

Name: ROALLOS, Jean Gabriel Vincent G.	Date Performed: 08 / 15 / 2025
Course/Section: CPE212 - CPE31S2	Date Submitted: 08 / 15 / 2025
Instructor: Engr. Robin Valenzuela	Semester and SY: 1st, 2025 - 2026
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). <i>Provide screenshots for each task.</i> 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 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?

The SSH connection did not ask for a password before logging into Server 1 remotely. This is possible due to the public key we assigned to Server 1.

```
roallos-ubuntu@workstation: ~
File Edit View Search Terminal Help

roallos-ubuntu@workstation:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/roallos-ubuntu/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/roallos-ubuntu/.ssh/id_rsa.
Your public key has been saved in /home/roallos-ubuntu/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:VkzuKjXevqzr7yB2l+TpAlHYN8BOVnCEm4fB85EuW7U roallos-ubuntu@workstation
The key's randomart image is:
+----[RSA 2048]-----+
|      =o*=.      |
|      . @== .    |
|      = X++ .    |
|      . *o= E    |
|      .S=o       |
|      .+.* o     |
|      +.= *      |
|      . +.*      |
|      .+*B.      |
+----[SHA256]-----+
roallos-ubuntu@workstation:~$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/roallos-ubuntu/.ssh/id_rsa):
/home/roallos-ubuntu/.ssh/id_rsa already exists.
Overwrite (y/n)?
roallos-ubuntu@workstation:~$ █

roallos-ubuntu@workstation:~$ ls -la .ssh
total 20
drwx----- 2 roallos-ubuntu roallos-ubuntu 4096 Aug 15 16:52 .
drwxr-xr-x 17 roallos-ubuntu roallos-ubuntu 4096 Aug 15 17:10 ..
-rw----- 1 roallos-ubuntu roallos-ubuntu 1679 Aug 15 16:47 id_rsa
-rw-r--r-- 1 roallos-ubuntu roallos-ubuntu 408 Aug 15 16:47 id_rsa.pub
-rw-r--r-- 1 roallos-ubuntu roallos-ubuntu 1332 Aug 15 17:11 known_hosts
roallos-ubuntu@workstation:~$
```

```
roallos-ubuntu@server1: ~  
File Edit View Search Terminal Help  
roallos-ubuntu@workstation:~$ ssh-copy-id -i ~/.ssh/id_rsa roallos-ubuntu@192.168.56.105  
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/roallos-ubuntu/.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  
roallos-ubuntu@192.168.56.105's password:  
  
Number of key(s) added: 1  
  
Now try logging into the machine, with: "ssh 'roallos-ubuntu@192.168.56.105'" and check to make sure that only the key(s) you wanted were added.
```

```
roallos-ubuntu@server1: ~  
File Edit View Search Terminal Help  
Now try logging into the machine, with: "ssh 'roallos-ubuntu@192.168.56.105'" and check to make sure that only the key(s) you wanted were added.  
  
roallos-ubuntu@workstation:~$ ssh roallos-ubuntu@192.168.56.105  
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/pro  
  
Expanded Security Maintenance for Infrastructure is not enabled.  
  
0 updates can be applied immediately.  
  
226 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: Fri Aug  8 18:27:10 2025 from 192.168.56.104  
roallos-ubuntu@server1:~$ mkdir testdir  
roallos-ubuntu@server1:~$ ls  
Desktop    Downloads  Music      Public     testdir  
Documents  examples.desktop  Pictures   Templates  Videos  
roallos-ubuntu@server1:~$
```

Reflections:

Answer the following:

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

It allows systems to connect to other systems via Secure Shell (SSH).

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

One way to find out is to login to the servers you have assigned the public key to. Connecting to that server would be easier due to not asking a password upon connection.

