage


 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

696 packages can be updated.
506 updates are security updates.

New release '20.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Your Hardware Enablement Stack (HWE) is supported until April 2023.
jem@ManagedNode:~$ 
```

4. On the local machine, verify that you can SSH with Server 1 and Server 2. What did you notice? Did the connection ask for a password? If not, why?



```
jem@ControlNode1:~$ ssh jem@ControlNode1
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 4.18.0-15-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

Expanded Security Maintenance for Infrastructure is not enabled.

0 updates can be applied immediately.

Enable ESM Infra to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

New release '20.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Your Hardware Enablement Stack (HWE) is supported until April 2023.
*** System restart required ***
Last login: Tue Jan 23 17:45:48 2024 from 192.168.56.117
jem@ControlNode1:~$ logout
Connection to controlnode1 closed.
jem@ControlNode1:~$ logout
Connection to 192.168.56.118 closed.
jem@ManagedNode:~$
```

Reflections:

Answer the following:

1. How will you describe the ssh-program? What does it do?
   - **It is used to establish a secure connection between two computers and encrypts all data transmitted between them. SSH is designed to prevent password-sniffing attacks and other malicious cyber-attacks It uses a client-server paradigm, in which clients and servers**

communicate via a secure channel. The SSH protocol has three layers: the transport layer, the authentication layer, and the connection layer

2. How do you know that you already installed the public key to the remote servers?

    - **You can attempt to establish an SSH connection to the remote server to see if you have successfully installed the public key. In the event that no password prompt appears, public key authentication is functioning 1. Verify that the public key is on the remote server in the proper position. Verify that the authorized_keys file and the.ssh directory have the proper permissions set. Check to make sure the public key is formatted correctly and free of any additional spaces or line breaks.**

Part 2: Discussion

Provide screenshots for each task.

It is assumed that you are done with the last activity (**Activity 2: SSH Key-Based Authentication**).

