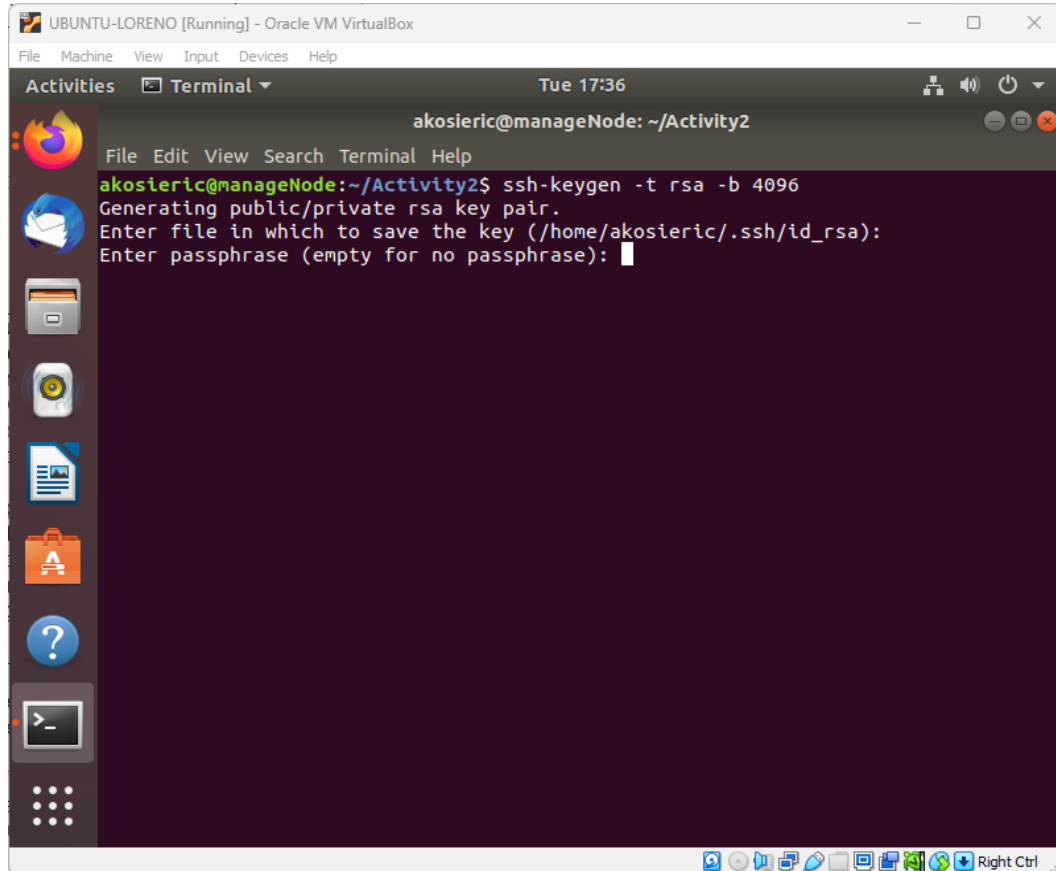


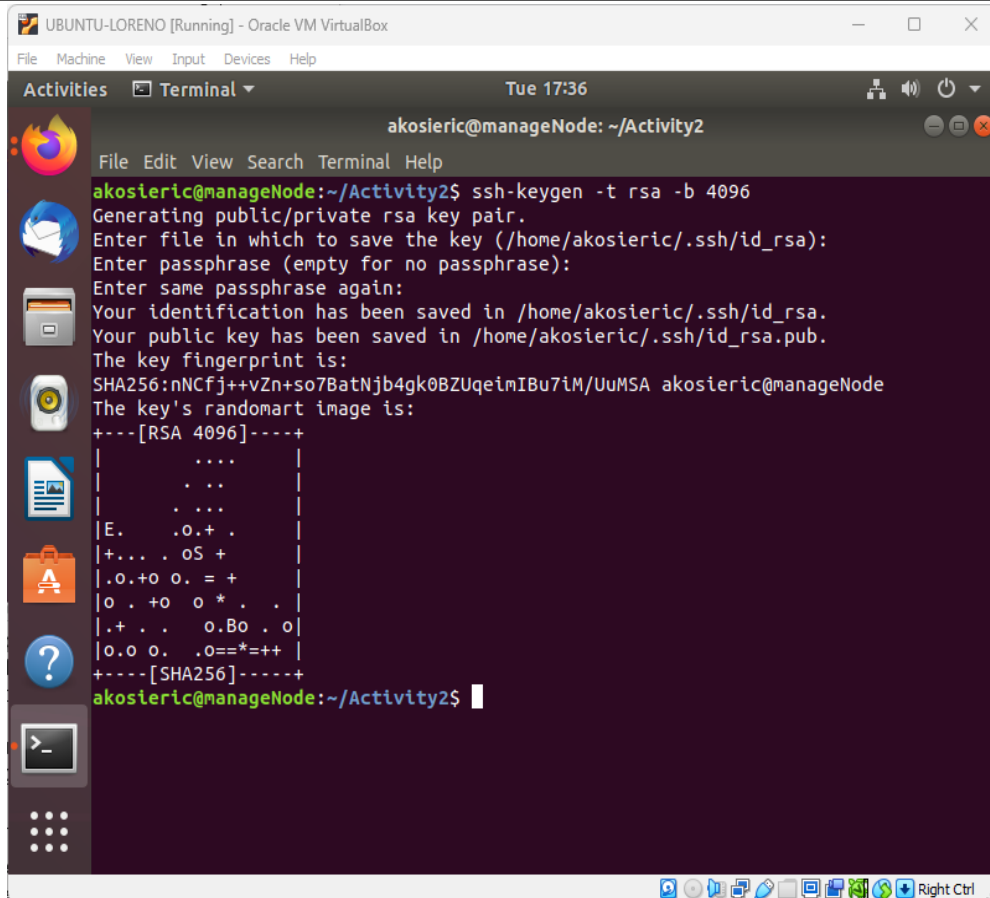
<b>Name: Loreno, Eric H.</b>	<b>Date Performed: 22/08/2023</b>
<b>Course/Section: CPE31S4</b>	<b>Date Submitted: 22/08/2023</b>
<b>Instructor: Dr. Jonathan V. Tylar</b>	<b>Semester and SY: 1st SEM 2023-2024</b>
<b>Activity 2: SSH Key-Based Authentication and Setting up Git</b>	
<p><b>1. Objectives:</b></p> <ul style="list-style-type: none"> <li>1.1 Configure remote and local machine to connect via SSH using a KEY instead of using a password</li> <li>1.2 Create a public key and private key</li> <li>1.3 Verify connectivity</li> <li>1.4 Setup Git Repository using local and remote repositories</li> <li>1.5 Configure and Run ad hoc commands from local machine to remote servers</li> </ul>	
<p><b>Part 1: Discussion</b></p> <p>It is assumed that you are already done with the last Activity (<b>Activity 1: Configure Network using Virtual Machines</b>). <i>Provide screenshots for each task.</i></p> <p>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.</p> <p><b>What is ssh-keygen?</b></p> <p>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.</p> <p><b>SSH Keys and Public Key Authentication</b></p> <p>The SSH protocol uses public key cryptography for authenticating hosts and users. The authentication keys, called SSH keys, are created using the keygen program.</p> <p>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.</p> <p>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.</p>	
<p><b>Task 1: Create an SSH Key Pair for User Authentication</b></p> <ul style="list-style-type: none"> <li>1. The simplest way to generate a key pair is to run <i>ssh-keygen</i> without arguments. In this case, it will prompt for the file in which to store keys. First,</li> </ul>	

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`.

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.



