Name: Renier L. Lope	**Date Performed: 09/09/2024**
Course/Section: CPE212 – CPE31S2	**Date Submitted: 09/10/2024**
Instructor: Robin Valenzuela	**Semester and SY: 1st 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.

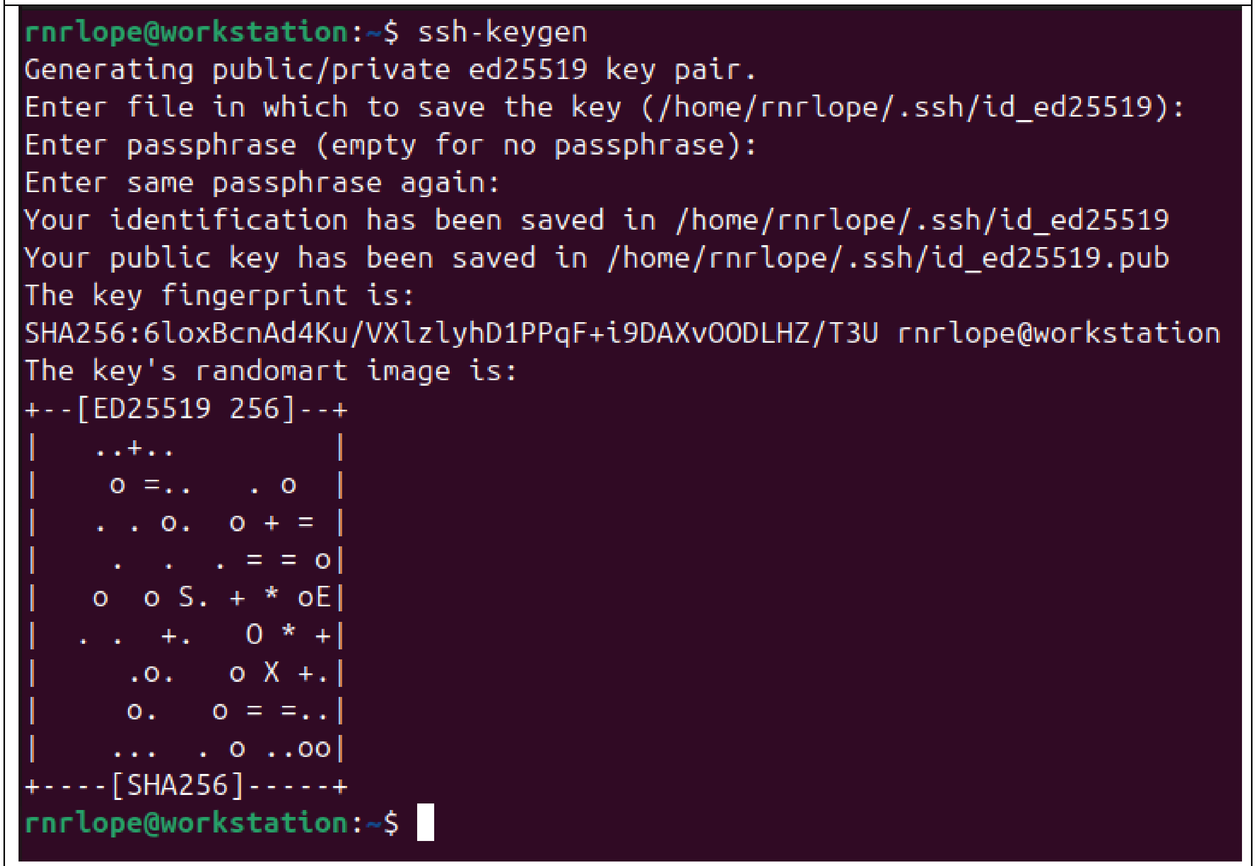
Task 1: Create an SSH Key Pair for User Authentication

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*.

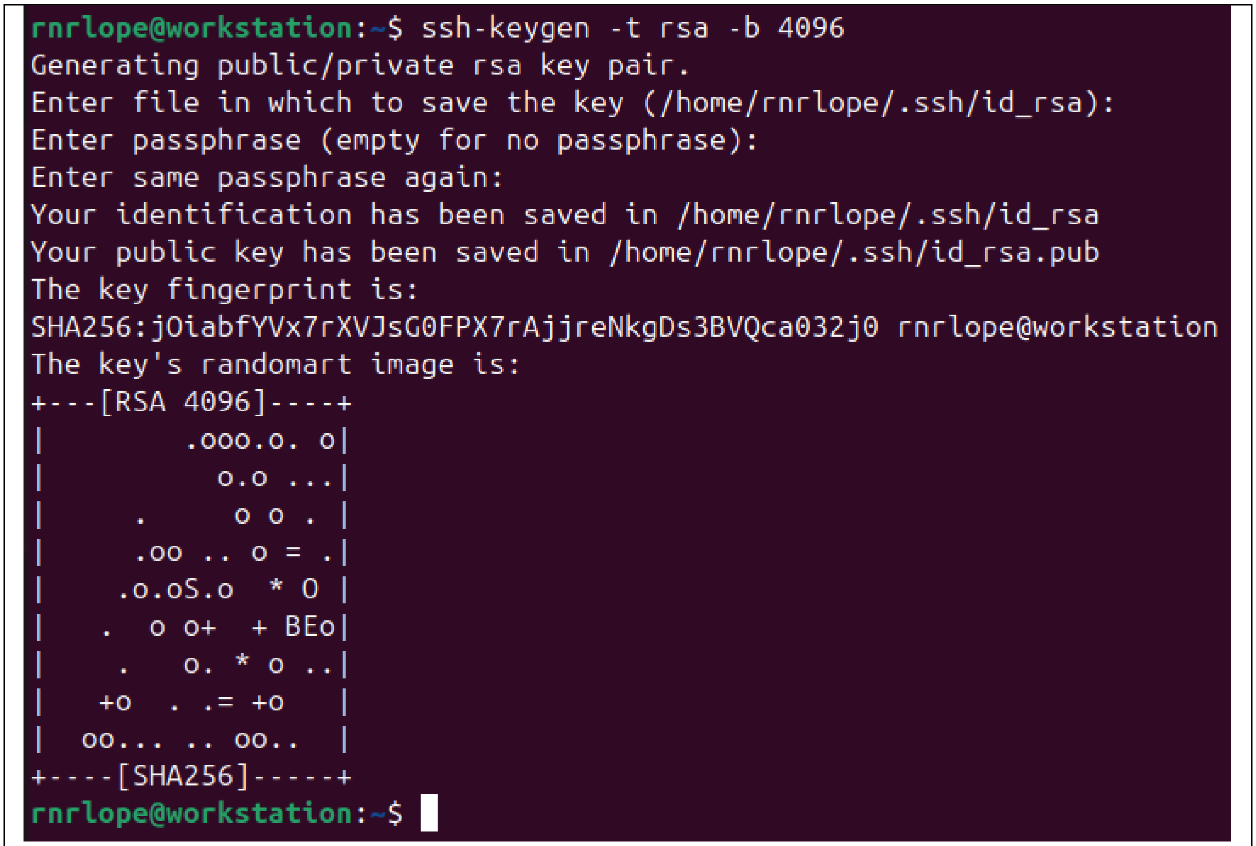
2. Issue the command *ssh-keygen -t rsa -b 4096.* The algorithm is selected using the -t option and key size using the -b option.
3. When asked for a passphrase, just press enter. The passphrase is used for encrypting the key, so that it cannot be used even if someone obtains the private key file. The passphrase should be cryptographically strong.
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.