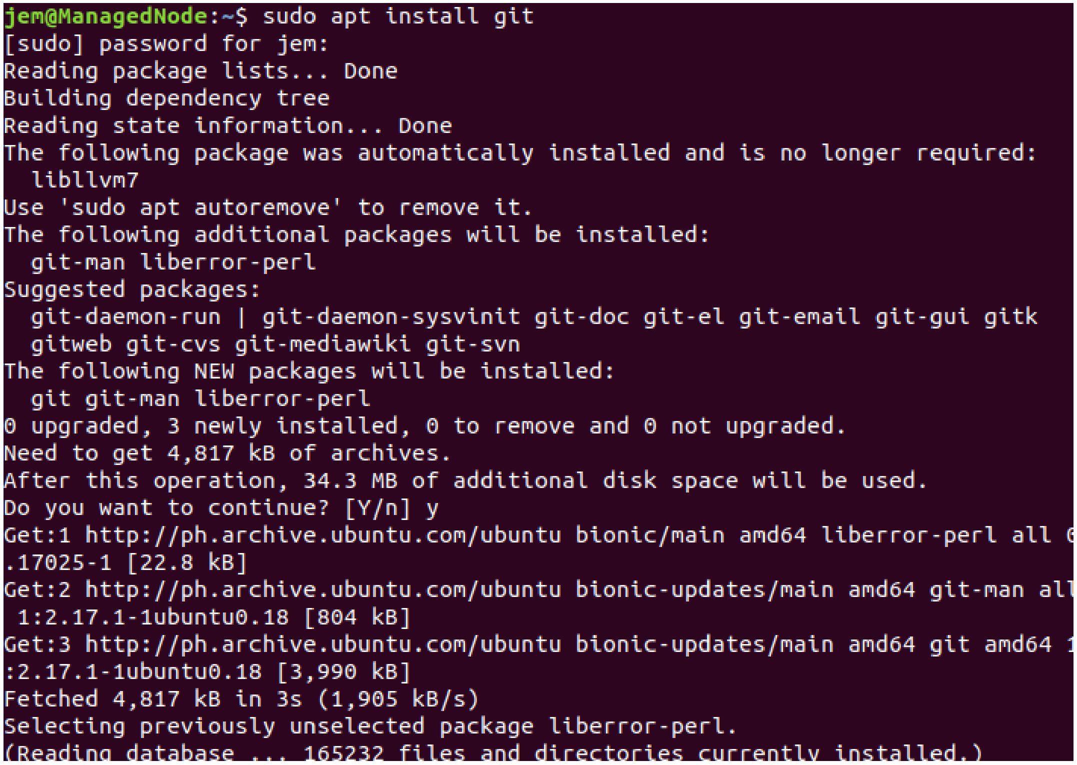
Set up Git

At the heart of GitHub is an open-source version control system (VCS) called Git. Git is responsible for everything GitHub-related that happens locally on your computer. To use Git on the command line, you'll need to download, install, and configure Git on your computer. You can also install GitHub CLI to use GitHub from the command line. If you don't need to work with files locally, GitHub lets you complete many Git-related actions directly in the browser, including:

- Creating a repository
- Forking a repository
- Managing files
- Being social

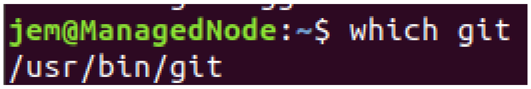
Task 3: Set up the Git Repository

1. On the local machine, verify the version of your git using the command *which git.* If a directory of git is displayed, then you don't need to install git. Otherwise, to install git, use the following command: *sudo apt install git*



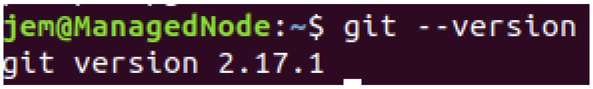
```
jem@ManagedNode:~$ sudo apt install git
[sudo] password for jem:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  libllvm7
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  git-man liberror-perl
Suggested packages:
  git-daemon-run | git-daemon-sysvinit git-doc git-el git-email git-gui gitk
  gitweb git-cvs git-mediawiki git-svn
The following NEW packages will be installed:
  git git-man liberror-perl
0 upgraded, 3 newly installed, 0 to remove and 0 not upgraded.
Need to get 4,817 kB of archives.
After this operation, 34.3 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ph.archive.ubuntu.com/ubuntu bionic/main amd64 liberror-perl all 0
.17025-1 [22.8 kB]
Get:2 http://ph.archive.ubuntu.com/ubuntu bionic-updates/main amd64 git-man all
 1:2.17.1-1ubuntu0.18 [804 kB]
Get:3 http://ph.archive.ubuntu.com/ubuntu bionic-updates/main amd64 git amd64 1
:2.17.1-1ubuntu0.18 [3,990 kB]
Fetched 4,817 kB in 3s (1,905 kB/s)
Selecting previously unselected package liberror-perl.
(Reading database ... 165232 files and directories currently installed.)
```

2. After the installation, issue the command *which git* again. The directory of git is usually installed in this location: *user/bin/git*.



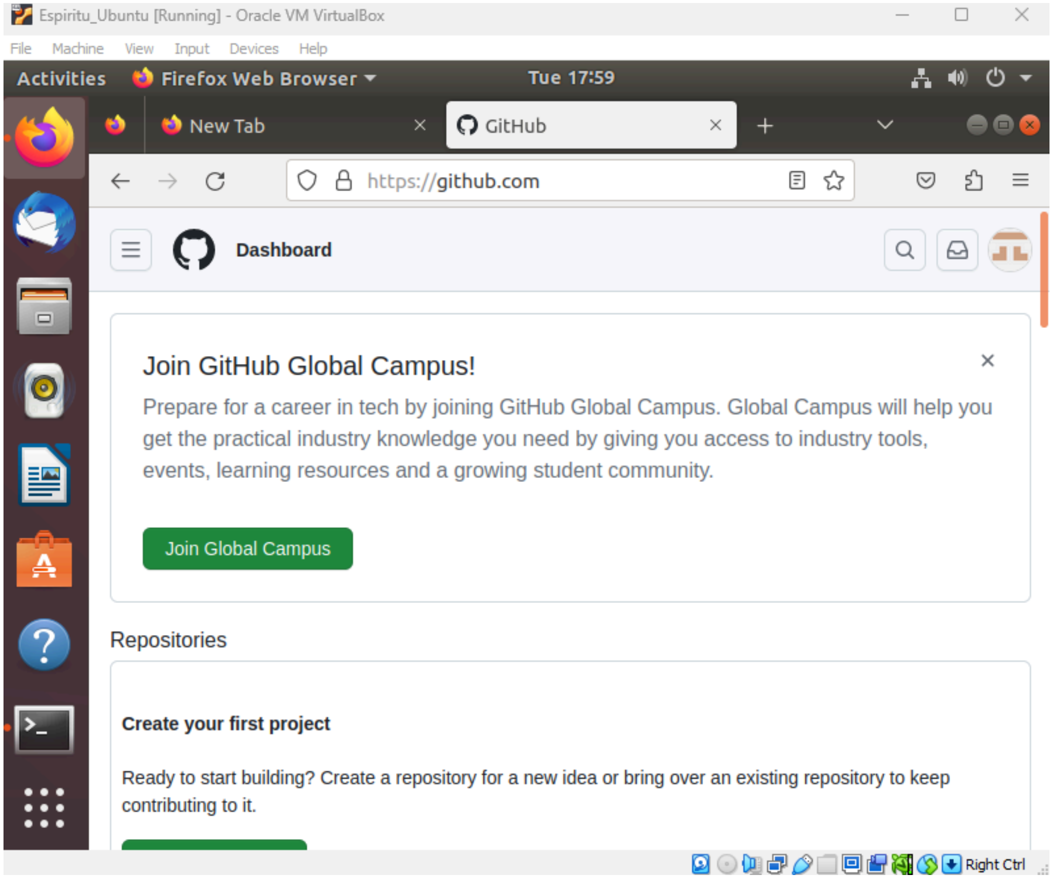
```
jem@ManagedNode:~$ which git
/usr/bin/git
```

3. The version of git installed in your device is the latest. Try issuing the command *git --version* to know the version installed.



