

Name: Tamayo, Ray Lan A.	Date Performed: 09/04/2024
Course/Section: CPE212-CPE31S21	Date Submitted: 09/04/2024
Instructor: Engr. Robin Valenzuela	Semester and SY: First & 2024-2025
Activity 2: SSH Key-Based Authentication and Setting up Git	
<p>1. Objectives:</p> <ul style="list-style-type: none"> 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 	
<p>Part 1: Discussion</p> <p>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></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>What is ssh-keygen?</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>SSH Keys and Public Key Authentication</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>	

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`.

```
tamayo@workstation:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/tamayo/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/tamayo/.ssh/id_rsa
Your public key has been saved in /home/tamayo/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:J5l4huWhCzKlbD9X7fRrVvS1QV8XEarkIXnVCIk6xb0 tamayo@workstation
The key's randomart image is:
+---[RSA 3072]---+
|      . o.o.o++ |
|      +.o...oo |
|      .   =o +... + |
|      . o   0 ==Eo ..o |
|      * . + S +o . .+ |
|      . + . = .   ... |
|      o o   . .. |
|      o     o. |
|      o. |
|      o. |
+-----[SHA256]-----+
```

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.

```
tamayo@workstation:~$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/tamayo/.ssh/id_rsa):
/home/tamayo/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
```

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.

```

tamayo@workstation:~$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/tamayo/.ssh/id_rsa):
/home/tamayo/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/tamayo/.ssh/id_rsa
Your public key has been saved in /home/tamayo/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:fPu4hM/UWqdDTQryyJiDXd2XvJ7Gb6l300ZcItVkmxI tamayo@workstation
The key's randomart image is:
+----[RSA 4096]-----+
|
|      . . E .o|
|     .o o . *+o|
|    o =S+.. *.+.|
|   . = oo.oo.+ |
|  .. +.+ooooo|
|  = =.=0=o|
|  *.o*=+o|
|+-----[SHA256]-----+

```

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`.

```

tamayo@workstation:~$ ls -la .ssh
total 24
drwx----- 2 tamayo tamayo 4096 Sep  4 03:30 .
drwxr-x--- 15 tamayo tamayo 4096 Sep  4 01:54 ..
-rw----- 1 tamayo tamayo  0 Sep  4 01:50 authorized_keys
-rw----- 1 tamayo tamayo 3381 Sep  4 03:32 id_rsa
-rw-r--r-- 1 tamayo tamayo  744 Sep  4 03:32 id_rsa.pub
-rw----- 1 tamayo tamayo 1120 Sep  4 03:23 known_hosts
-rw-r--r-- 1 tamayo tamayo  142 Sep  4 03:18 known_hosts.old

```

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.

```

tamayo@workstation:~$ ssh-copy-id tamayo@server1
/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 prompt
ed now it is to install the new keys
tamayo@server1's password:

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh 'tamayo@server1'"
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`

```
tamayo@workstation:~$ ssh-copy-id -i ~/.ssh/id_rsa tamayo@server1
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/tamayo/.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: WARNING: All keys were skipped because they already exist
on the remote system.
(if you think this is a mistake, you may want to use -f option)
```

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.

```
tamayo@workstation:~$ cd .ssh
tamayo@workstation:~/.ssh$ ls
authorized_keys id_rsa id_rsa.pub known_hosts known_hosts.old
tamayo@workstation:~/.ssh$
```

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?

```
tamayo@workstation:~/.ssh$ ssh tamayo@server1
Welcome to Ubuntu 23.10 (GNU/Linux 6.5.0-44-generic x86_64)

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

0 updates can be applied immediately.

Your Ubuntu release is not supported anymore.
For upgrade information, please visit:
http://www.ubuntu.com/releaseendoflife

New release '24.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Wed Sep 4 03:23:19 2024 from 127.0.0.1
```

No. Because it means that the SSH key is already set up for authentication on Server 1 and Server 2.

Reflections:

Answer the following:

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

By connecting through the ip addresses or networks of each device, the ssh- software, also known as the secure shell program, enables users to remotely host or manage another device. Users of the SSH-program or secure shell can also benefit from more robust encryption, which guards against disruption or interference from other malicious users.

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

A user can verify or test their installation of the public key by establishing a connection to the distant servers. The user can successfully install the public key to the distant servers if they are able to connect to them using the ssh command and establish a 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*