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

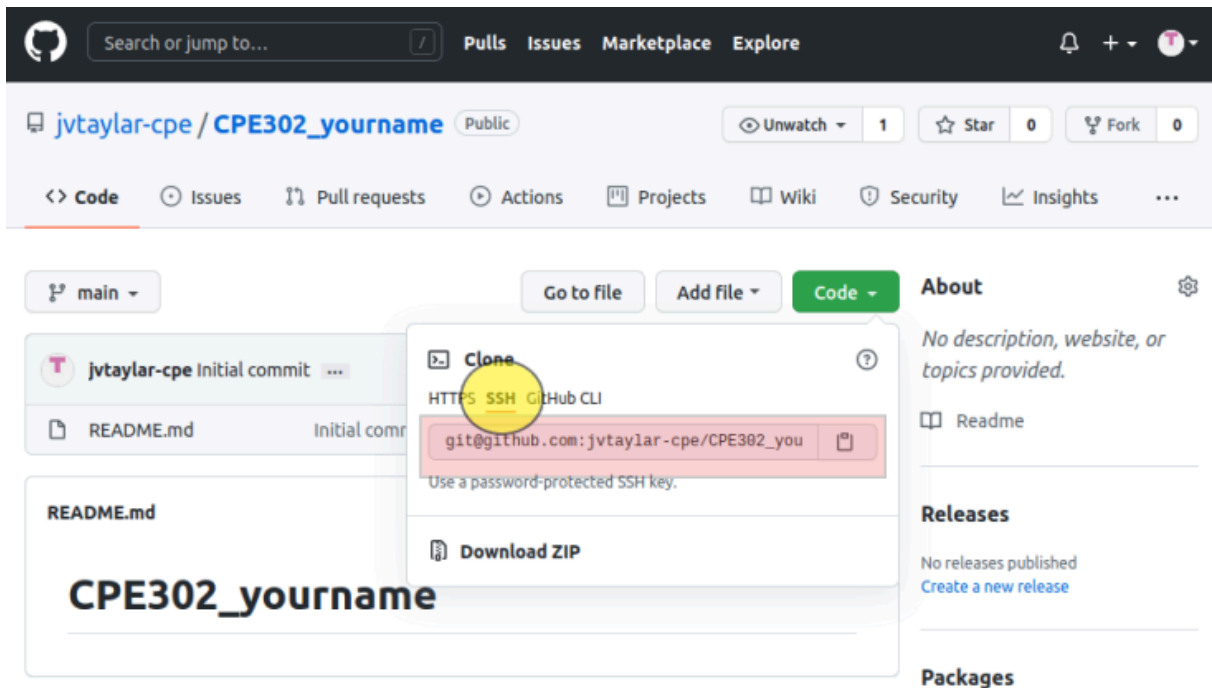
At the heart of GitHub is an open-source version control system (VCS) called Git. Git is responsible for everything GitHub-related that happens locally on your computer. To use Git on the command line, you'll need to download, install, and configure Git on your computer. You can also install GitHub CLI 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***
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.
5. Sign up in case you don't have an account yet. Otherwise, login to your GitHub account.

- Create a new repository and name it as CPE232_yourname. Check Add a README file and click Create repository.
- 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.
- 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.
- 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.



- Issue the command `git clone` followed by the copied link. For example, `git clone git@github.com:jvtaylor-cpe/CPE232_yourname.git`. When prompted to continue connecting, type yes and press enter.
- 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.
- Use the following commands to personalize your git.
 - `git config --global user.name "Your Name"`
 - `git config --global user.email yourname@email.com`