```
jem@ManagedNode:~$ git --version
git version 2.17.1
```

4. Using the browser in the local machine, go to www.github.com.



Join GitHub Global Campus!

Prepare for a career in tech by joining GitHub Global Campus. Global Campus will help you get the practical industry knowledge you need by giving you access to industry tools, events, learning resources and a growing student community.

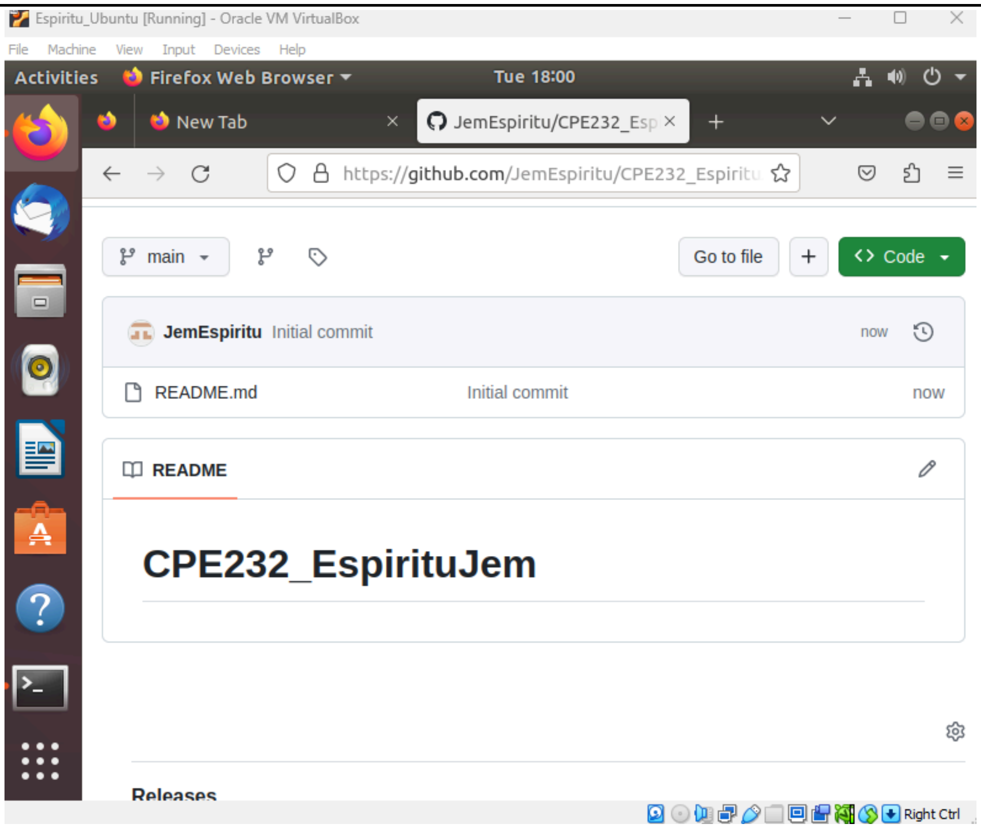
Join Global Campus

Repositories

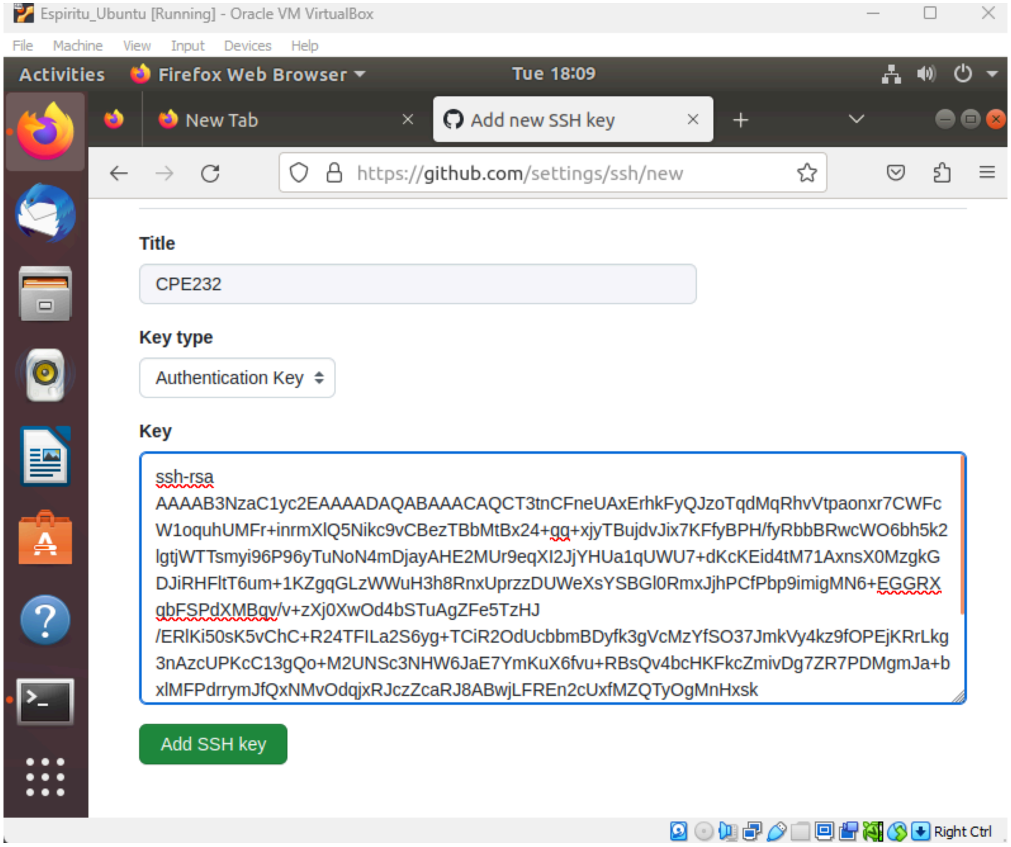
Create your first project

Ready to start building? Create a repository for a new idea or bring over an existing repository to keep contributing to it.

5. Sign up in case you don't have an account yet. Otherwise, login to your GitHub account.
    a. Create a new repository and name it as CPE232_yourname. Check Add a README file and click Create repository.



b. Create a new SSH key on GitHub. Go your profile's setting and click SSH and GPG keys. If there is an existing key, make sure to delete it. To create a new SSH keys, click New SSH Key. Write CPE232 key as the title of the key.



Title

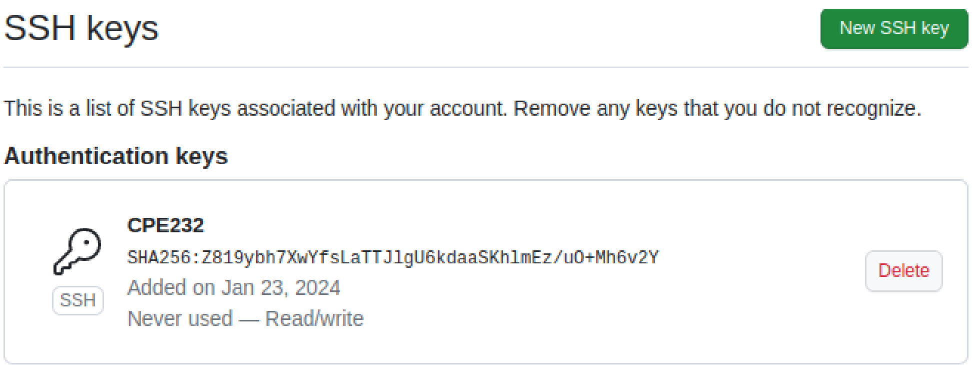
CPE232

Key type

Authentication Key

Key

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAACAQCT3tnCFneUAxErhkFyQJzoTqdMqRhvVtpaonxr7CWFc
W1oquhUMFr+inrmXlQ5Nikc9vCBezTBbMtBx24+qq+xjyTBujdvJix7KFfyBPH/fyRbbBRwcWO6bh5k2
lgtjWTTsmyi96P96yTuNoN4mDjayAHE2MUr9eqXI2JjYHUa1qUWU7+dKcKEid4tM71AxnsX0MzgkG
DJiRHFltT6um+1KZgqGLzWWuH3h8RnxUprzzDUWeXsYSBGl0RmxJjhPCfPbp9imigMN6+EGGRX
qbFSPdXMBqv/v+zXj0XwOd4bSTuAgZFe5TzHJ
/ERlKi50sK5vChC+R24TFILa2S6yg+TCiR2OdUcbbmBDyfk3gVcMzYfSO37JmkVy4kz9fOPEjKRrLkg
3nAzcUPKcC13gQo+M2UNSc3NHW6JaE7YmKuX6fvu+RBsQv4bcHKFkcZmivDg7ZR7PDMgmJa+b
xlMFPdrrymJfQxNMvOdqjxRJczZcaRJ8ABwjLFREn2cUxfMZQTyOgMnHxsk
```