```
tamayo@workstation:~$ sudo apt install git
[sudo] password for tamayo:
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 0 not upgraded.
Need to get 4,718 kB of archives.
After this operation, 24.1 MB of additional disk space will be used.
Do you want to continue? [Y/n] yes
Get:1 http://ph.archive.ubuntu.com/ubuntu mantic/main amd64 liberror-perl all 0.
17029-2 [25.6 kB]
Get:2 http://ph.archive.ubuntu.com/ubuntu mantic-updates/main amd64 git-man all
1:2.40.1-1ubuntu1.1 [1,085 kB]
Get:3 http://ph.archive.ubuntu.com/ubuntu mantic-updates/main amd64 git amd64 1:
2.40.1-1ubuntu1.1 [3,607 kB]
```

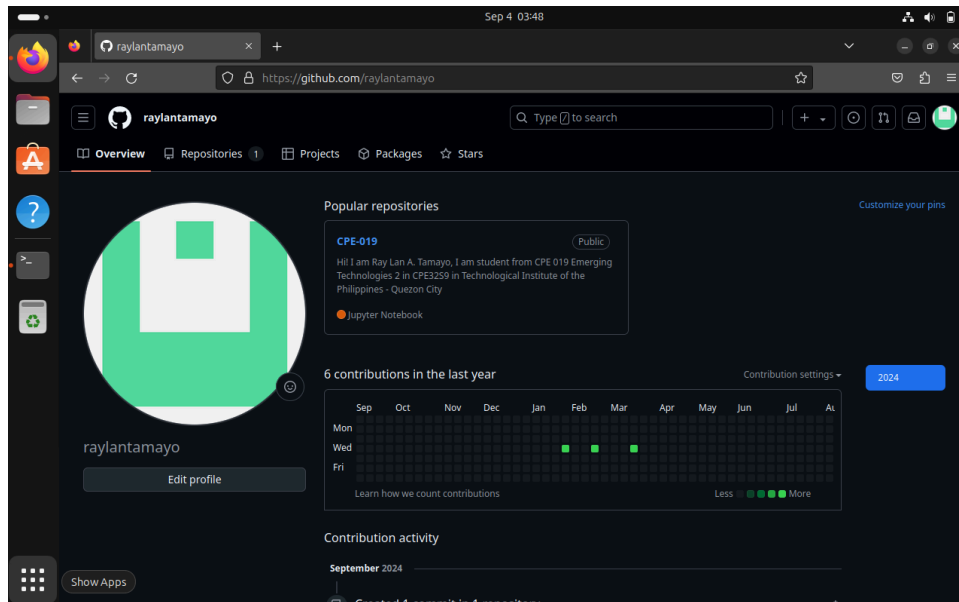
2. After the installation, issue the command *which git* again. The directory of git is usually installed in this location: *user/bin/git*.

```
tamayo@workstation:~$ 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.

```
tamayo@workstation:~$ git --version
git version 2.40.1
```

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 CPE232_yourname. Check Add a README file and click Create repository.

Create a new repository

A repository contains all project files, including the revision history. Already have a project elsewhere? [Import a repository](#).

Required fields are marked with an asterisk (*).

Owner *

 raylantaiamayo ▾

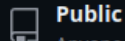
Repository name *

CPE232_Tamayo

✔ CPE232_Tamayo is available.

Great repository names are short and memorable. Need inspiration? How about **stunn**

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](#).

- 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.



raylantaiamayo (raylantaiamayo)