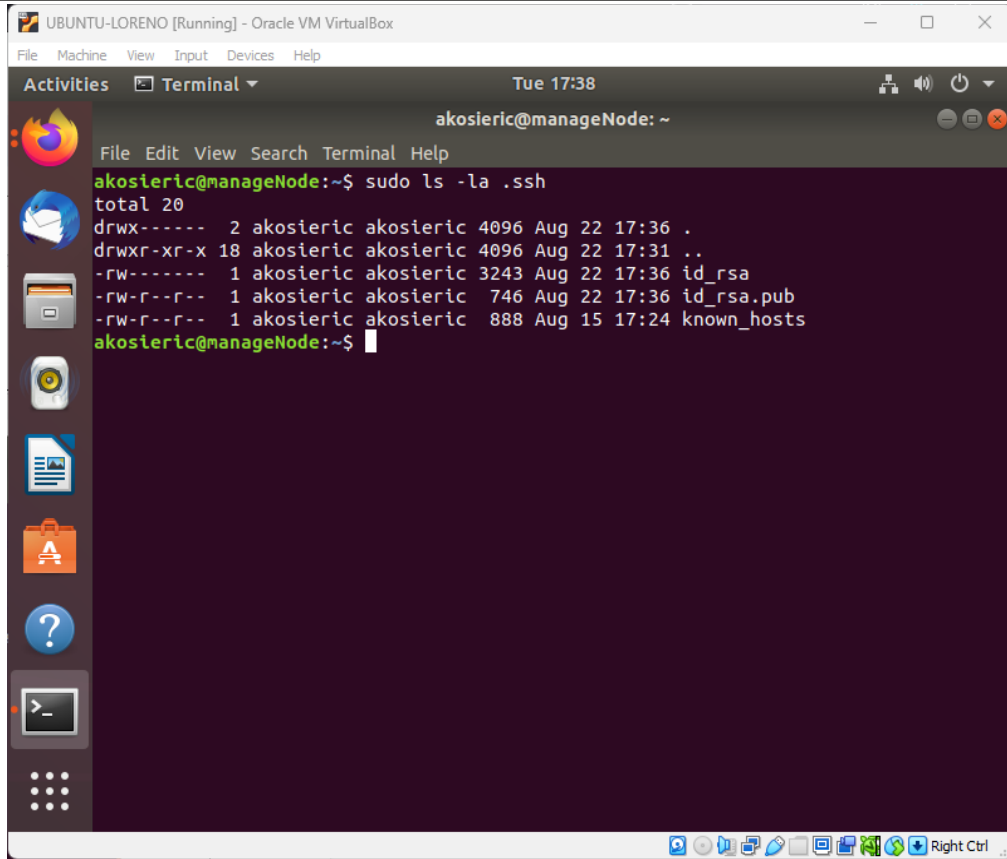
3. When asked for a passphrase, just press enter. The passphrase is used for encrypting the key, so that it cannot be used even if someone obtains the private key file. The passphrase should be cryptographically strong.



The screenshot shows a terminal window titled "akosieric@manageNode: ~/Activity2". The user has executed the command `ssh-keygen -t rsa -b 4096`. The terminal output shows the process of generating a public/private RSA key pair, saving the files to `/home/akosieric/.ssh/id_rsa` and `/home/akosieric/.ssh/id_rsa.pub`, and displaying the key's fingerprint and randomart image. The randomart image is a colorful ASCII art representation of the key. The terminal window is part of an Ubuntu-Loreno virtual machine running in Oracle VM VirtualBox.

```
akosieric@manageNode:~/Activity2$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/akosieric/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/akosieric/.ssh/id_rsa.
Your public key has been saved in /home/akosieric/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:nNcfj++vZn+so7BatNjb4gk0BZUqeimIBu7iM/UuMSA akosieric@manageNode
The key's randomart image is:
+---[RSA 4096]-----+
|
| ..
| ..
| ..
|E. .o.+ .
|+... .oS +
|.o.+o o. = +
|o . +o o * . .
|. + . . o.Bo . o
|o.o o. .o==*++
+---[SHA256]-----+
akosieric@manageNode:~/Activity2$
```

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`.



The screenshot shows a terminal window titled 'UBUNTU-LORENO [Running] - Oracle VM VirtualBox'. The terminal is running as user 'akosieric' on host 'manageNode'. The command 'sudo ls -la .ssh' has been executed, showing the following output:

```
akosieric@manageNode:~$ sudo ls -la .ssh
total 20
drwx----- 2 akosieric akosieric 4096 Aug 22 17:36 .
drwxr-xr-x 18 akosieric akosieric 4096 Aug 22 17:31 ..
-rw----- 1 akosieric akosieric 3243 Aug 22 17:36 id_rsa
-rw-r--r-- 1 akosieric akosieric 746 Aug 22 17:36 id_rsa.pub
-rw-r--r-- 1 akosieric akosieric 888 Aug 15 17:24 known_hosts
```

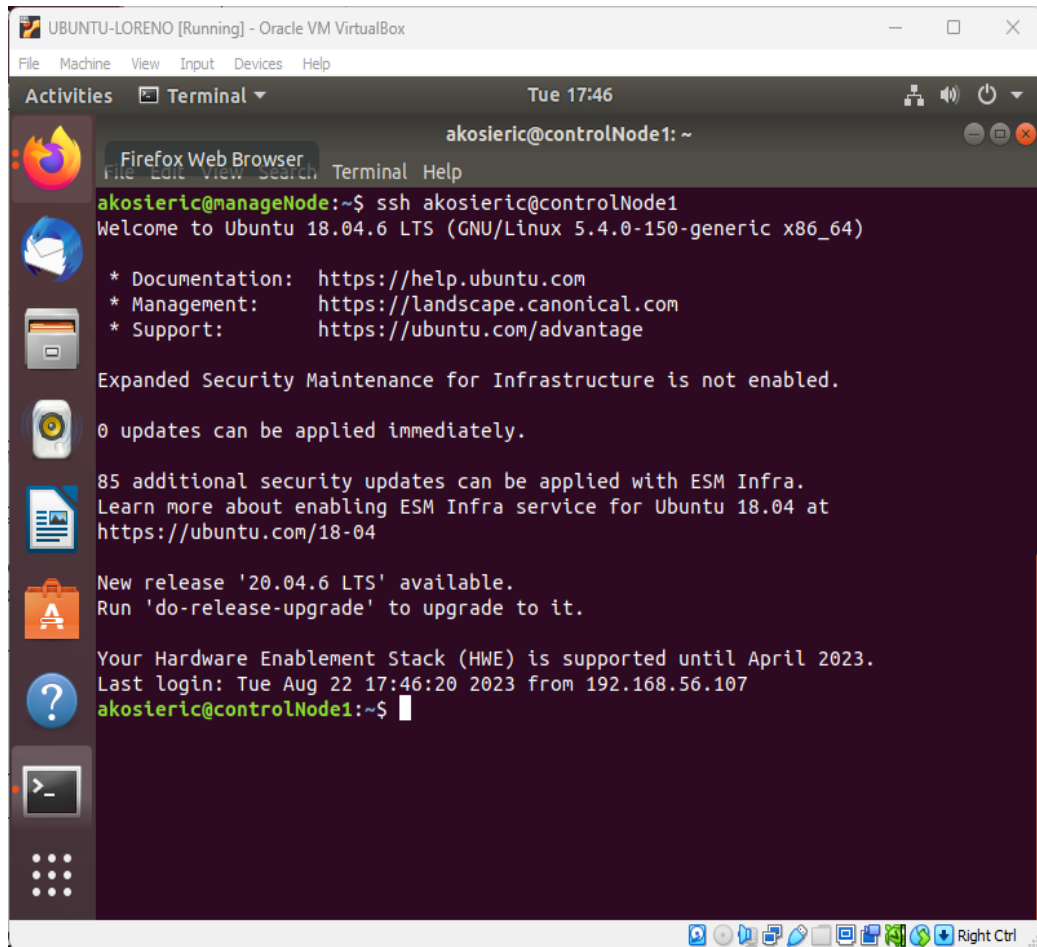
## Task 2: Copying the Public Key to the remote servers

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

```
UBUNTU-LORENO [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Tue 17:44
akosieric@manageNode: ~
File Edit View Search Terminal Help
akosieric@manageNode:~$ ssh-copy-id -i ~/.ssh/id_rsa akosieric@controlNode1
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/akosieric/
.ssh/id_rsa.pub"
/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
akosieric@controlNode1's password:
Number of key(s) added: 1
Now try logging into the machine, with: "ssh 'akosieric@controlNode1'"
and check to make sure that only the key(s) you wanted were added.
akosieric@manageNode:~$
```

```
UBUNTU-LORENO [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Tue 17:44
akosieric@manageNode: ~
File Edit View Search Terminal Help
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/akosieric/
.ssh/id_rsa.pub"
/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
akosieric@controlNode1's password:
Number of key(s) added: 1
Now try logging into the machine, with: "ssh 'akosieric@controlNode1'"
and check to make sure that only the key(s) you wanted were added.
akosieric@manageNode:~$ ssh-copy-id -i ~/.ssh/id_rsa akosieric@controlNode2
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/akosieric/
.ssh/id_rsa.pub"
/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
akosieric@controlNode2's password:
Number of key(s) added: 1
Now try logging into the machine, with: "ssh 'akosieric@controlNode2'"
and check to make sure that only the key(s) you wanted were added.
akosieric@manageNode:~$
```

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.
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?