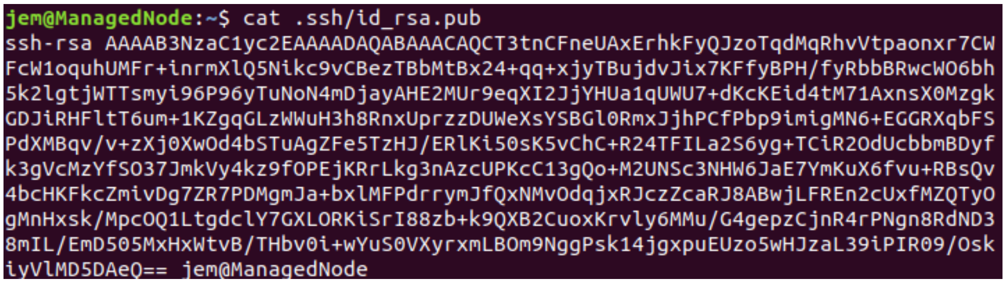
SSH keys

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

Authentication keys

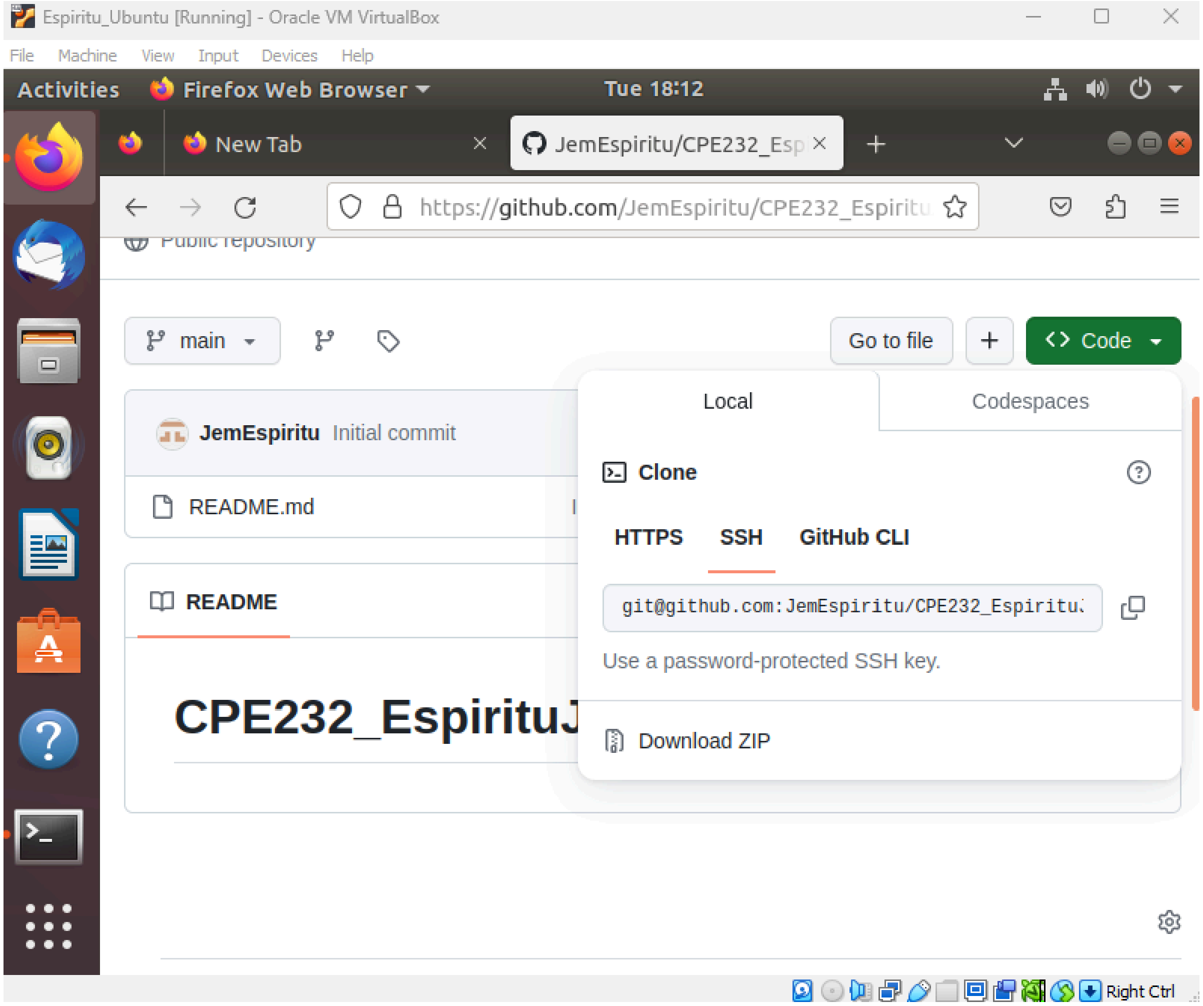
CPE232
SHA256:Z819ybh7XwYfsLaTTJ1gU6kdaaSKhlmEz/uO+Mh6v2Y
Added on Jan 23, 2024
Never used — Read/write

c. On the local machine's terminal, issue the command cat .ssh/id_rsa.pub and copy the public key. Paste it on the GitHub key and press Add SSH key.



```
jem@ManagedNode:~$ cat .ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAACAQCT3tnCFneUAxErhkFyQJzoTqdMqRhvVtpaonxr7CW
FcW1oquhUMFr+inrmXlQ5Nikc9vCBezTBbMtBx24+qq+xjyTBujdvJix7KFfyBPH/fyRbbBRwcWO6bh
5k2lgtjWTTsmyi96P96yTuNoN4mDjayAHE2MUr9eqXI2JjYHUa1qUWU7+dKcKEid4tM71AxnsX0Mzgk
GDJiRHFltT6um+1KZgqGLzWWuH3h8RnxUprzzDUWeXsYSBGl0RmxJjhPCfPbp9imigMN6+EGGRXqbFS
PdXMBqv/v+zXj0XwOd4bSTuAgZFe5TzHJ/ERlKi50sK5vChC+R24TFILa2S6yg+TCiR2OdUcbbmBDyf
k3gVcMzYfSO37JmkVy4kz9fOPEjKRrLkg3nAzcUPKcC13gQo+M2UNSc3NHW6JaE7YmKuX6fvu+RBsQv
4bcHKFkcZmivDg7ZR7PDMgmJa+bxlMFPdrrymJfQxNMvOdqjxRJczZcaRJ8ABwjLFREn2cUxfMZQTyO
gMnHxsk/MpcOQ1LtgdclY7GXLORKiSrI88zb+k9QXB2CuoxKrvly6MMu/G4gepzCjnR4rPNgn8RdND3
8mIL/EmD505MxHxWtvB/THbv0i+wYuS0VXyrxmLBOm9NggPsk14jgxpuEUzo5wHJzaL39iPIR09/Osk
iyVlMD5DAeQ== jem@ManagedNode
```