Your personal account



Public profile



Account



Appearance



Accessibility



Notifications

Access



Billing and plans ▾



Emails



Password and authentication



Sessions



SSH and GPG keys

Add new SSH Key

Title

CPE232

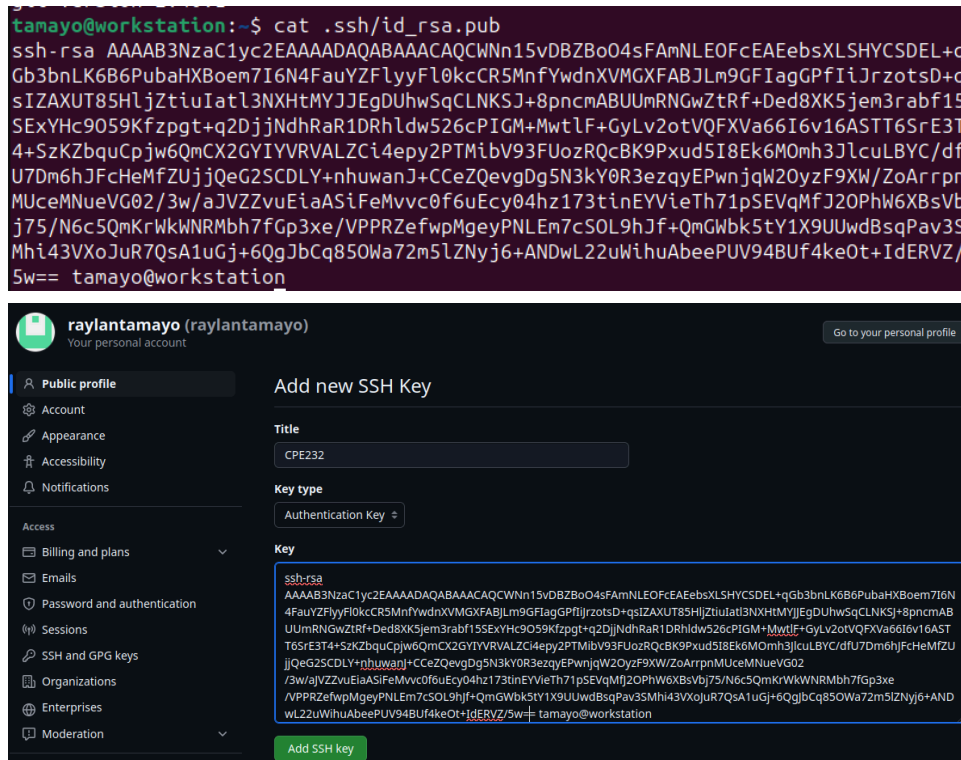
Key type

Authentication Key ▾

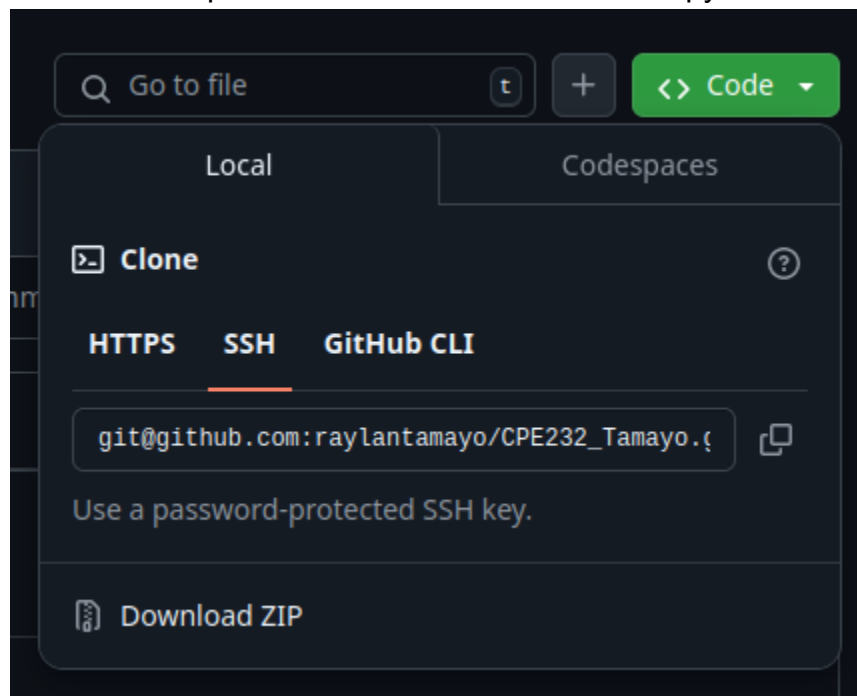
Key

Begins with 'ssh-rsa', 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384', 'ecdsa-sha2-nistp521', 'ssh-ed25519', 'sk-ssh-ed25519@openssh.com', or 'sk-ssh-ed25519@openssh.com'