Task 1 Documentation



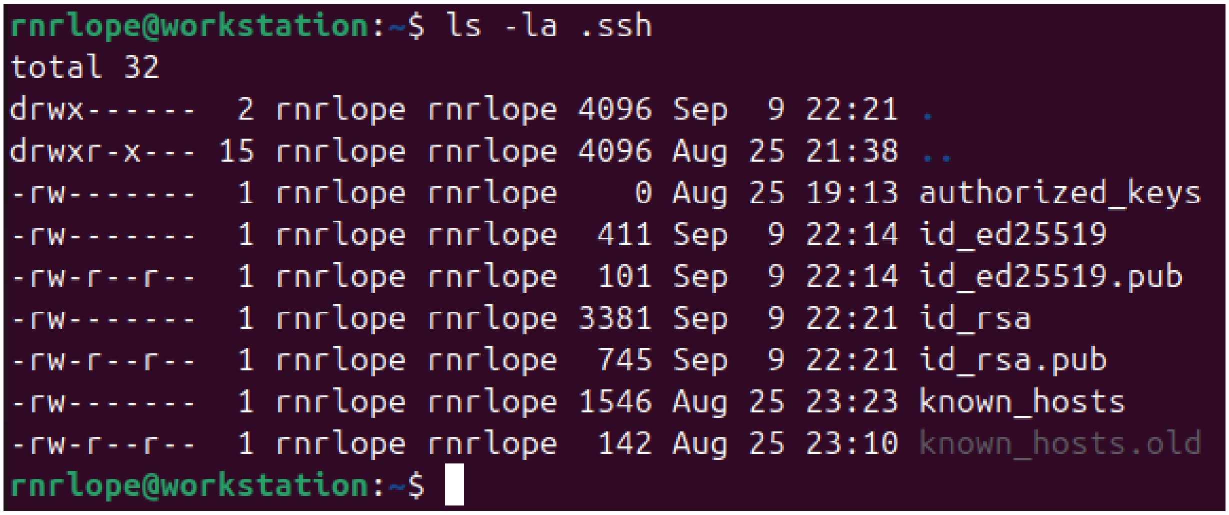
```
rnrlope@workstation:~$ ssh-keygen
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/rnrlope/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/rnrlope/.ssh/id_ed25519
Your public key has been saved in /home/rnrlope/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:6loxBcnAd4Ku/VXlzlyhD1PPqF+i9DAXvOODLHZ/T3U rnrlope@workstation
The key's randomart image is:
+--[ED25519 256]--+
|   ..+..         |
|    o =..   . o  |
|   . . o.  o + = |
|    .  .  . = = o|
|   o  o S. + * oE|
|  . .  +.   O * +|
|     .o.   o X +.|
|     o.   o = =..|
|    ...  . o ..oo|
+----[SHA256]-----+
rnrlope@workstation:~$ 
```

Figure 1.1 Running ssh-keygen without arguments



```
rnrlope@workstation:~$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/rnrlope/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/rnrlope/.ssh/id_rsa
Your public key has been saved in /home/rnrlope/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:jOiabfYVx7rXVJsG0FPX7rAjjreNkgDs3BVQca032j0 rnrlope@workstation
The key's randomart image is:
+---[RSA 4096]----+
|        .ooo.o. o|
|          o.o ...|
|     .     o o . |
|     .oo .. o = .|
|    .o.oS.o  * O |
|   .  o o+  + BEo|
|    .   o. * o ..|
|   +o  . .= +o   |
|  oo... .. oo..  |
+----[SHA256]-----+
rnrlope@workstation:~$
```

Figure 1.2 Running ssh-keygen -t rsa -b 4096



```
rnrlope@workstation:~$ ls -la .ssh
total 32
drwx------  2 rnrlope rnrlope 4096 Sep  9 22:21 .
drwxr-x--- 15 rnrlope rnrlope 4096 Aug 25 21:38 ..
-rw-------  1 rnrlope rnrlope    0 Aug 25 19:13 authorized_keys
-rw-------  1 rnrlope rnrlope  411 Sep  9 22:14 id_ed25519
-rw-r--r--  1 rnrlope rnrlope  101 Sep  9 22:14 id_ed25519.pub
-rw-------  1 rnrlope rnrlope 3381 Sep  9 22:21 id_rsa
-rw-r--r--  1 rnrlope rnrlope  745 Sep  9 22:21 id_rsa.pub
-rw-------  1 rnrlope rnrlope 1546 Aug 25 23:23 known_hosts
-rw-r--r--  1 rnrlope rnrlope  142 Aug 25 23:10 known_hosts.old
rnrlope@workstation:~$
```