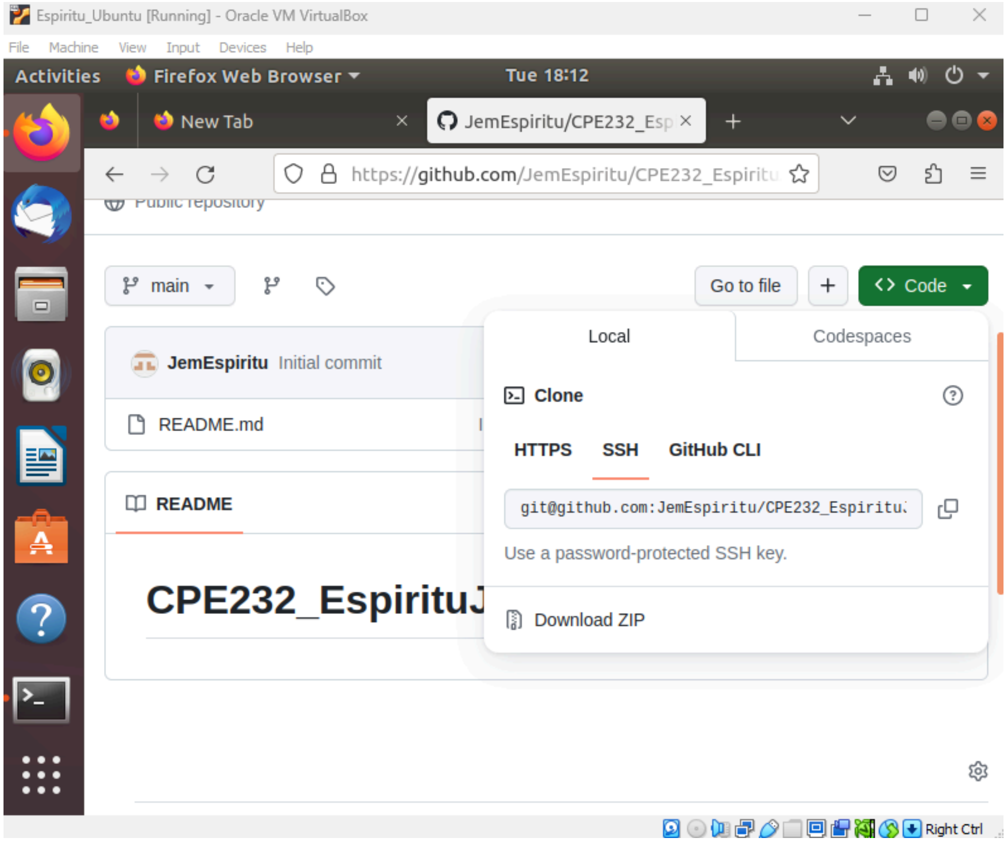
d. Clone the repository that you created. In doing this, you need to get the link from GitHub. Browse to your repository as shown below. Click on the Code drop down menu. Select SSH and copy the link.

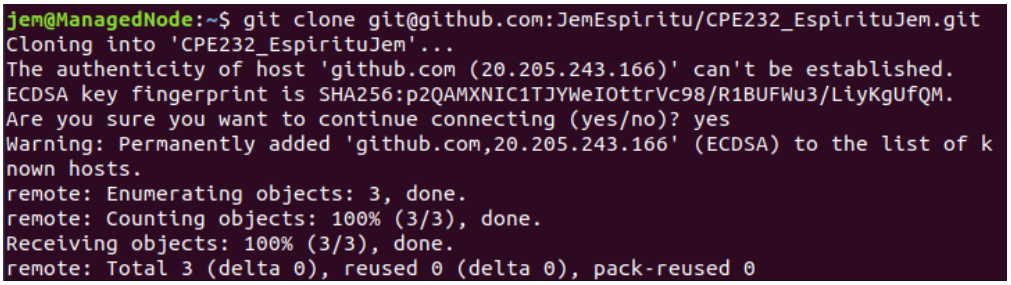


```
git@github.com:JemEspiritu/CPE232_Espiritu.
```

Use a password-protected SSH key.

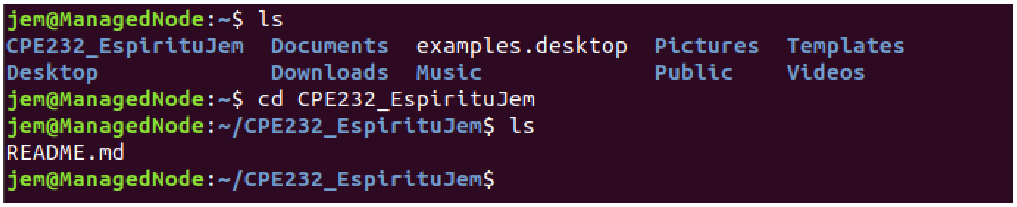
e. Issue the command git clone followed by the copied link. For example, *git clone* *git@github.com:jvtaylar-cpe/CPE232_yourname.git*. When prompted to continue connecting, type yes and press enter.