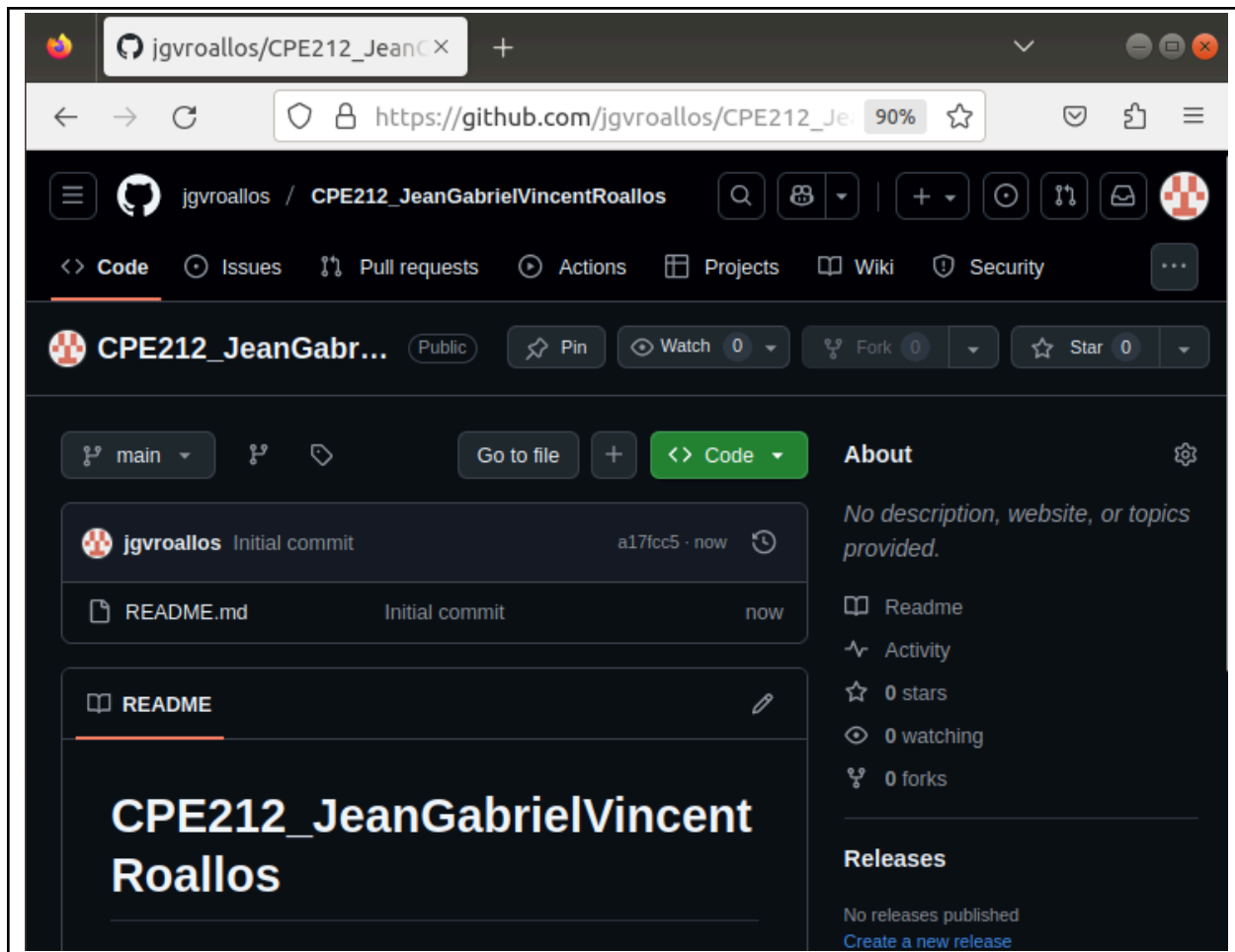
- Verify that you have personalized the config file using the command `cat ~/.gitconfig`
- h. Edit the README.md file using nano command. Provide any information on the markdown file pertaining to the repository you created. Make sure to write out or save the file and exit.
- i. Use the `git status` 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?

This displays the changes made into the repository before pushing into the 'main' branch.

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

```
root@workstation: ~
File Edit View Search Terminal Help
roallos-ubuntu is not in the sudoers file. This incident will be reported.
roallos-ubuntu@workstation:~$ su -
Password:
root@workstation:~# apt install git
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  libllvm7
Use '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]
36% [3 git 544 kB/3,990 kB 14%] 137 kB/s 25s

root@workstation:~# which git
/usr/bin/git
root@workstation:~# git --version
git version 2.17.1
root@workstation:~#
```

SSH and GPG keys

+


90%

https://github.com/settings/keys

☆

🔍

☰



Vincent Roallos (jgvroallos)
Your personal account

Go to your personal profile

Public profile

Account

Appearance

Accessibility

Notifications

Access

Billing and licensing

Emails

Password and authentication

Sessions

SSH and GPG keys

Organizations


Enterprises

SSH keys

New SSH key

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

Authentication keys



CPE212
SHA256:VkzuKjXevqzr7yB2l+TpA1HYN8B0VnCEm4fB85EuW7U
Added on Aug 15, 2025
Never used — Read/write