Figure 1.3 Verifying if the key is created

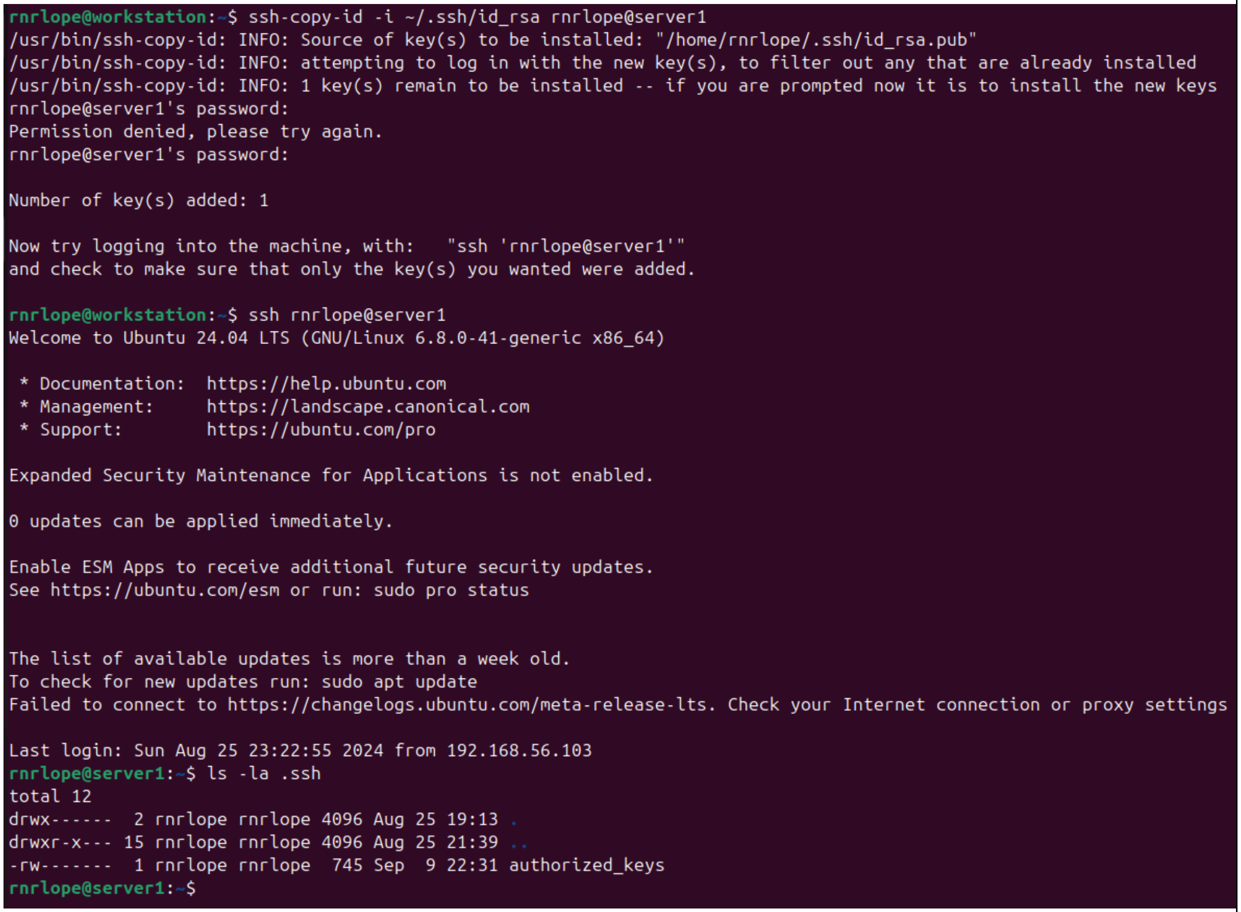
Task 2: Copying the Public Key to the remote servers

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*

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?

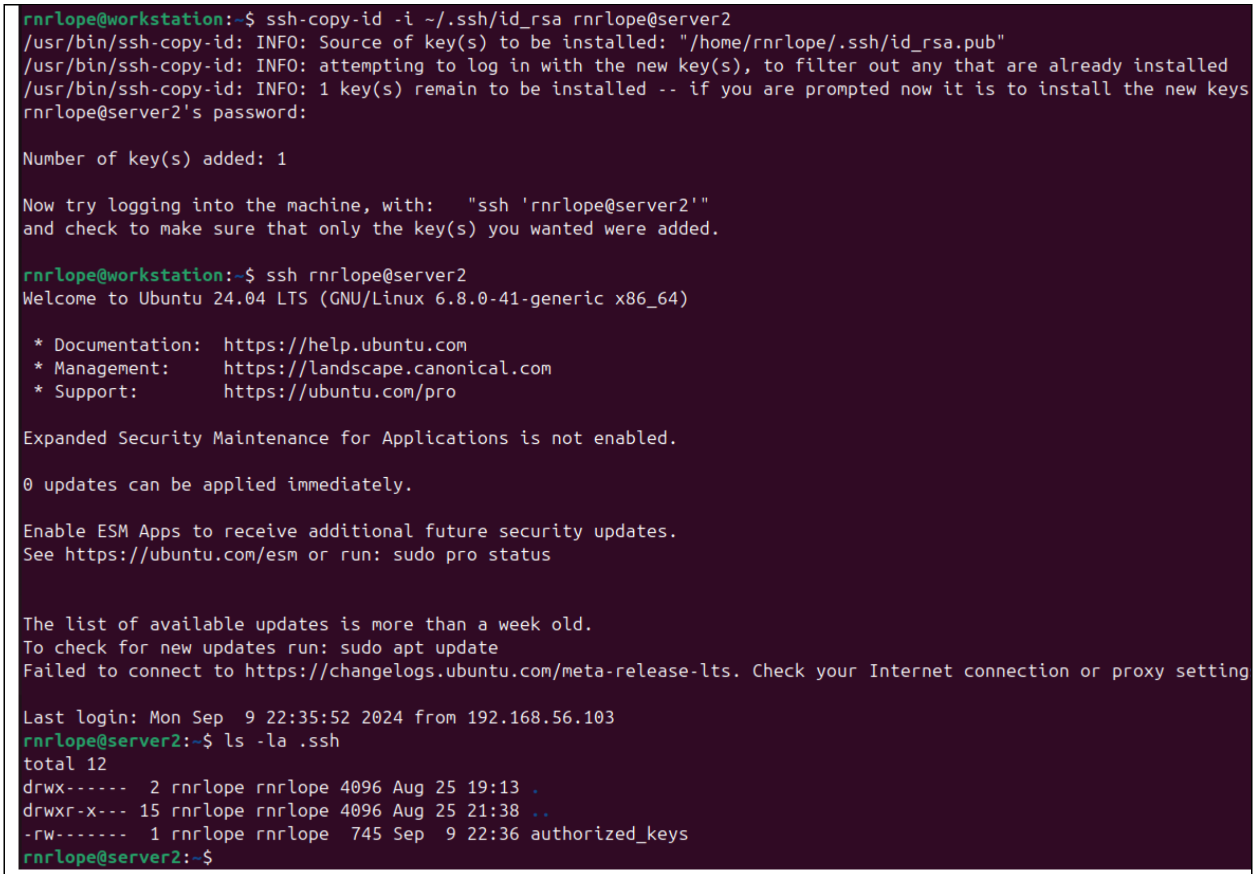
- **After Copying the public key to the Server1 and Server2 using ssh to connect to the servers, the connection did not ask for a password because the client proves possession of the private key by digitally signing the key exchange.**
-