Espiritu_Ubuntu [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Firefox Web Browser
Tue 18:12
New Tab
JemEspiritu/CPE232_Esp
https://github.com/JemEspiritu/CPE232_Espiritu
Public repository
main
Go to file
+
Code
Local
Codespaces
JemEspiritu Initial commit
README.md
README
CPE232_EspirituJ
Clone
HTTPS
SSH
GitHub CLI
git@github.com:JemEspiritu/CPE232_Espiritu.
Use a password-protected SSH key.
Download ZIP
Right Ctrl



```
jem@ManagedNode:~$ git clone git@github.com:JemEspiritu/CPE232_EspirituJem.git
Cloning into 'CPE232_EspirituJem'...
The authenticity of host 'github.com (20.205.243.166)' can't be established.
ECDSA key fingerprint is SHA256:p2QAMXNIC1TJYWeIOttrVc98/R1BUFWu3/LiyKgUfQM.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'github.com,20.205.243.166' (ECDSA) to the list of k
nown hosts.
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
Receiving objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
```

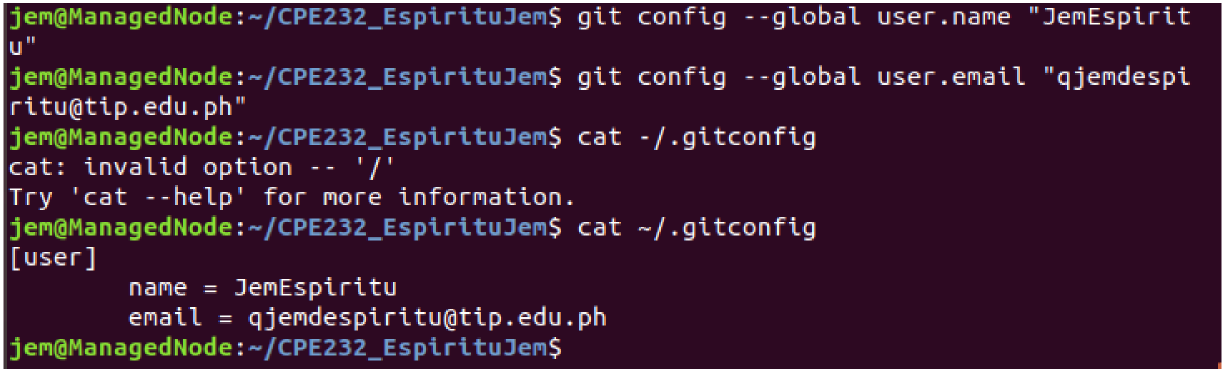
f. To verify that you have cloned the GitHub repository, issue the command *ls*. Observe that you have the CPE232_yourname in the list of your directories. Use CD command to go to that directory and LS command to see the file README.md.



```
jem@ManagedNode:~$ ls
CPE232_EspirituJem  Documents  examples.desktop  Pictures  Templates
Desktop             Downloads  Music             Public    Videos
jem@ManagedNode:~$ cd CPE232_EspirituJem
jem@ManagedNode:~/CPE232_EspirituJem$ ls
README.md
jem@ManagedNode:~/CPE232_EspirituJem$
```

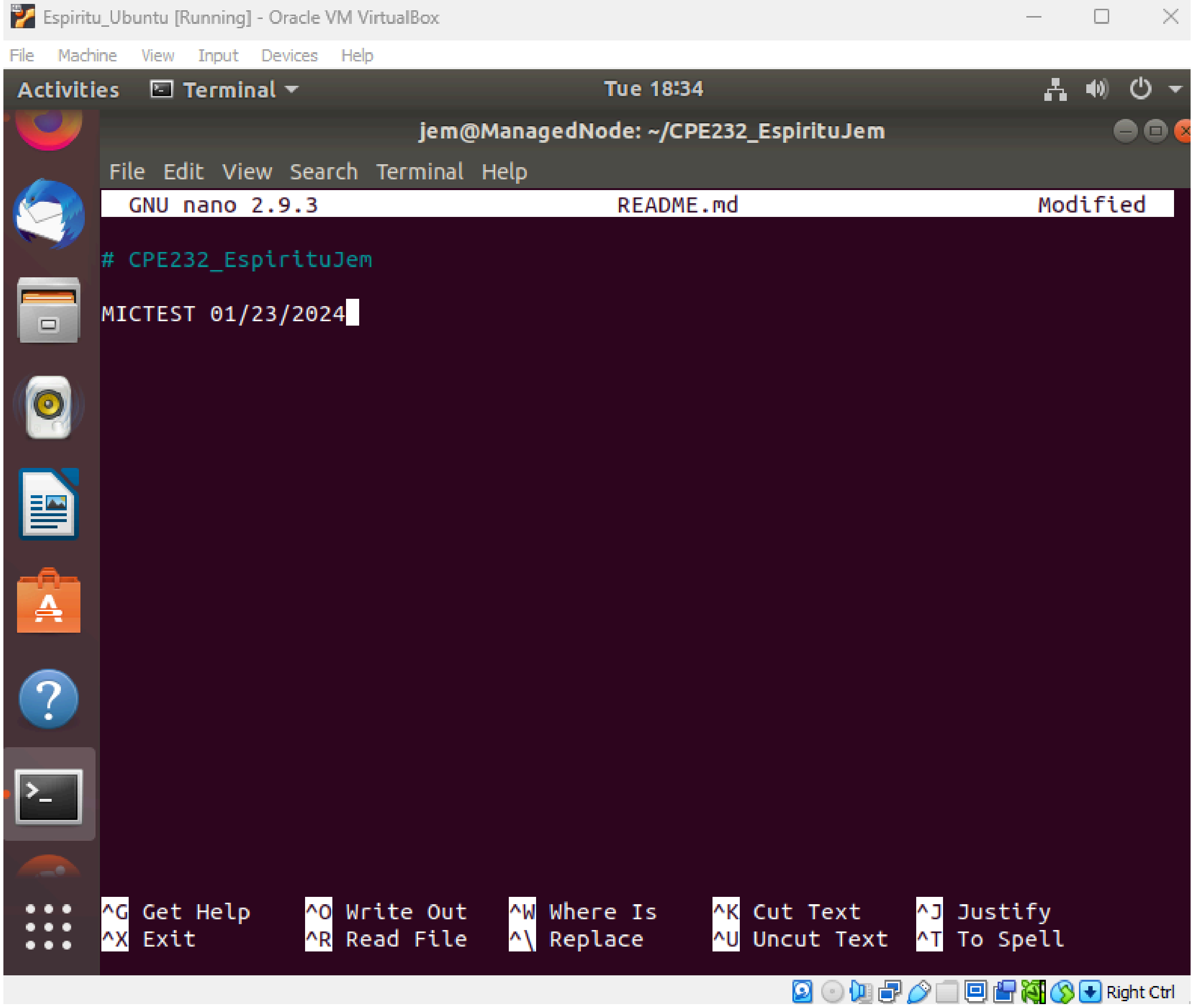
g. Use the following commands to personalize your git.