The screenshot shows a terminal window titled "akosieric@controlNode1: ~" with a menu bar (File, Edit, View, Search, Terminal, Help) and a status bar (Tue 17:46). The terminal displays the output of an SSH command executed from a local machine (akosieric@manageNode). The output includes a welcome message for Ubuntu 18.04.6 LTS, system information, and security updates. The terminal also shows a sidebar with application icons (Firefox, Mail, Files, etc.) and a bottom status bar with system icons and a "Right Ctrl" label.

```
akosieric@manageNode:~$ ssh akosieric@controlNode1
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 5.4.0-150-generic x86_64)

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

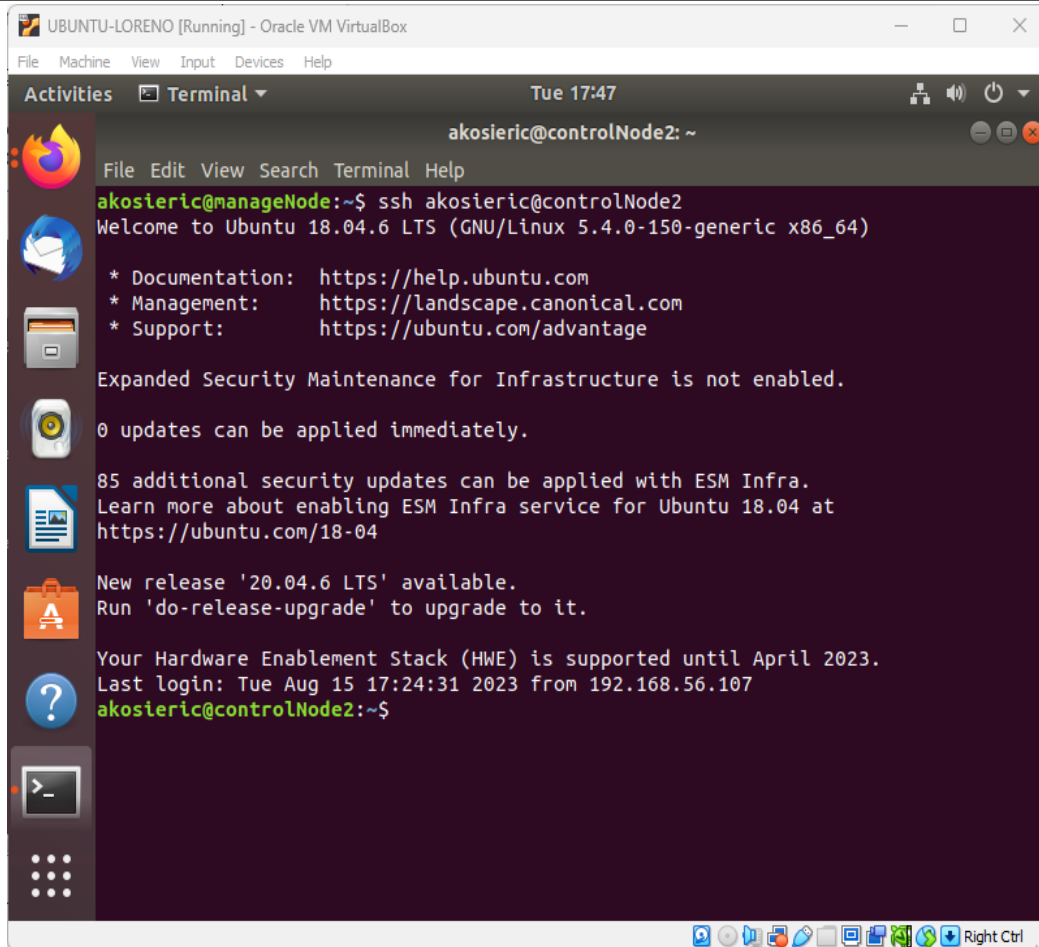
Expanded Security Maintenance for Infrastructure is not enabled.

0 updates can be applied immediately.

85 additional security updates can be applied with ESM Infra.
Learn more about enabling ESM Infra service for Ubuntu 18.04 at
https://ubuntu.com/18-04

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.
Last login: Tue Aug 22 17:46:20 2023 from 192.168.56.107
akosieric@controlNode1:~$
```



The screenshot shows a terminal window titled 'UBUNTU-LORENO [Running] - Oracle VM VirtualBox'. The terminal is running on a host named 'akosieric@controlNode2'. The user has executed the command 'ssh akosieric@controlNode2'. The terminal output shows the Ubuntu 18.04.6 LTS login banner, including documentation, management, and support links. It also displays security updates and a new release '20.04.6 LTS' available. The last login was on Tue Aug 15 17:24:31 2023 from 192.168.56.107. The prompt is 'akosieric@controlNode2:~\$'.

*It did not ask for a password because if you successfully copied your public key to the server's `authorized_keys` file and you're not prompted for a password when connecting, it means your SSH key authentication is working as intended. This is a secure and convenient way to authenticate to remote servers.*

### Reflections:

Answer the following:

1. How will you describe the ssh-program? What does it do?

- ***The SSH program is a powerful network protocol and software tool used for secure remote access and communication between computers. It establishes encrypted connections, allowing users to securely log into remote servers, execute commands, and transfer files. SSH ensures data confidentiality and integrity, making it a vital tool for system administrators. It also offers features like authentication via key pairs, secure tunnels for port forwarding, and encrypted file transfer, enhancing both security and convenience in remote computing.***

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

- *To confirm the successful installation of your public key on a remote server, attempt an SSH connection. If you're able to log in without being prompted for a password, your public key is correctly installed. You can also check the `~/.ssh/authorized_keys` file on the server to ensure your public key is present. Using the `-v` flag with the SSH command provides verbose output about the authentication process, helping you identify any issues.*

## 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

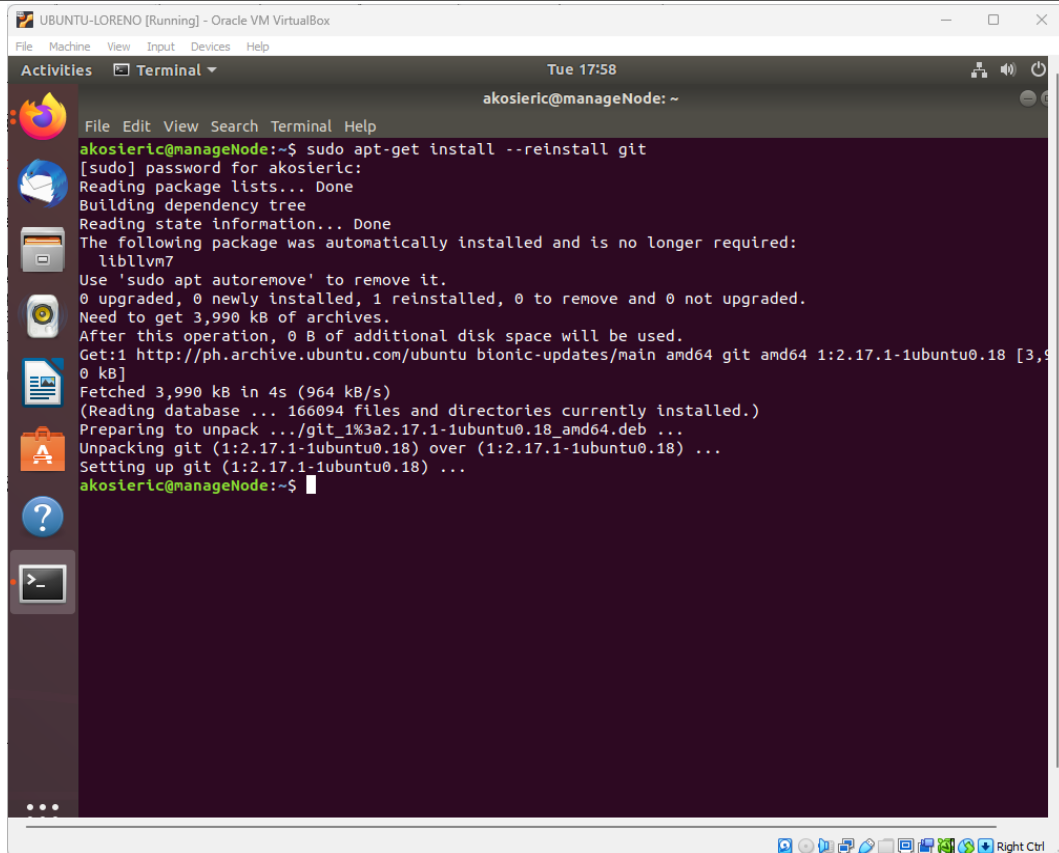
At the heart of GitHub is an open-source version control system (VCS) called Git. Git is responsible for everything GitHub-related that happens locally on your computer. To use Git on the command line, you'll need to download, install, and configure Git on your computer. You can also install GitHub CLI 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*