Task 2 Documentation



```
rnrlope@workstation:~$ ssh-copy-id -i ~/.ssh/id_rsa rnrlope@server1
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/rnrlope/.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 prompted now it is to install the new keys
rnrlope@server1's password: 
Permission denied, please try again.
rnrlope@server1's password: 

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'rnrlope@server1'"
and check to make sure that only the key(s) you wanted were added.

rnrlope@workstation:~$ ssh rnrlope@server1
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-41-generic x86_64)

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

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

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


The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Last login: Sun Aug 25 23:22:55 2024 from 192.168.56.103
rnrlope@server1:~$ ls -la .ssh
total 12
drwx------  2 rnrlope rnrlope 4096 Aug 25 19:13 .
drwxr-x--- 15 rnrlope rnrlope 4096 Aug 25 21:39 ..
-rw-------  1 rnrlope rnrlope  745 Sep  9 22:31 authorized_keys
rnrlope@server1:~$
```

Figure 2.1 Copying and verifying public key for server1



```
rnrlope@workstation:~$ ssh-copy-id -i ~/.ssh/id_rsa rnrlope@server2
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/rnrlope/.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 prompted now it is to install the new keys
rnrlope@server2's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'rnrlope@server2'"
and check to make sure that only the key(s) you wanted were added.

rnrlope@workstation:~$ ssh rnrlope@server2
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-41-generic x86_64)

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

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

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


The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy setting

Last login: Mon Sep  9 22:35:52 2024 from 192.168.56.103
rnrlope@server2:~$ ls -la .ssh
total 12
drwx------  2 rnrlope rnrlope 4096 Aug 25 19:13 .
drwxr-x--- 15 rnrlope rnrlope 4096 Aug 25 21:38 ..
-rw-------  1 rnrlope rnrlope  745 Sep  9 22:36 authorized_keys
rnrlope@server2:~$
```

Figure 2.2 Copying and verifying public key for server2

Reflections:

Answer the following:

1. How will you describe the ssh-program? What does it do?
   - **SSH Protocol is a cryptographic network protocol that give users particularly system administrators a secure way to access a computer over an unsecured network. Remote login and command-line execution are its most notable application.**
2. How do you know that you already installed the public key to the remote servers?
   - **If connecting to them does not requires password anymore.**

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**).