- *git config --global user.name "Your Name"*
- *git config --global user.email yourname@email.com*
- Verify that you have personalized the config file using the command *cat ~/.gitconfig*



```
jem@ManagedNode:~/CPE232_EspirituJem$ git config --global user.name "JemEspirit
u"
jem@ManagedNode:~/CPE232_EspirituJem$ git config --global user.email "qjemdespi
ritu@tip.edu.ph"
jem@ManagedNode:~/CPE232_EspirituJem$ cat -/.gitconfig
cat: invalid option -- '/'
Try 'cat --help' for more information.
jem@ManagedNode:~/CPE232_EspirituJem$ cat ~/.gitconfig
[user]
        name = JemEspiritu
        email = qjemdespiritu@tip.edu.ph
jem@ManagedNode:~/CPE232_EspirituJem$
```

h. Edit the README.md file using nano command. Provide any information on the markdown file pertaining to the repository you created. Make sure to write out or save the file and exit.

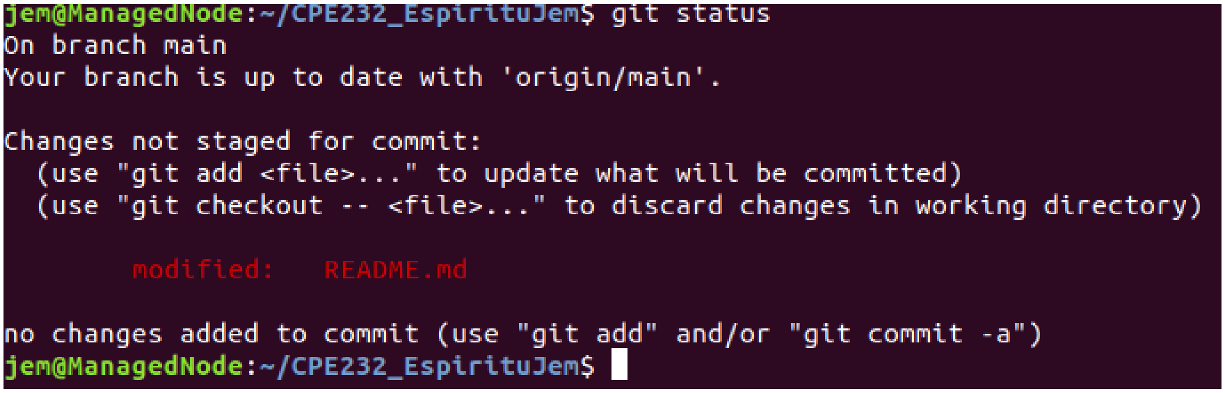


Espiritu_Ubuntu [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Terminal
Tue 18:34
jem@ManagedNode: ~/CPE232_EspirituJem
File Edit View Search Terminal Help
Right Ctrl

```
  GNU nano 2.9.3                    README.md                          Modified

# CPE232_EspirituJem

MICTEST 01/23/2024





^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify
^X Exit        ^R Read File   ^\ Replace     ^U Uncut Text  ^T To Spell
```

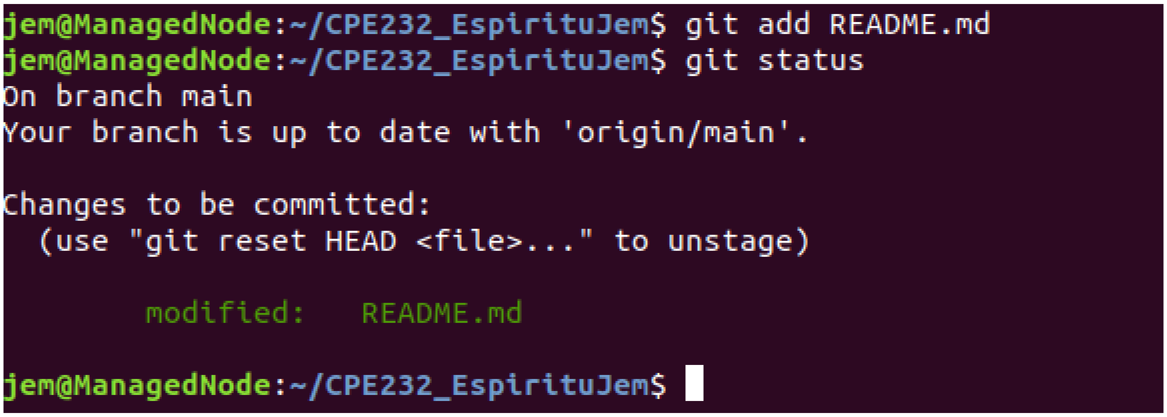
i. Use the *git status* command to display the state of the working directory and the staging area. This command shows which changes have been staged, which haven't, and which files aren't being tracked by Git.

Status output does not show any information regarding the committed project history. What is the result of issuing this command?



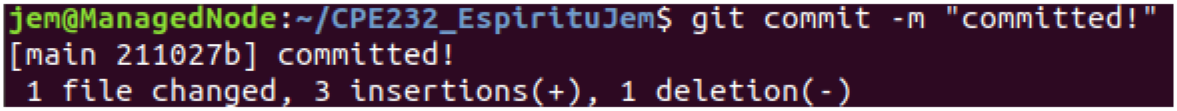
```
jem@ManagedNode:~/CPE232_EspirituJem$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")
jem@ManagedNode:~/CPE232_EspirituJem$ 
```

j. Use the command *git add README.md* to add the file into the staging area.



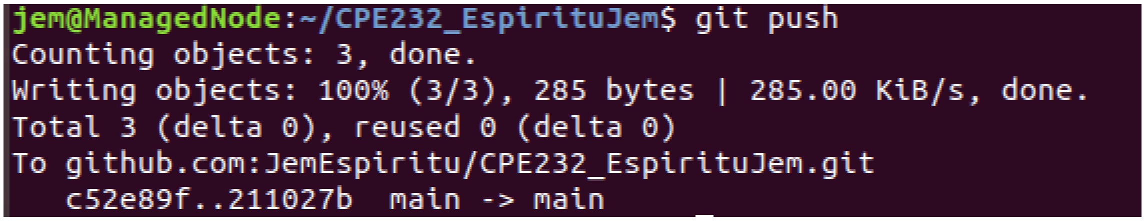
```
jem@ManagedNode:~/CPE232_EspirituJem$ git add README.md
jem@ManagedNode:~/CPE232_EspirituJem$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   README.md

jem@ManagedNode:~/CPE232_EspirituJem$ 
```

k. Use the *git commit -m "your message"* to create a snapshot of the staged changes along the timeline of the Git projects history. The use of this command is required to select the changes that will be staged for the next commit.



```
jem@ManagedNode:~/CPE232_EspirituJem$ git commit -m "committed!"
[main 211027b] committed!
 1 file changed, 3 insertions(+), 1 deletion(-)
```