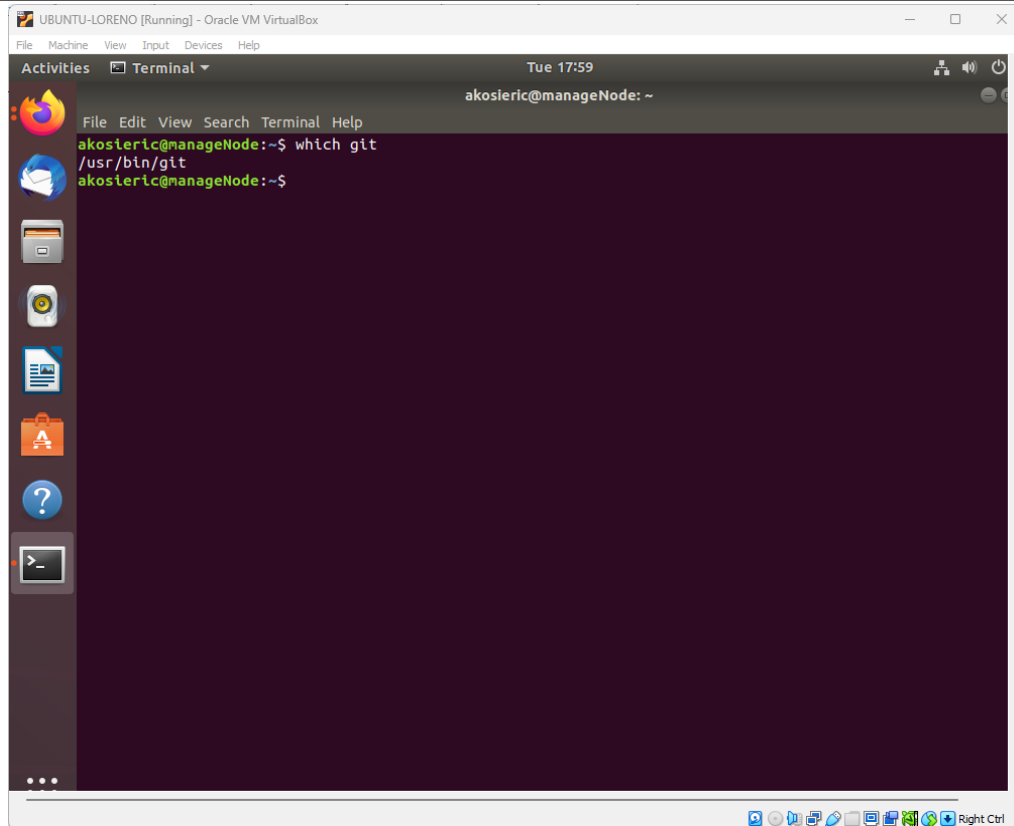




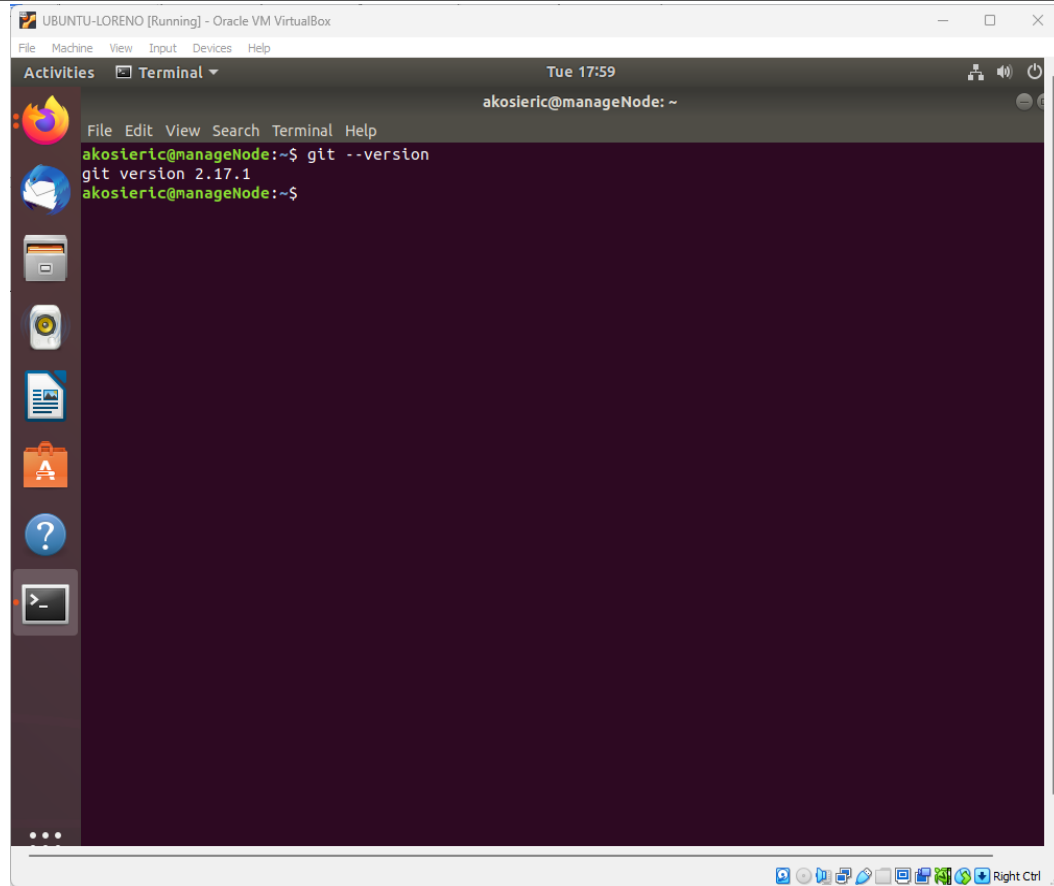
The screenshot shows a terminal window titled "UBUNTU-LORENO [Running] - Oracle VM VirtualBox". The terminal is running as the user "akosieric" on the host "manageNode". The user has executed the command "sudo apt-get install --reinstall git". The terminal output shows the following steps:

```
akosieric@manageNode:~$ sudo apt-get install --reinstall git
[sudo] password for akosieric:
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.
0 upgraded, 0 newly installed, 1 reinstalled, 0 to remove and 0 not upgraded.
Need to get 3,990 kB of archives.
After this operation, 0 B of additional disk space will be used.
Get:1 http://ph.archive.ubuntu.com/ubuntu bionic-updates/main amd64 git amd64 1:2.17.1-1ubuntu0.18 [3,990 kB]
Fetched 3,990 kB in 4s (964 kB/s)
(Reading database ... 166094 files and directories currently installed.)
Preparing to unpack .../git_1%3a2.17.1-1ubuntu0.18_amd64.deb ...
Unpacking git (1:2.17.1-1ubuntu0.18) over (1:2.17.1-1ubuntu0.18) ...
Setting up git (1:2.17.1-1ubuntu0.18) ...
akosieric@manageNode:~$
```