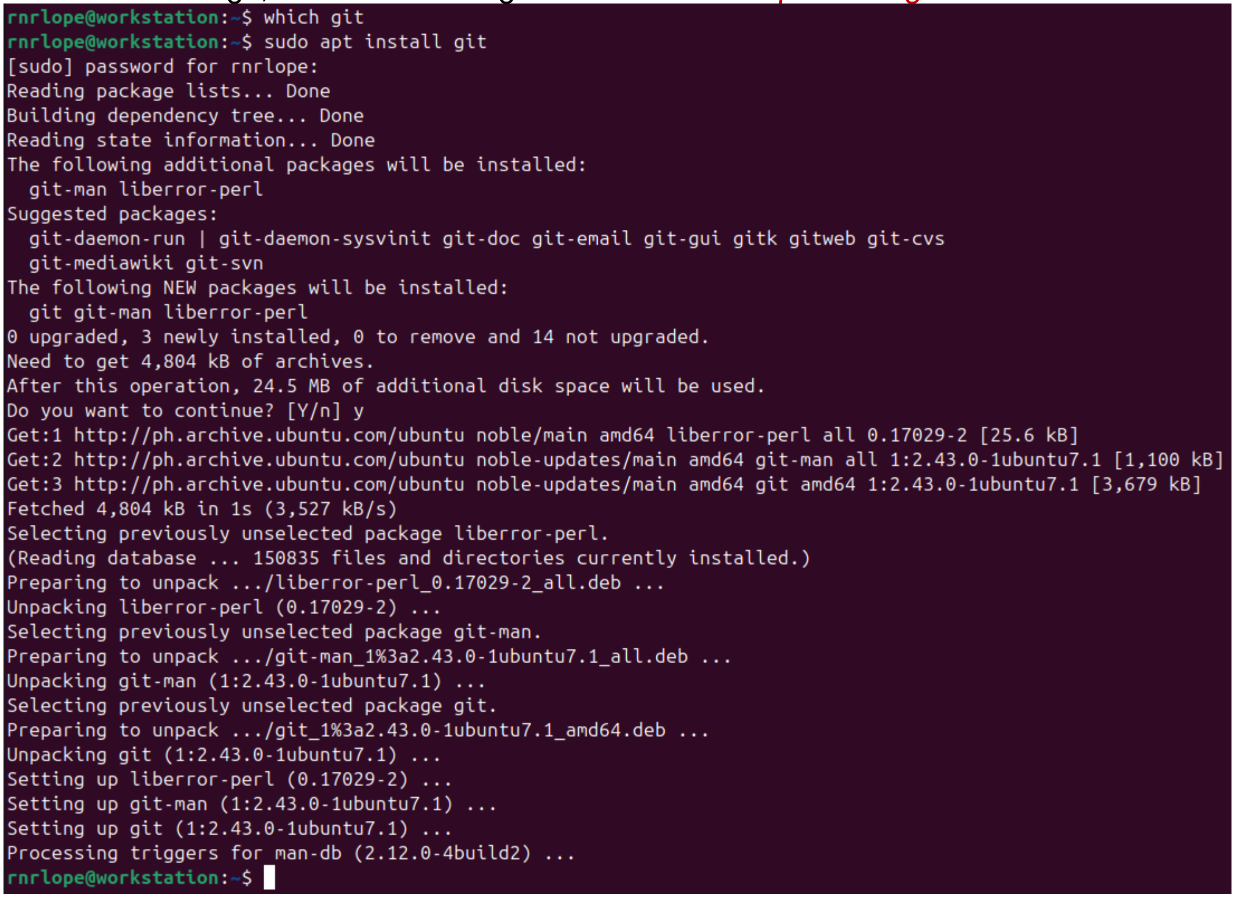
Set up Git

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

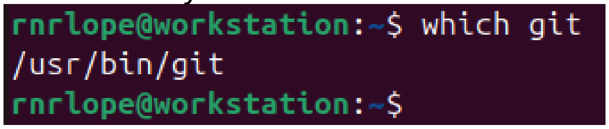
Task 3: Set up the Git Repository

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*



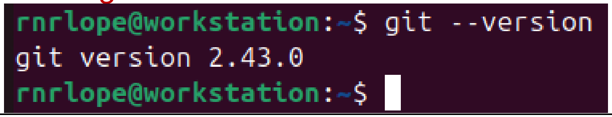
```
rnrlope@workstation:~$ which git
rnrlope@workstation:~$ sudo apt install git
[sudo] password for rnrlope:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  git-man liberror-perl
Suggested packages:
  git-daemon-run | git-daemon-sysvinit git-doc 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 14 not upgraded.
Need to get 4,804 kB of archives.
After this operation, 24.5 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ph.archive.ubuntu.com/ubuntu noble/main amd64 liberror-perl all 0.17029-2 [25.6 kB]
Get:2 http://ph.archive.ubuntu.com/ubuntu noble-updates/main amd64 git-man all 1:2.43.0-1ubuntu7.1 [1,100 kB]
Get:3 http://ph.archive.ubuntu.com/ubuntu noble-updates/main amd64 git amd64 1:2.43.0-1ubuntu7.1 [3,679 kB]
Fetched 4,804 kB in 1s (3,527 kB/s)
Selecting previously unselected package liberror-perl.
(Reading database ... 150835 files and directories currently installed.)
Preparing to unpack .../liberror-perl_0.17029-2_all.deb ...
Unpacking liberror-perl (0.17029-2) ...
Selecting previously unselected package git-man.
Preparing to unpack .../git-man_1%3a2.43.0-1ubuntu7.1_all.deb ...
Unpacking git-man (1:2.43.0-1ubuntu7.1) ...
Selecting previously unselected package git.
Preparing to unpack .../git_1%3a2.43.0-1ubuntu7.1_amd64.deb ...
Unpacking git (1:2.43.0-1ubuntu7.1) ...
Setting up liberror-perl (0.17029-2) ...
Setting up git-man (1:2.43.0-1ubuntu7.1) ...
Setting up git (1:2.43.0-1ubuntu7.1) ...
Processing triggers for man-db (2.12.0-4build2) ...
rnrlope@workstation:~$
```

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



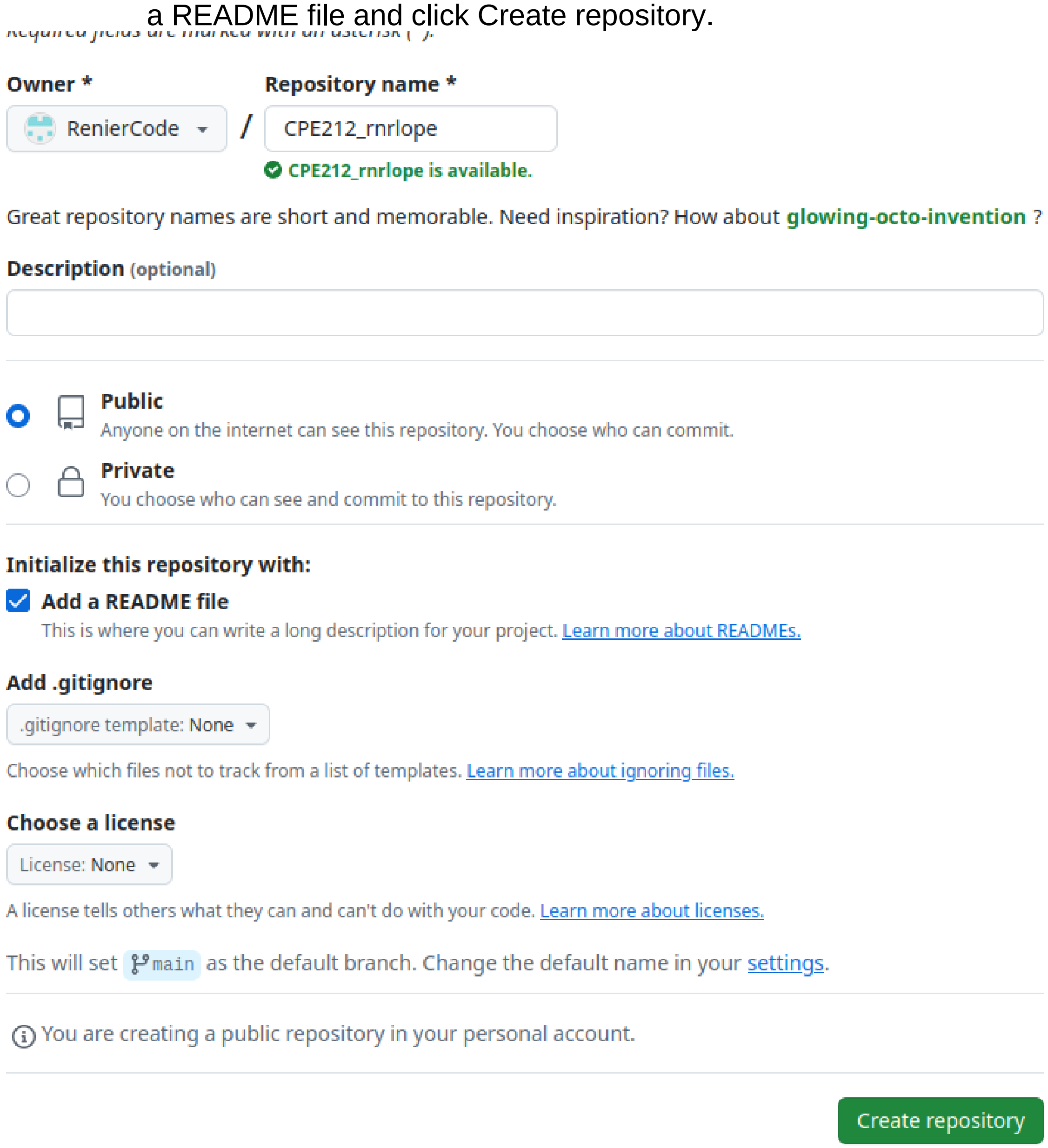
```
rnrlope@workstation:~$ which git
/usr/bin/git
rnrlope@workstation:~$
```

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



```
rnrlope@workstation:~$ git --version
git version 2.43.0
rnrlope@workstation:~$
```

4. Using the browser in the local machine, go to 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 CPE212_yourname. Check Add a README file and click Create repository.



Required fields are marked with an asterisk (*).

Owner * RenierCode / **Repository name *** CPE212_rnrlope

✓ CPE212_rnrlope is available.

Great repository names are short and memorable. Need inspiration? How about **glowing-octo-invention** ?

Description (optional)

● **Public**
Anyone on the internet can see this repository. You choose who can commit.

○ **Private**
You choose who can see and commit to this repository.

Initialize this repository with:

☑ **Add a README file**
This is where you can write a long description for your project. Learn more about READMEs.

Add .gitignore

.gitignore template: None

Choose which files not to track from a list of templates. Learn more about ignoring files.