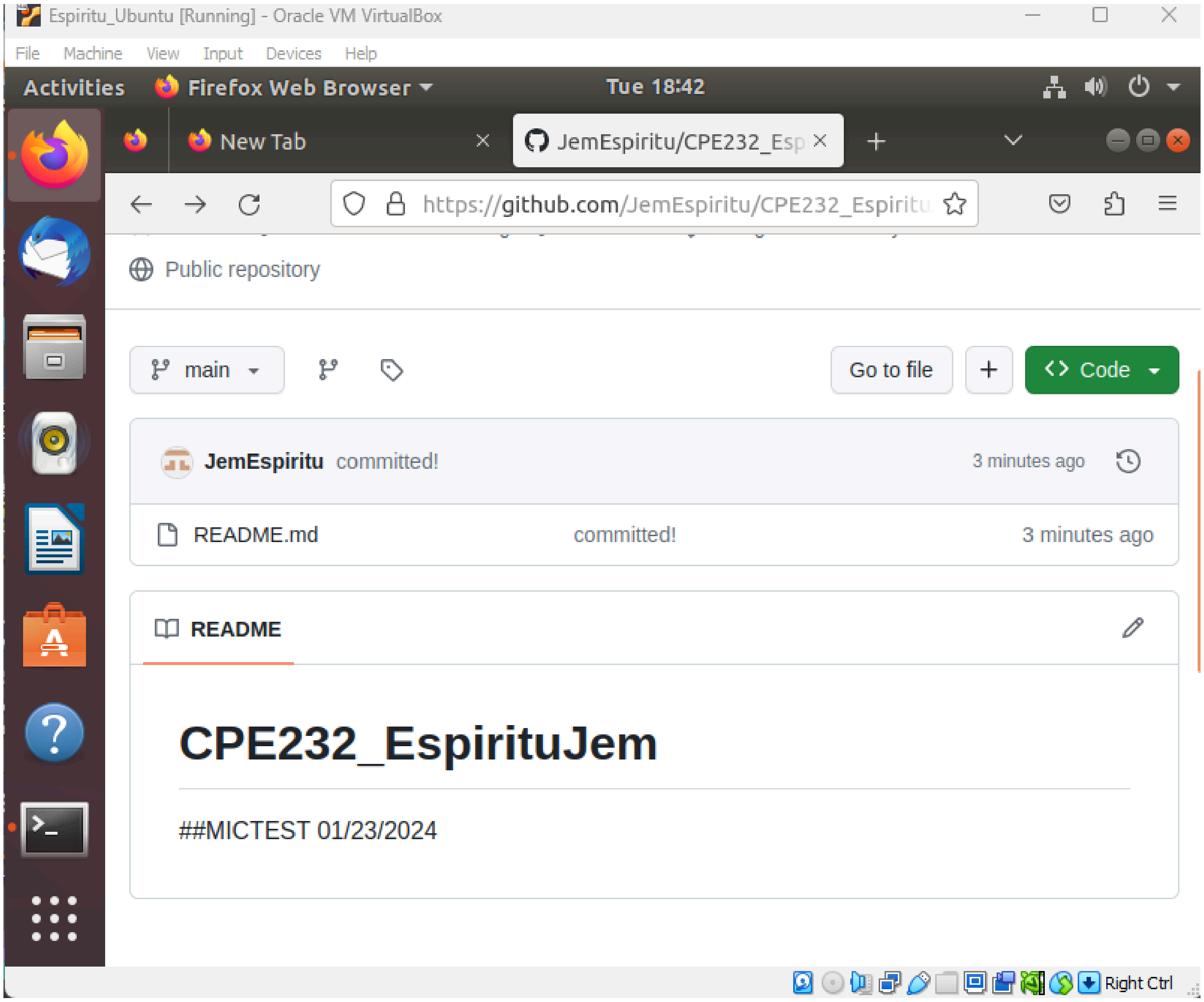
l. Use the command *git push <remote><branch>* to upload the local repository content to GitHub repository. Pushing means to transfer commits from the local repository to the remote repository. As an example, you may issue *git push origin main.*



```
jem@ManagedNode:~/CPE232_EspirituJem$ git push
Counting objects: 3, done.
Writing objects: 100% (3/3), 285 bytes | 285.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To github.com:JemEspiritu/CPE232_EspirituJem.git
   c52e89f..211027b  main -> main
```

m. On the GitHub repository, verify that the changes have been made to README.md by refreshing the page. Describe the README.md file.

You can notice the how long was the last commit. It should be some minutes ago and the message you typed on the git commit command should be there. Also, the README.md file should have been edited according to the text you wrote.


Espiritu_Ubuntu [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Firefox Web Browser
Tue 18:42
New Tab
JemEspiritu/CPE232_Esp
https://github.com/JemEspiritu/CPE232_Espiritu
Public repository
main
Go to file
+
Code
Right Ctrl

JemEspiritu committed! 3 minutes ago

README.md committed! 3 minutes ago

README

CPE232_EspirituJem

##MICTEST 01/23/2024

Reflections:

Answer the following:

3. What sort of things have we so far done to the remote servers using ansible commands?
    - In the discussion, we established a pair of SSH keys for user authentication. I was able to transfer the public key to the two distant servers, ControlNode1 and ControlNode2, and save data using this encryption key.

      and after making changes to the repository, we linked the local server to Github.

4. How important is the inventory file?

- Popular automation tool Ansible uses it to handle the hosts it controls 1. Ansible can connect to and maintain a list of hosts and groups of hosts in its inventory file, which is a text file. Nevertheless, the inventory file can also be used to define the SSH key to use for authentication when connecting to host 1. The inventory file's ansible_ssh_private_key_file variable is updated with the path to the SSH key file to accomplish this.

Conclusions/Learnings:

- **In this Activity I learn how to Use the terminal to type ls -al ~/.ssh to see if you already have an SSH key pair generated for your system in order to set up SSH key-based authentication for Git Check. Run mkdir $HOME/.ssh and use ssh-keygen -t rsa -b 4096 -C your@email.com to generate a new set of keys if you don't see any output or if that directory doesn't exist. and selecting the "New SSH key" button on the GitHub settings page. After giving your key a memorable name, enter your public key.**