1. **Name: Maxine Audrey D. PulaoDate Performed: August 23, 2022Course/Section: CPE31S2Date Submitted: August 23, 2022Instructor: Dr. Jonathan TaylarSemester and SY: 1st year (2022-2023Activity 2: SSH Key-Based Authentication and Setting up GitObjectives:**

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.

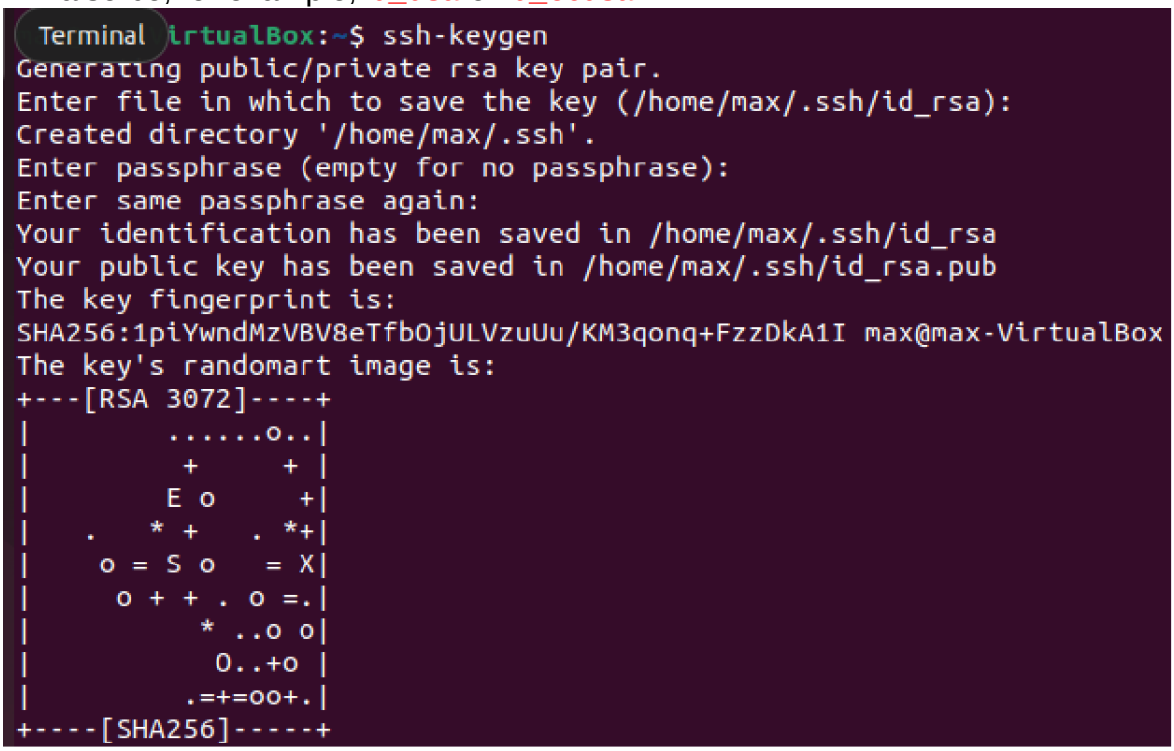
I1 SSH Keys and Public Key Authentication

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

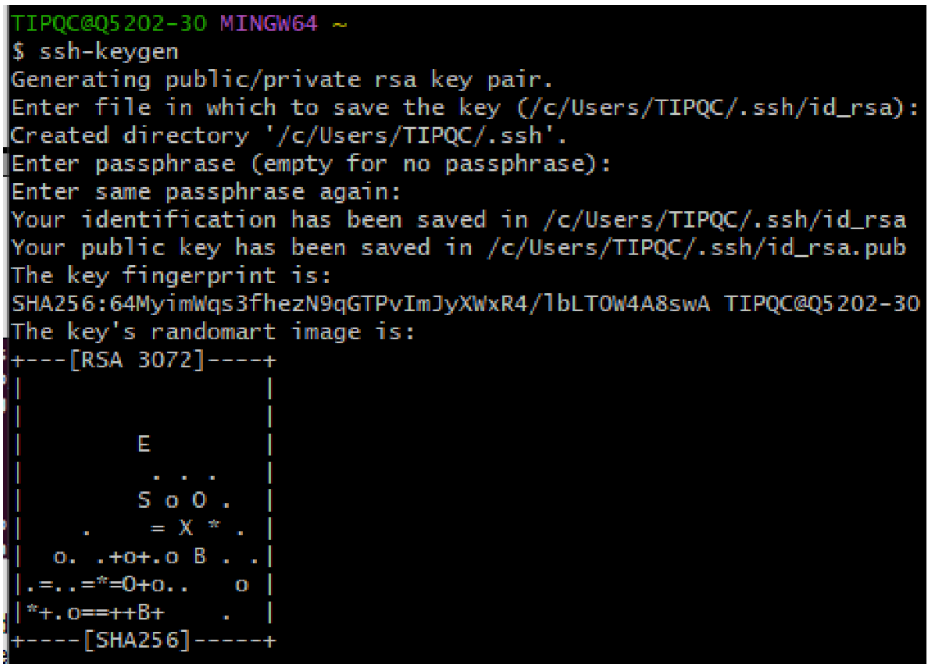
SSH introduced public key authentication as a more secure alternative to the older .rhosts authentication. It improved security by avoiding the need to have password stored in files and eliminated the possibility of a compromised server stealing the user's password.