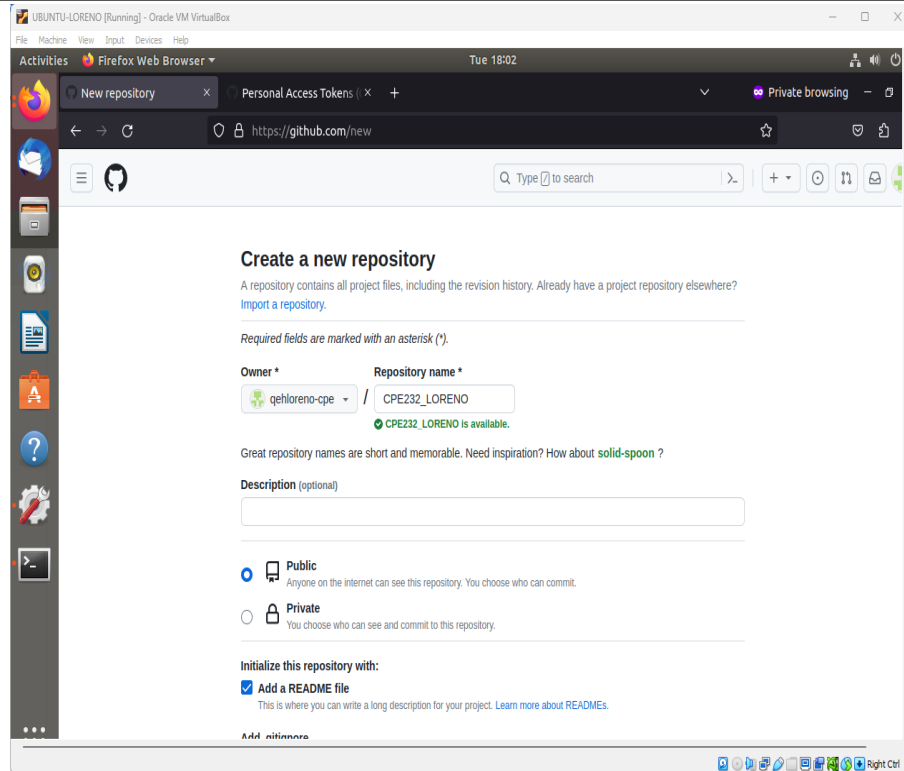
2. After the installation, issue the command *which git* again. The directory of git is usually installed in this location: *user/bin/git*.



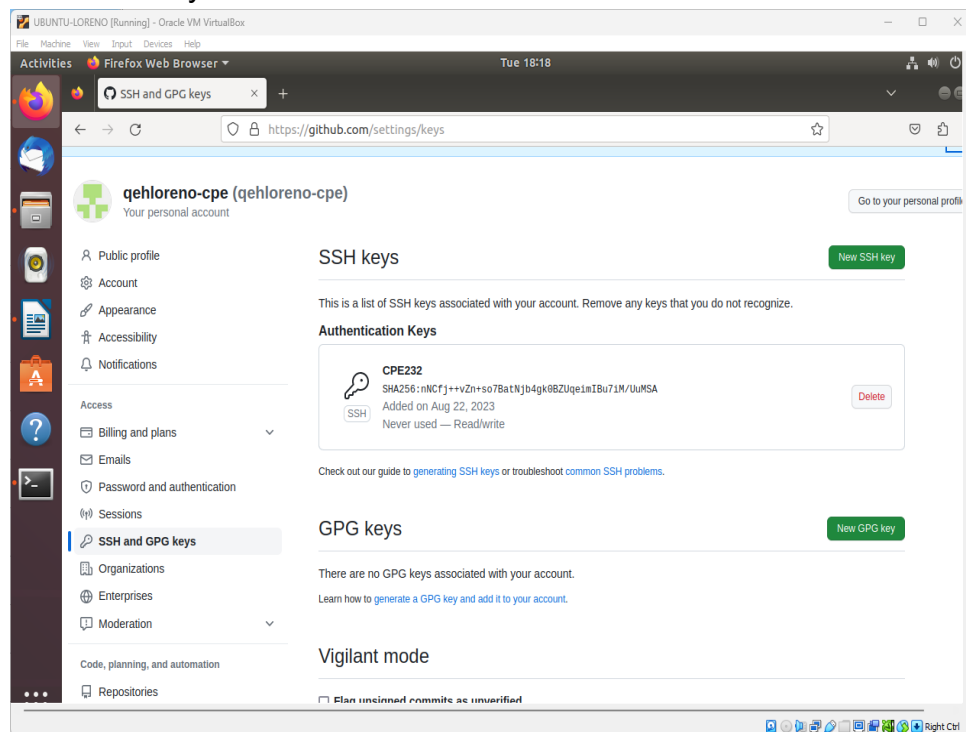
3. The version of git installed in your device is the latest. Try issuing the command *git --version* to know the version installed.



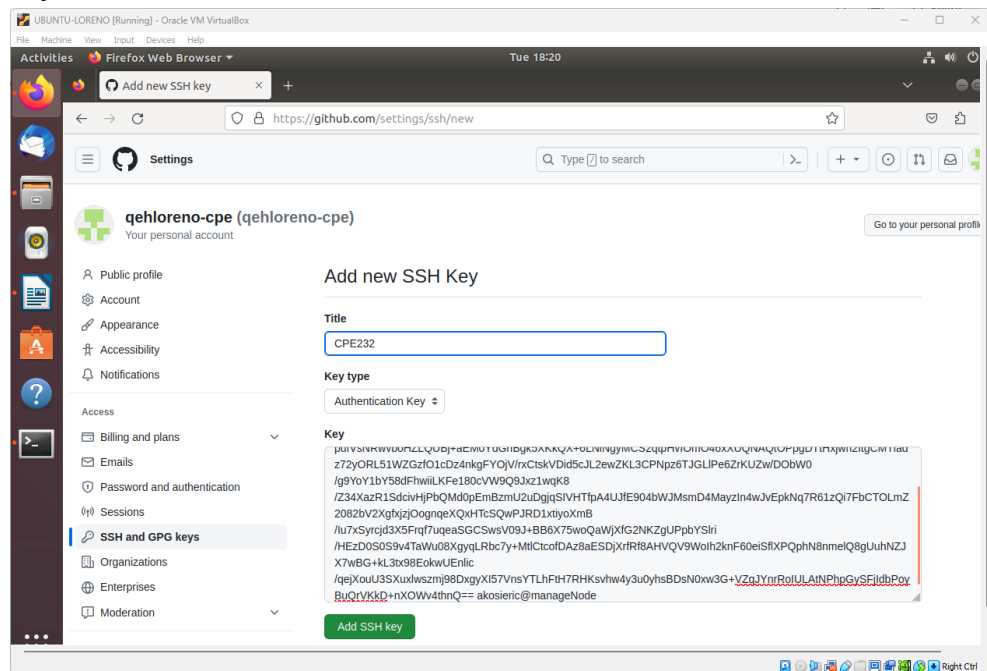
4. Using the browser in the local machine, go to [www.github.com](https://www.github.com).
5. Sign up in case you don't have an account yet. Otherwise, login to your GitHub account.
  - a. Create a new repository and name it as CPE232\_yourname. Check Add a README file and click Create repository.



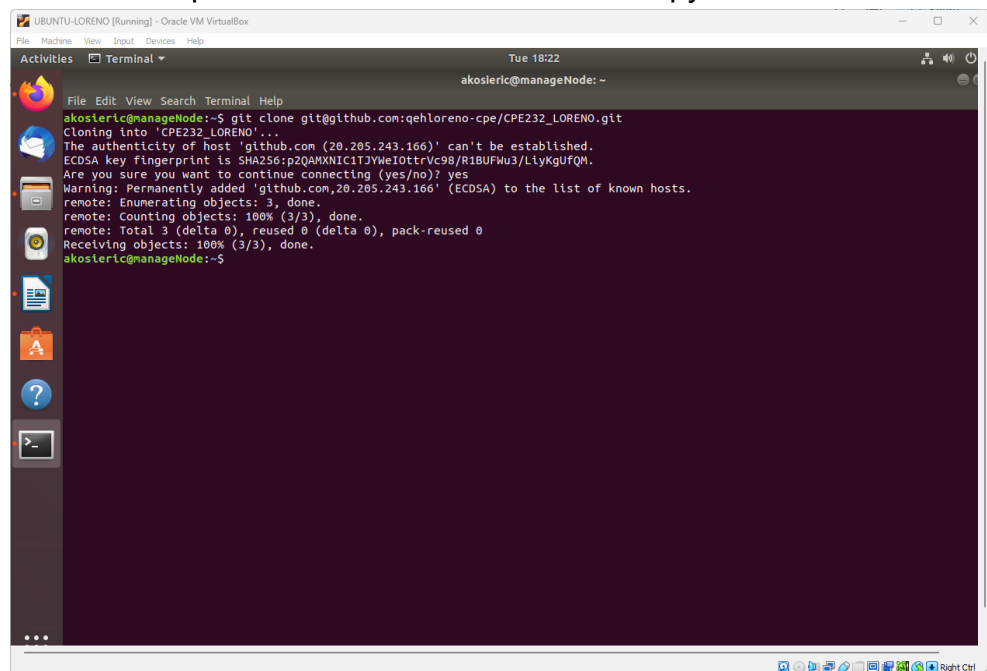
- b. Create a new SSH key on GitHub. Go your profile's setting and click SSH and GPG keys. If there is an existing key, make sure to delete it. To create a new SSH keys, click New SSH Key. Write CPE232 key as the title of the key.