Choose a license

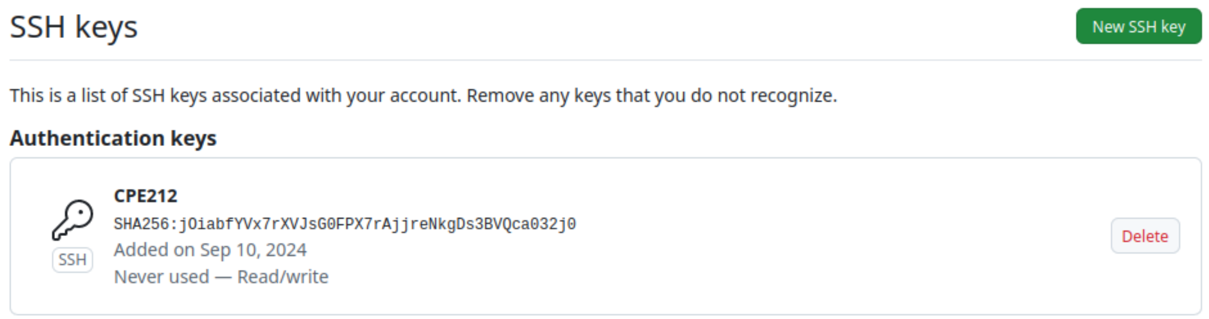
License: None

A license tells others what they can and can't do with your code. Learn more about licenses.

This will set `main` as the default branch. Change the default name in your settings.

ⓘ You are creating a public repository in your personal account.

    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 CPE212 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.

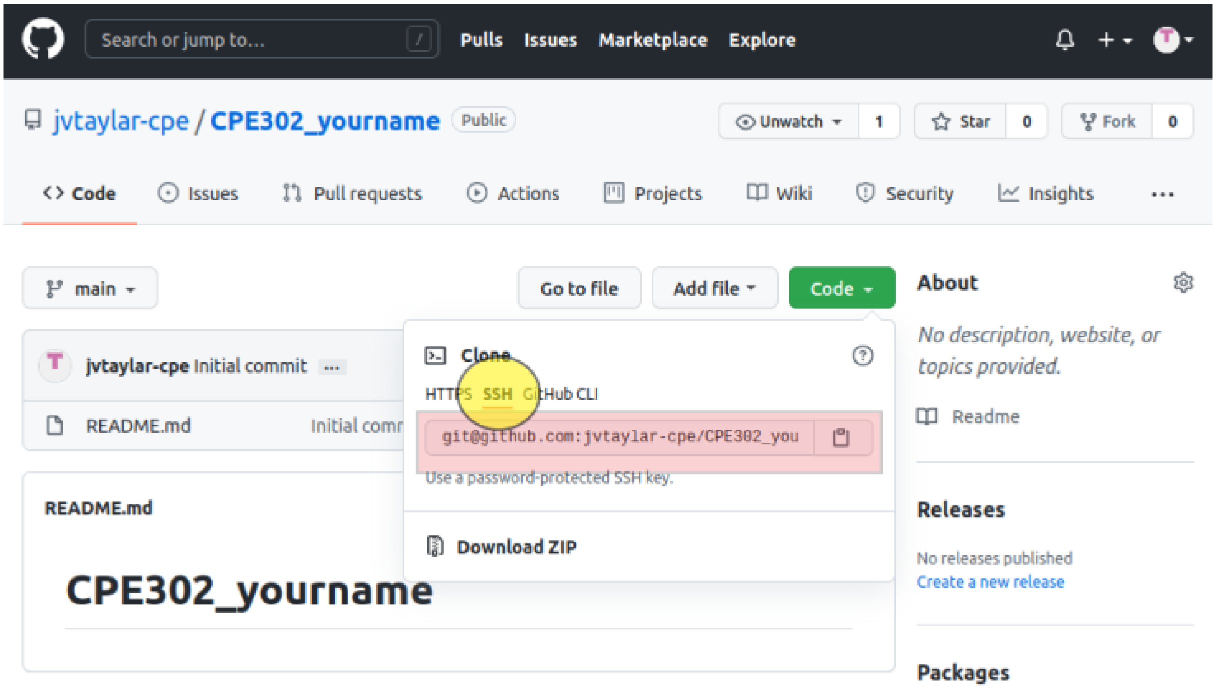


This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

```
SHA256:jOiabfYVx7rXVJsG0FPX7rAjjreNkgDs3BVQca032j0
```

Added on Sep 10, 2024
Never used — Read/write

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.



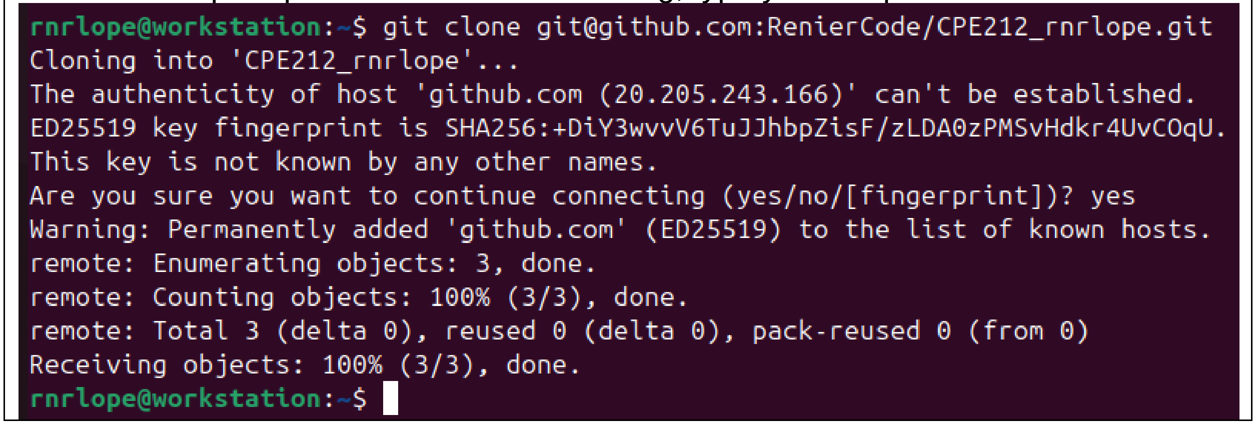
```
git@github.com:jvtaylar-cpe/CPE302_you
```

Use a password-protected SSH key.

No description, website, or topics provided.