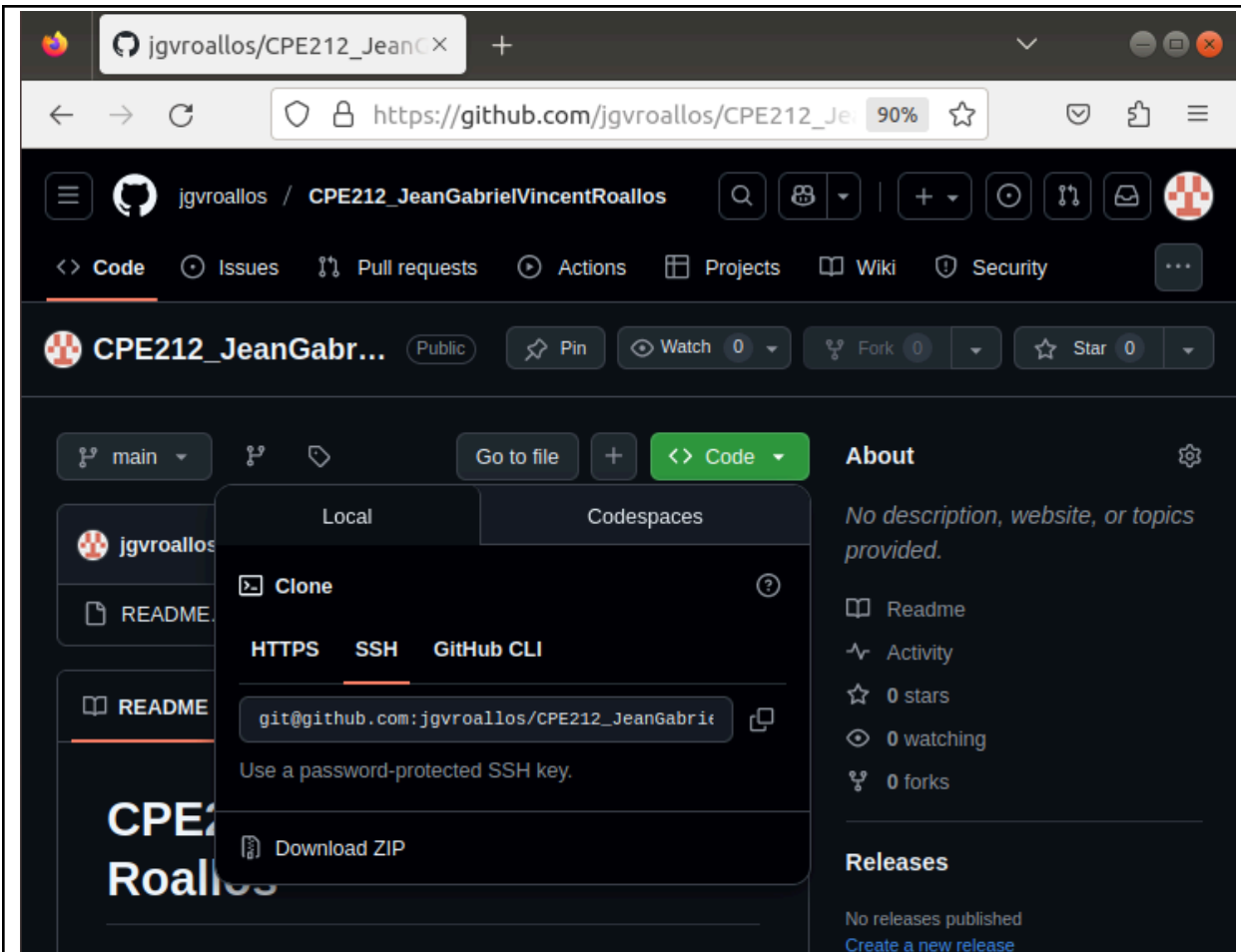
SSH

Delete

Check out our guide to [connecting to GitHub using SSH keys](#) or troubleshoot [common SSH problems](#).