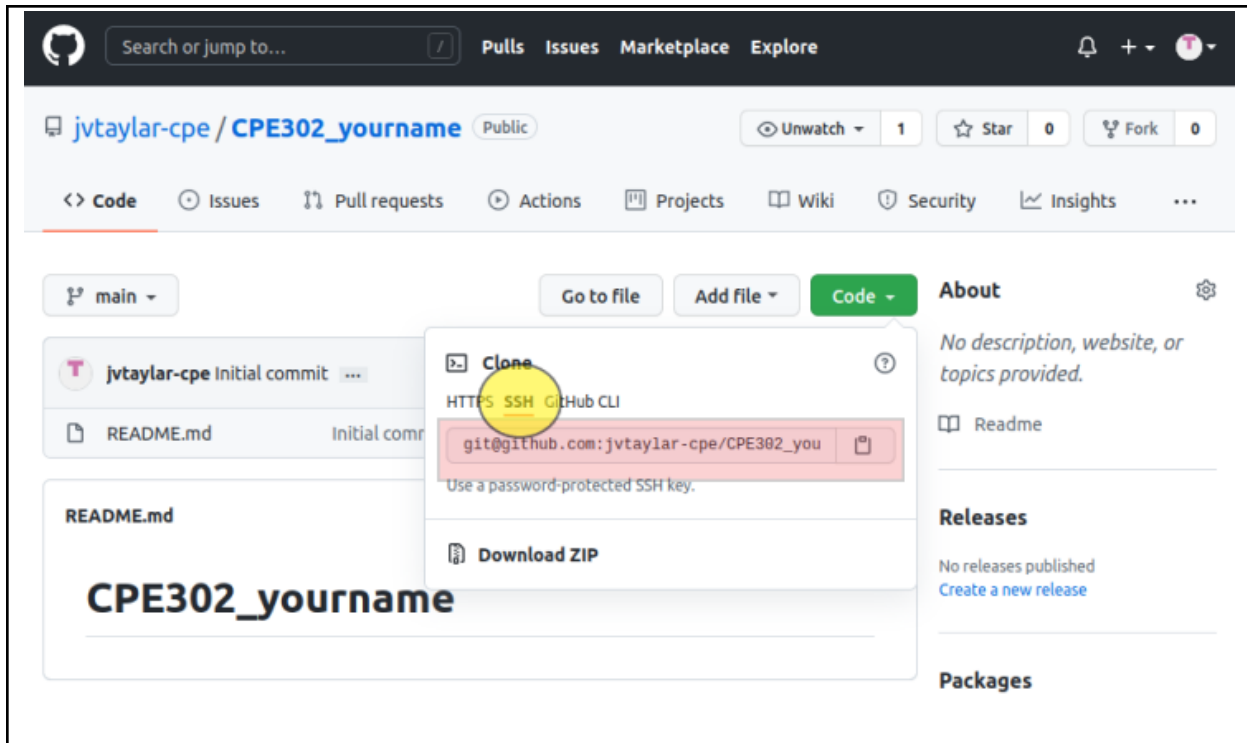


- c. On the local machine's terminal, issue the command `cat .ssh/id_rsa.pub` and copy the public key. Paste it on the GitHub key and press Add SSH key.



- 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.



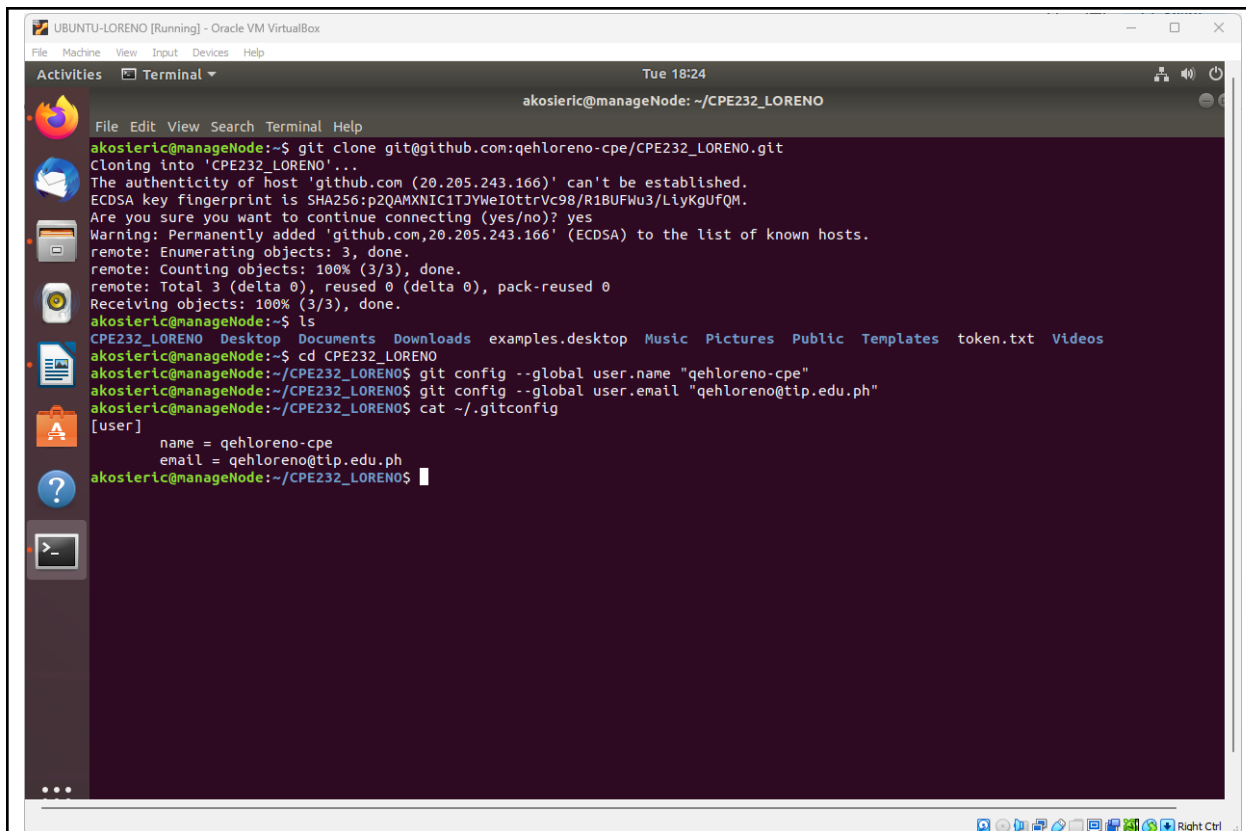


e. Issue the command `git clone` followed by the copied link. For example, `git clone git@github.com:jvtaylor-cpe/CPE302\_yourname.git`. When prompted to continue connecting, type yes and press enter.

f. To verify that you have cloned the GitHub repository, issue the command `ls`. Observe that you have the `CPE302_yourname` in the list of your directories. Use `CD` command to go to that directory and `LS` command to see the file `README.md`.

g. Use the following commands to personalize your git.