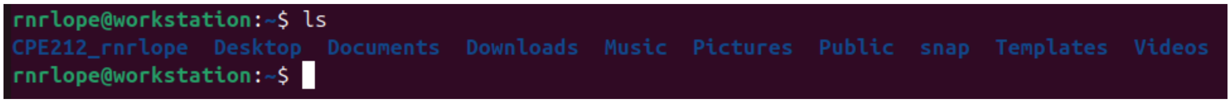
No releases published

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.



```
rnrlope@workstation:~$ git clone git@github.com:RenierCode/CPE212_rnrlope.git
Cloning into 'CPE212_rnrlope'...
The authenticity of host 'github.com (20.205.243.166)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvCOqU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'github.com' (ED25519) 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 (from 0)
Receiving objects: 100% (3/3), done.
rnrlope@workstation:~$ 
```

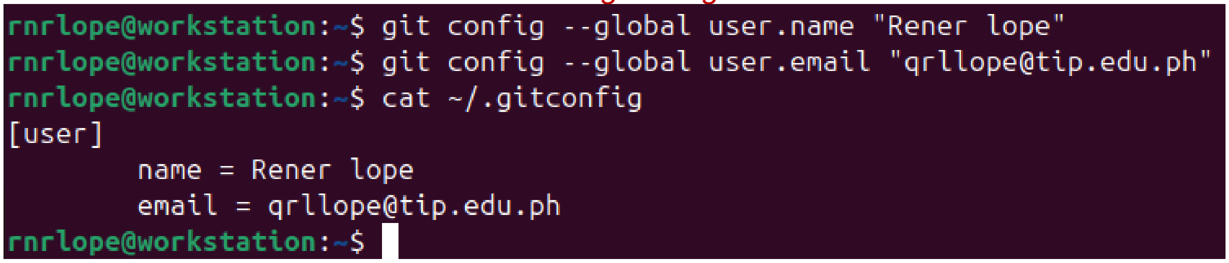
f. To verify that you have cloned the GitHub repository, issue the command *ls*. Observe that you have the CPE232_yourname in the list of your directories. Use CD command to go to that directory and LS command to see the file README.md.



```
rnrlope@workstation:~$ ls
CPE212_rnrlope  Desktop  Documents  Downloads  Music  Pictures  Public  snap  Templates  Videos
rnrlope@workstation:~$ 
```

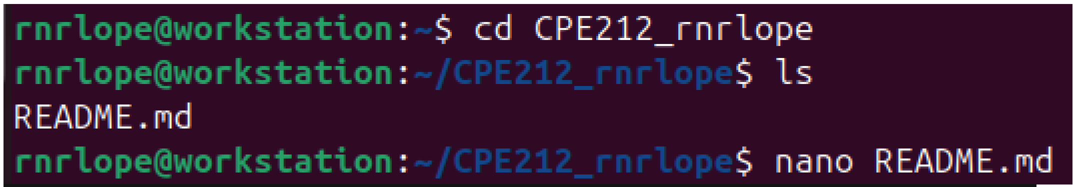
g. Use the following commands to personalize your git.

- *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*

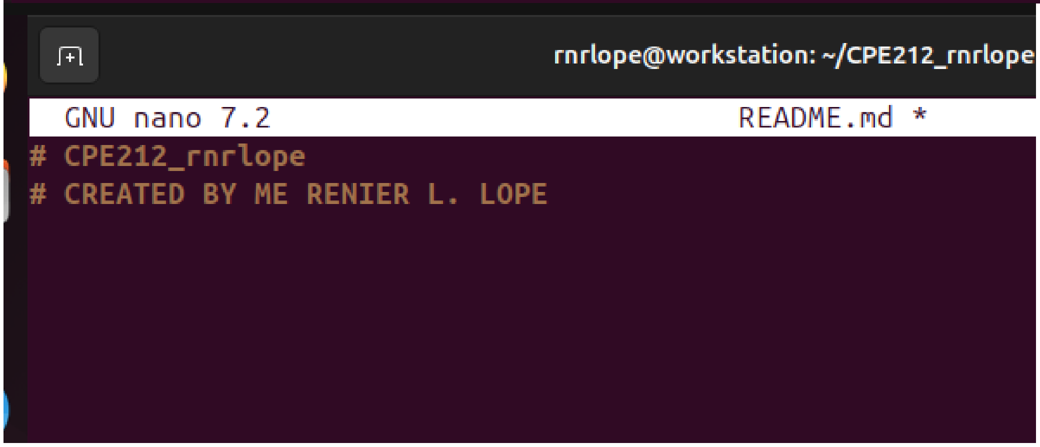


```
rnrlope@workstation:~$ git config --global user.name "Rener lope"
rnrlope@workstation:~$ git config --global user.email "qrllope@tip.edu.ph"
rnrlope@workstation:~$ cat ~/.gitconfig
[user]
        name = Rener lope
        email = qrllope@tip.edu.ph
rnrlope@workstation:~$ 
```

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.

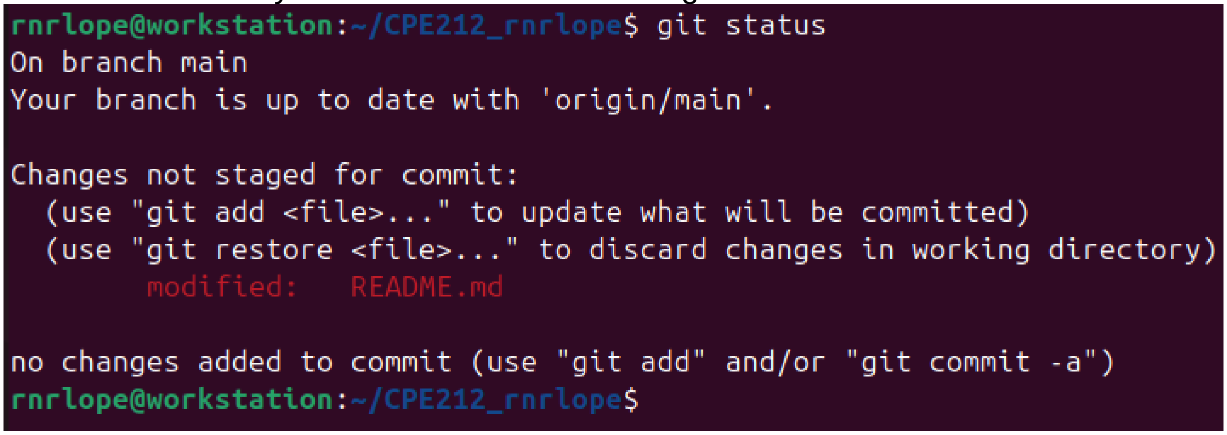


```
rnrlope@workstation:~$ cd CPE212_rnrlope
rnrlope@workstation:~/CPE212_rnrlope$ ls
README.md
rnrlope@workstation:~/CPE212_rnrlope$ nano README.md
```



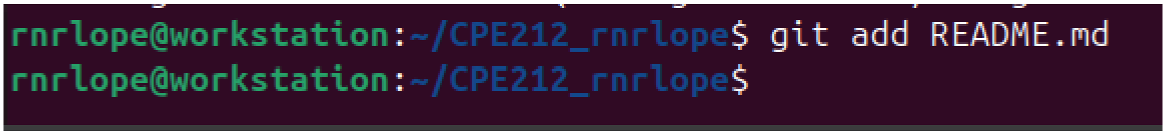
```
  GNU nano 7.2                                README.md *
# CPE212_rnrlope
# CREATED BY ME RENIER L. LOPE
```

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?