GPG keys

New GPG key



```
roallos-ubuntu@workstation:~$ git clone git@github.com:jgvroallos/CPE212_JeanGabrielVincentRoallos.git
Cloning into 'CPE212_JeanGabrielVincentRoallos'...
The authenticity of host 'github.com (4.237.22.38)' 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,4.237.22.38' (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 (from 0)
Receiving objects: 100% (3/3), done.
roallos-ubuntu@workstation:~$

roallos-ubuntu@workstation:~/CPE212_JeanGabrielVincentRoallos$ git config --global user.name "VincentRoallos"
roallos-ubuntu@workstation:~/CPE212_JeanGabrielVincentRoallos$ git config user.name
VincentRoallos
roallos-ubuntu@workstation:~/CPE212_JeanGabrielVincentRoallos$ git config --global user.email qjgvgroallos@tip.edu.ph
roallos-ubuntu@workstation:~/CPE212_JeanGabrielVincentRoallos$ git config user.email
qjgvgroallos@tip.edu.ph
roallos-ubuntu@workstation:~/CPE212_JeanGabrielVincentRoallos$
```

```
roallos-ubuntu@workstation:~/CPE212_JeanGabrielVincentRoallos$ cat ~/.gitconfig
# This is Git's per-user configuration file.
[user]
# Please adapt and uncomment the following lines:
#     name = Roallos-Ubuntu
#     email = roallos-ubuntu@workstation.(none)
[user]
    name = VincentRoallos
    email = qjgvgroallos@tip.edu.ph
roallos-ubuntu@workstation:~/CPE212_JeanGabrielVincentRoallos$
```

```
roallos-ubuntu@workstation: ~/CPE212_JeanGabrielVincentRoallos
File Edit View Search Terminal Help
GNU nano 2.9.3 README.md
CPE212_JeanGabrielVincentRoallos
Created 08/15/2025. This repository contains file submissions for CPE212, Auto$
```

```
roallos-ubuntu@workstation: ~/CPE212_JeanGabrielVincentRoallos
File Edit View Search Terminal Help
GNU nano 2.9.3 README.md
# CPE212_JeanGabrielVincentRoallos
$, Automating Server Management.
```

```
roallos-ubuntu@workstation:~/CPE212_JeanGabrielVincentRoallos$ 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")
roallos-ubuntu@workstation:~/CPE212_JeanGabrielVincentRoallos$
```

```
roallos-ubuntu@workstation:~/CPE212_JeanGabrielVincentRoallos$ git add README.m
d
roallos-ubuntu@workstation:~/CPE212_JeanGabrielVincentRoallos$ git commit -m "E
dit README.md of REPO"
[main d5956c3] Edit README.md of REPO
1 file changed, 3 insertions(+), 1 deletion(-)
roallos-ubuntu@workstation:~/CPE212_JeanGabrielVincentRoallos$
```

```
roallos-ubuntu@workstation:~/CPE212_JeanGabrielVincentRoallos$ git push origin main
Counting objects: 3, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 383 bytes | 383.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To github.com:jgvroallos/CPE212_JeanGabrielVincentRoallos.git
a17fcc5..d5956c3 main -> main
roallos-ubuntu@workstation:~/CPE212_JeanGabrielVincentRoallos$
```

The screenshot shows a web browser window with the address bar displaying `https://github.com/jgvroallos/CPE212_JeanGabrielVincentRoallos`. The page shows the repository details for 'CPE212_JeanGabrielVincentRoallos', which is a public repository. The repository has 0 stars, 0 forks, and 0 watchers. The main branch is 'main'. The repository was created on 08/15/2025 and contains file submissions for CPE212, Automating Server Management. The README file is visible, and there is a button to 'Edit README.md of REPO'.

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:

I have learned how to set up Git and Github into a Linux system. Generating SSH keypairs, assigning public keys and establishing passwordless SSH logins into other (virtual) systems.