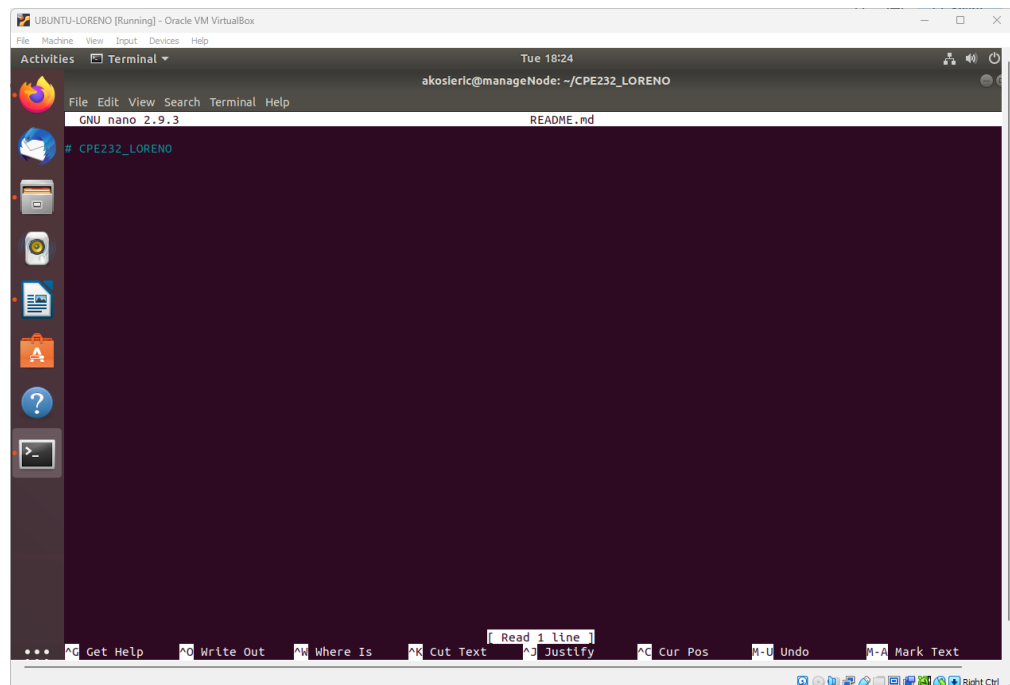
- `git config --global user.name "Your Name"`
- `git config --global user.email yourname@email.com`
- Verify that you have personalized the config file using the command `cat ~/.gitconfig`



```
UBUNTU-LORENO [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal
Tue 18:24
akosleric@manageNode: ~/CPE232_LORENO

File Edit View Search Terminal Help
akosleric@manageNode:~$ git clone git@github.com:qehloreno-cpe/CPE232_LORENO.git
Cloning into 'CPE232_LORENO'...
The authenticity of host 'github.com (20.205.243.166)' can't be established.
ECDSA key fingerprint is SHA256:p2QAMXNIC1TJYWeI0trVc98/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 known hosts.
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
akosleric@manageNode:~$ ls
CPE232_LORENO Desktop Documents Downloads examples.desktop Music Pictures Public Templates token.txt Videos
akosleric@manageNode:~$ cd CPE232_LORENO
akosleric@manageNode:~/CPE232_LORENO$ git config --global user.name "qehloreno-cpe"
akosleric@manageNode:~/CPE232_LORENO$ git config --global user.email "qehloreno@tip.edu.ph"
akosleric@manageNode:~/CPE232_LORENO$ cat ~/.gitconfig
[user]
  name = qehloreno-cpe
  email = qehloreno@tip.edu.ph
akosleric@manageNode:~/CPE232_LORENO$
```

- 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.

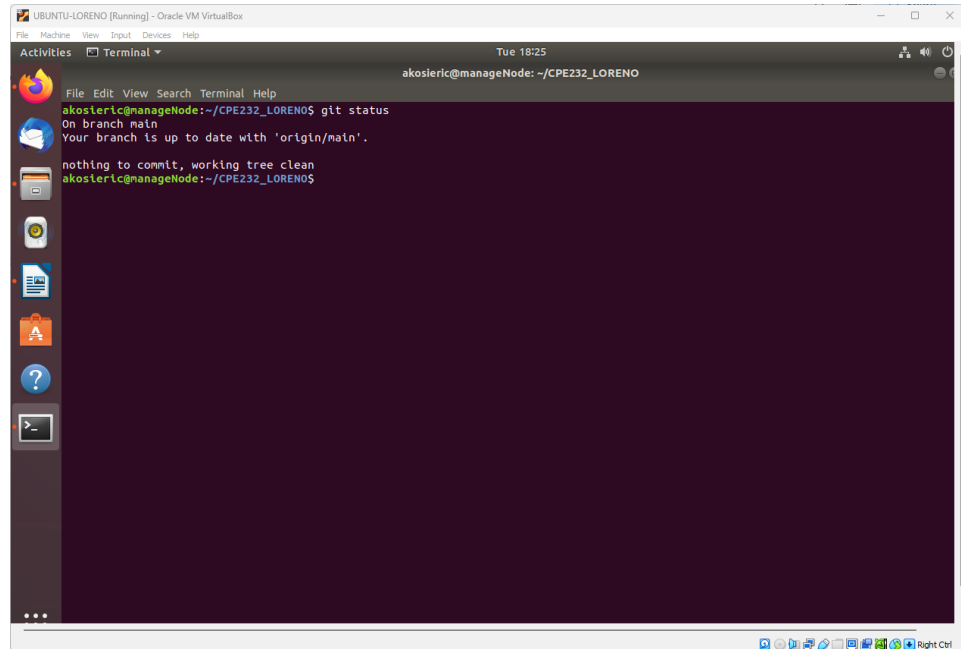


```
UBUNTU-LORENO [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal
Tue 18:24
akosleric@manageNode: ~/CPE232_LORENO

GNU nano 2.9.3 README.md
# CPE232_LORENO

[ Read 1 line ]
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^Q Cur Pos ^U Undo ^A Mark Text
```

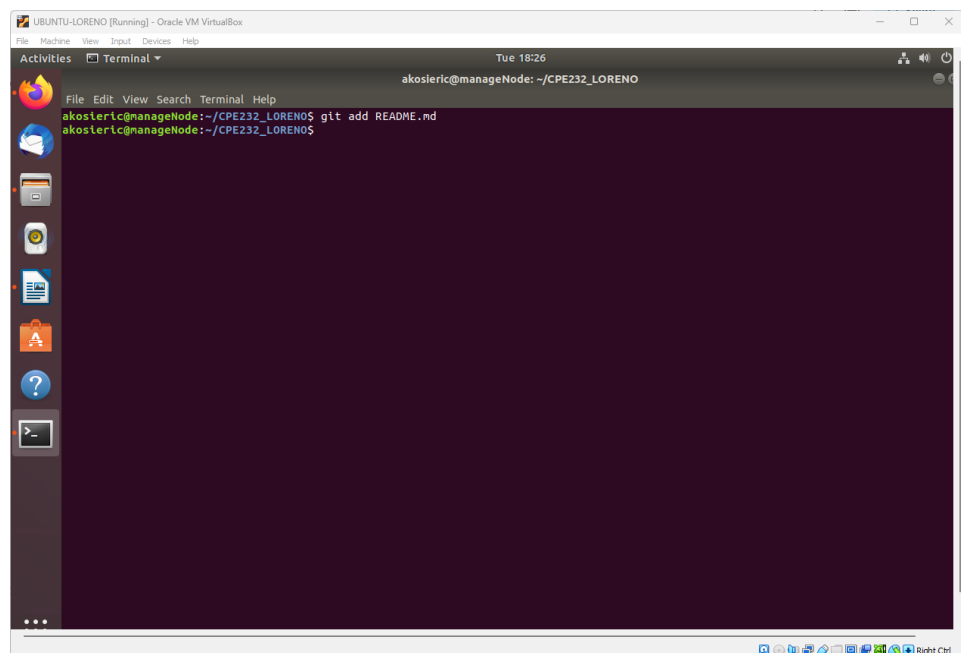
- i. Use the *git status* command to display the state of the working directory and the staging area. This command shows which changes have been staged, which haven't, and which files aren't being tracked by Git. Status output does not show any information regarding the committed project history. What is the result of issuing this command?



```
akosleric@manageNode: ~/CPE232_LORENO
akosleric@manageNode:~/CPE232_LORENO$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
akosleric@manageNode:~/CPE232_LORENO$
```