- 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/CPE232_yourname.git`. When prompted to continue connecting, type yes and press enter.

```
tamayo@workstation:~$ git clone git@github.com:raylantamayo/CPE232_Tamayo.git
Cloning into 'CPE232_Tamayo'...
The authenticity of host 'github.com (20.205.243.166)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvV6TuJJhbpZisF/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.
```

- 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`.

```
tamayo@workstation:~$ ls
CPE232_Tamayo  Documents  Music      Public  Templates
Desktop        Downloads  Pictures   snap    Videos
```

- 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`

```
tamayo@workstation:~$ git config --global user.name "Tamayo"
tamayo@workstation:~$ git config --global user.email raylantamayo@gmail.com
tamayo@workstation:~$ cat ~/.gitconfig
[user]
    name = Tamayo
    email = raylantamayo@gmail.com
```

- 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.

```
GNU nano 7.2                                README.md *
#CPE232_Tamayo

git@github.com:raylantamayo/CPE232_Tamayo.git
```

- 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?

```
tamayo@workstation:~/CPE232_Tamayo$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```

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

```
nothing to commit, working tree clean
tamayo@workstation:~/CPE232_Tamayo$ git add README.md
```

- 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.

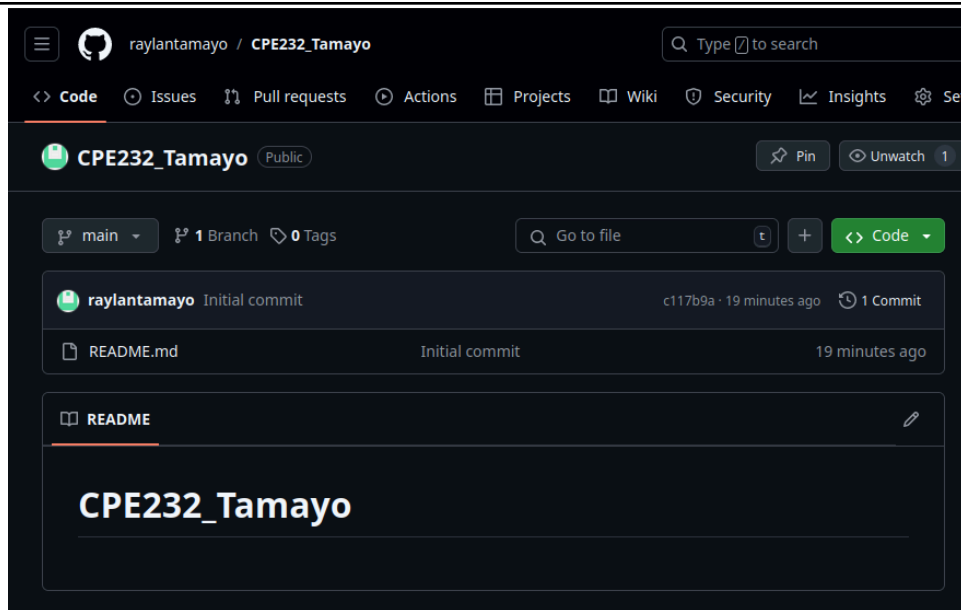
```
nothing to commit, working tree clean
tamayo@workstation:~/CPE232_Tamayo$ git commit -m "Activity 2 CPE212 CPE31521"
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```

- 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*.

```
tamayo@workstation:~/CPE232_Tamayo$ git push origin main
Everything up-to-date
tamayo@workstation:~/CPE232_Tamayo$
```

- 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 can investigate and learn the fundamentals of remotely hosting or accessing other stations. With a rudimentary understanding of remote hosting, we can discover how simple it is to connect to github using our own terminal. Using several commands that let us create, amend, and apply changes to the repository, we are also able to learn how to alter the local repository on github.

4. How important is the inventory file?

The inventory file gives the user access to a list or infrastructure that enables the administrator or primary user to manage the hosts and files in a flexible way.

Conclusions/Learnings:

In this exercise, we'll examine how keys offer a more secure and encrypted authentication for the user, making them vital or important to remote access or hosting. We gained knowledge on how to generate our own keys and apply them for easier login authentication on other terminals, stations, or servers. Through our github accounts, we were able to discover the various commands we could use to access a local repository. We learned the importance of SSH and how it lets administrators or users control inventories.