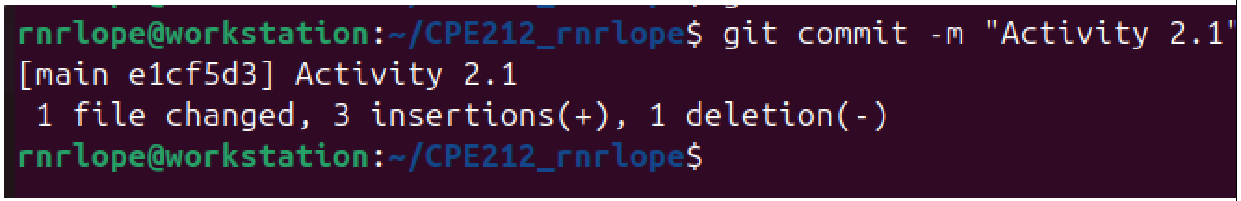
```
rnrlope@workstation:~/CPE212_rnrlope$ git status
On branch main
Your branch is up to date with 'origin/main'.

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

no changes added to commit (use "git add" and/or "git commit -a")
rnrlope@workstation:~/CPE212_rnrlope$
```

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



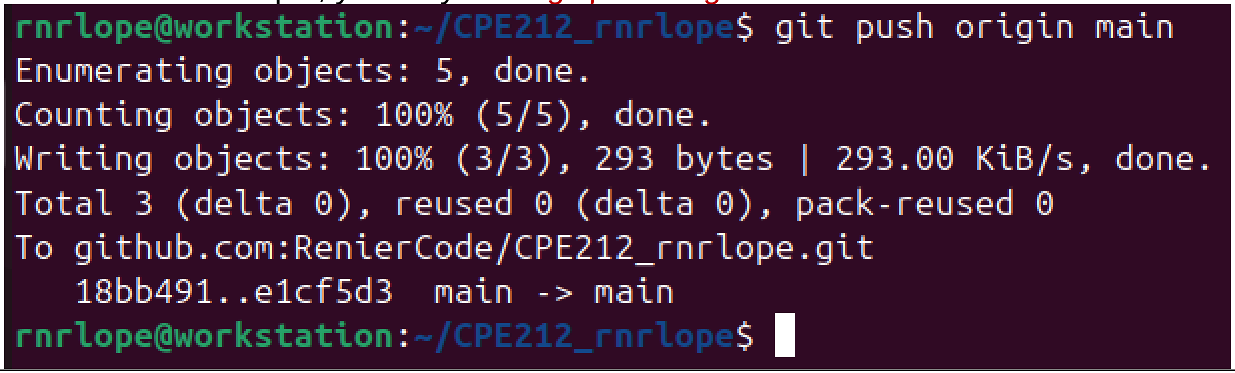
```
rnrlope@workstation:~/CPE212_rnrlope$ git add README.md
rnrlope@workstation:~/CPE212_rnrlope$
```

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.



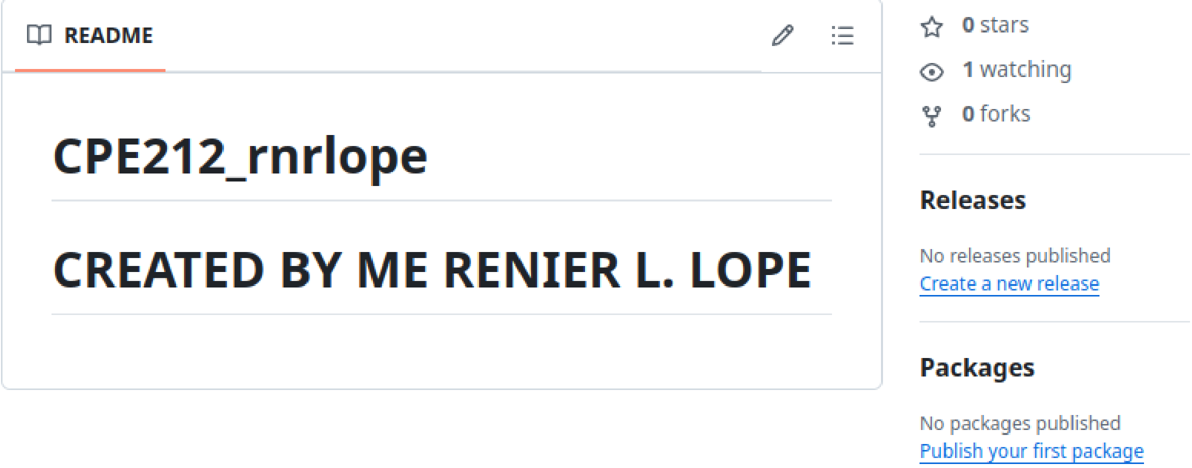
```
rnrlope@workstation:~/CPE212_rnrlope$ git commit -m "Activity 2.1"
[main e1cf5d3] Activity 2.1
 1 file changed, 3 insertions(+), 1 deletion(-)
rnrlope@workstation:~/CPE212_rnrlope$
```

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.*



```
rnrlope@workstation:~/CPE212_rnrlope$ git push origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Writing objects: 100% (3/3), 293 bytes | 293.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:RenierCode/CPE212_rnrlope.git
   18bb491..e1cf5d3  main -> main
rnrlope@workstation:~/CPE212_rnrlope$
```

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.



CPE212_rnrlope

CREATED BY ME RENIER L. LOPE

Reflections:

Answer the following:

3. What sort of things have we so far done to the remote servers using ansible commands?
4. How important is the inventory file?

Conclusions/Learnings:

- **Throughout the process of following the instructions and completing this activity, I manage to learn and demonstrate how to create an SSH Key Pair for User Authentication, how to copy the public to servers, and to install git and most importantly is to create a GitHub repository and to also clone it to my local machine to get local repository that will act as a connection with the main GitHub repository. In the end I manage to upload or transfer the changes and commits that I made in the local repository to the GitHub repository.**