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

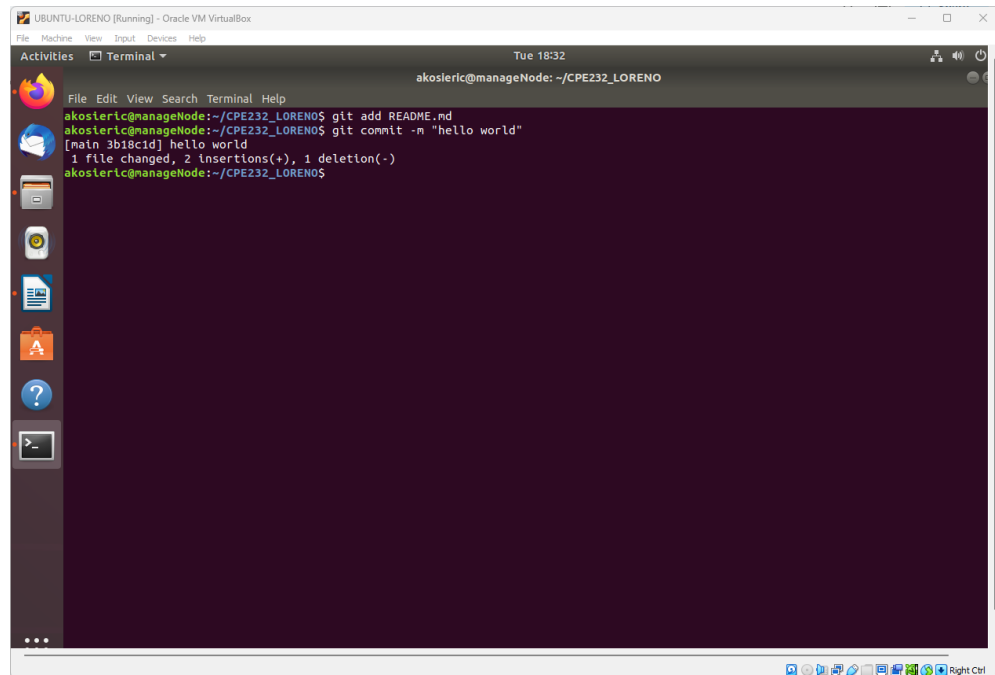


```
akosleric@manageNode: ~/CPE232_LORENO
akosleric@manageNode:~/CPE232_LORENO$ git add README.md
akosleric@manageNode:~/CPE232_LORENO$
```

- 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

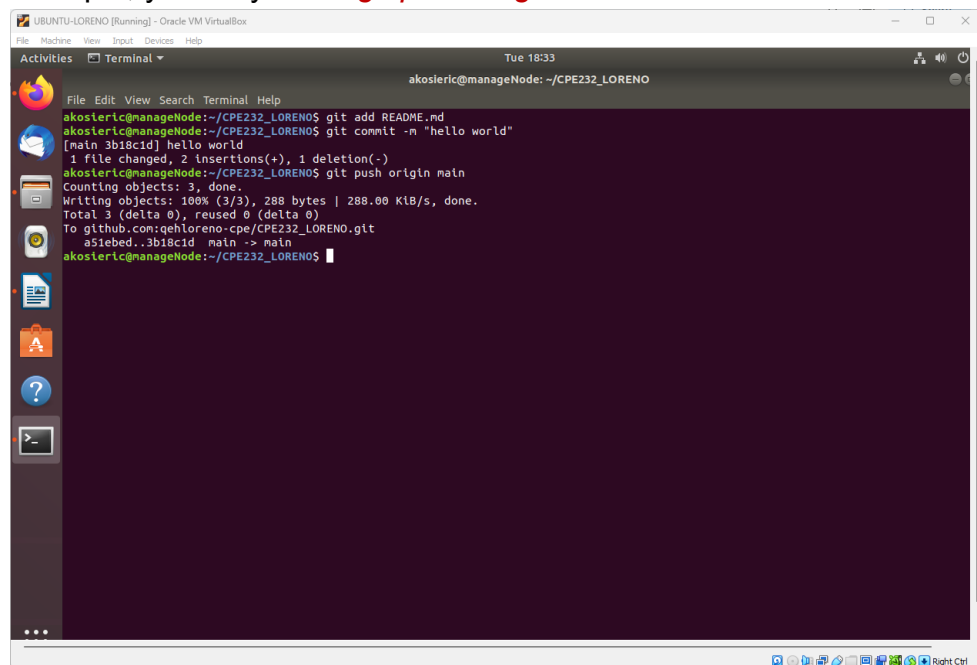


this command is required to select the changes that will be staged for the next commit.



```
akosieric@manageNode: ~/CPE232_LORENO
akosieric@manageNode:~/CPE232_LORENO$ git add README.md
akosieric@manageNode:~/CPE232_LORENO$ git commit -m "hello world"
[main 3b18c1d] hello world
1 file changed, 2 insertions(+), 1 deletion(-)
```

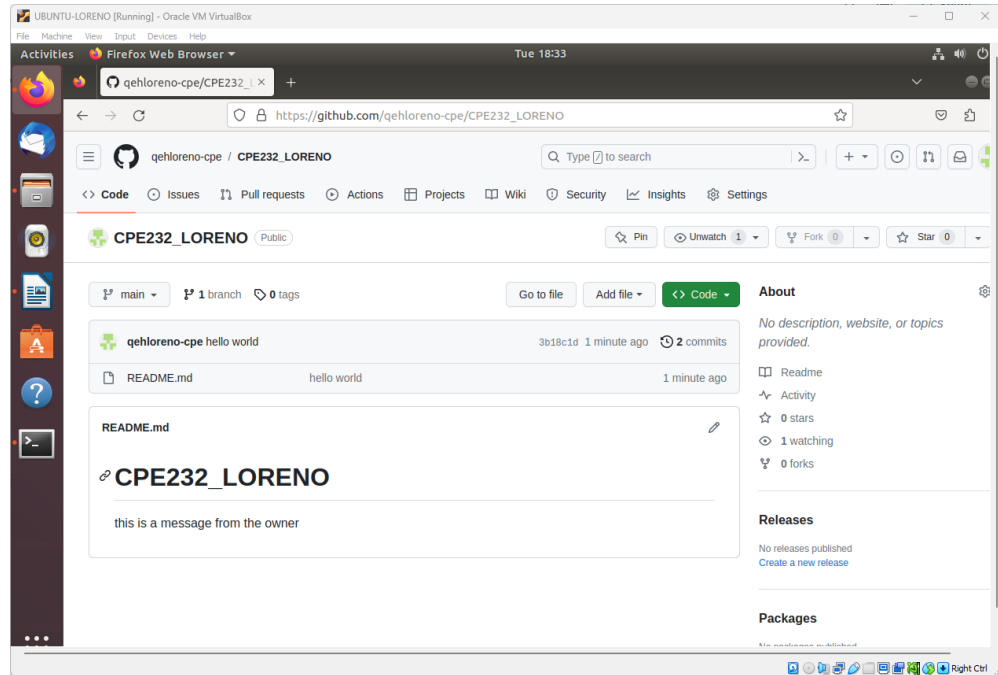
- I. Use the command *git push <remote><branch>* to upload the local repository content to GitHub repository. Pushing means to transfer commits from the local repository to the remote repository. As an example, you may issue *git push origin main*.



```
akosieric@manageNode:~/CPE232_LORENO$ git push origin main
Counting objects: 3, done.
Writing objects: 100% (3/3), 288 bytes | 288.00 KIB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To github.com:qehloreno-cpe/CPE232_LORENO.git
 a51ebd..3b18c1d  main -> main
akosieric@manageNode:~/CPE232_LORENO$
```

- 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.



## Reflections:

Answer the following:

3. What sort of things have we so far done to the remote servers using ansible commands?
  - ***We manipulated different files and directories to commit and push a file onto a github account. We also generated SSH keys using the terminal.***
4. How important is the inventory file?
  - ***The inventory file in Ansible is really important as it serves as the foundation for managing remote servers. It defines the hosts and groups of hosts that Ansible will target for various tasks. The inventory file enables dynamic grouping, specifying variables, and controlling the execution of playbooks. Its importance lies in providing a structured and maintainable way to manage a wide range of hosts and their configurations, making it an essential component for effective automation, scalability, and consistency in Ansible-based infrastructure management.***

**Conclusions/Learnings:**

- SSH key-based authentication is a secure and efficient method for authenticating to remote servers and services. By generating a key pair and storing the public key on the server, it eliminates the need for transmitting passwords over networks, enhancing security. This mechanism also plays a pivotal role in Git setup, allowing developers to securely interact with repositories. Git leverages SSH keys to establish secure connections between local repositories and remote Git servers, facilitating seamless collaboration and version control while maintaining data integrity. In both cases, SSH key-based authentication establishes a robust foundation for secure remote access and distributed version control.