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*.



1.



1.

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

```
max@max-VirtualBox:~$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/max/.ssh/id_rsa):
/home/max/.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/max/.ssh/id_rsa
Your public key has been saved in /home/max/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:OHzv8qOfAmEhKM1leC4vNODZUoX/BEAWOUndvkFecxY max@max-VirtualBox
The key's randomart image is:
+---[RSA 4096]----+
| ooX@..    E.    |
|o *Boo.o o o     |
|.o+oo.=.. +      |
| ++..oo*         |
| ..+ .*.S        |
|  . . .= .       |
|   .   .  .      |
|        o...     |
|          *      |
Show Applications  --+
```

1.

```
TIPQC@Q5202-30 MINGW64 ~
$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/TIPQC/.ssh/id_rsa):
/c/Users/TIPQC/.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 /c/Users/TIPQC/.ssh/id_rsa
Your public key has been saved in /c/Users/TIPQC/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:PQxUumw2B485t7rgUXE6BtyDpOsTE53ZXwOydYbO/ic TIPQC@Q5202-30
The key's randomart image is:
+---[RSA 4096]----+
|      ...=. o.o+o|
|     . =+..  ooo.|
|      * O .. +   |
|       B #  . o  |
|      + S B  .   |
|     . B * o  .  |
|      +   .    . |
|     . o .    E o|
|      . o.     ..|
+----[SHA256]-----+
```

2. 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.

1. 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.

```
max@max-VirtualBox:~$ ls -ls .ssh
total 8
4 -rw------- 1 max max 3381 Aug 23 08:59 id_rsa
4 -rw-r--r-- 1 max max  744 Aug 23 08:59 id_rsa.pu
```

Task 2: Copying the Public Key to the remote servers

```
TIPQC@Q5202-30 MINGW64 ~
$ ls -ls .ssh
total 8
4 -rw-r--r-- 1 TIPQC 197121 3381 Aug 23 09:14 id_rsa
4 -rw-r--r-- 1 TIPQC 197121  740 Aug 23 09:14 id_rsa.pub
```

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.

```
max@max-VirtualBox:~$ ssh-copy-id
Usage: /usr/bin/ssh-copy-id [-h|-?|-f|-n|-s] [-i [identity_file]] [-p port] [-F
 alternative ssh_config file] [[-o <ssh -o options>] ...] [user@]hostname
        -f: force mode -- copy keys without trying to check if they are already
 installed
        -n: dry run    -- no keys are actually copied
        -s: use sftp   -- use sftp instead of executing remote-commands. Can be
 useful if the remote only allows sftp
        -h|-?: print this help
```

```
TIPQC@Q5202-30 MINGW64 ~
$ ssh-copy-id
Usage: /usr/bin/ssh-copy-id [-h|-?|-f|-n|-s] [-i [identity_file]] [-p port] [-F alternative ssh_
config file] [[-o <ssh -o options>] ...] [user@]hostname
       -f: force mode -- copy keys without trying to check if they are already installed
       -n: dry run    -- no keys are actually copied
       -s: use sftp   -- use sftp instead of executing remote-commands. Can be useful if the re
mote only allows sftp
       -h|-?: print this help
```

1.

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

```
max@max-VirtualBox:~$ ssh-copy-id max@192.168.56.101
The authenticity of host '192.168.56.101 (192.168.56.101)' can't be established
.
ED25519 key fingerprint is SHA256:cbzhYiKcqyIbl4IpmIbRaAksuBxd4bxkk5fcX6J2tGw.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? 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
max@192.168.56.101's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'max@192.168.56.101'"
and check to make sure that only the key(s) you wanted were added.
```

1.

2. 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.

3. 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?

Server 1

```
max@max-VirtualBox:~$ ssh max@192.168.56.101
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-46-generic x86_64)

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

0 updates can be applied immediately.

Last login: Tue Aug 16 12:24:17 2022 from 192.168.56.103
```

server 2

```
max@max-VirtualBox:~$ ssh max@192.168.56.102
The authenticity of host '192.168.56.102 (192.168.56.102)' can't be established
.
ED25519 key fingerprint is SHA256:FTkWBm6/PhyEE+dpu47qPD7M6MqXZdSoOf/dGTYCWXA.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.56.102' (ED25519) to the list of known host
s.
max@192.168.56.102's password:
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-46-generic x86_64)

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

0 updates can be applied immediately.

Last login: Tue Aug 16 12:26:33 2022 from 192.168.56.103
max@max-Server2-VirtualBox:~$
```

Reflections:

Answer the following:

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

- The SSH program is a secured protocol and connects remotely in Linux. The SSH program is a text-based interface where it uses the remote shell where it provides a safe way of executing commands.

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

- After generating and corresponding the public key to the server, the server will automatically show that it is approved. With that, it allows anyone to access who proves ownership corresponding to the private key.

3 To verify the if we have previously installed the servers, we use the commands cat ~/.ssh/id_rsa.pub

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.*

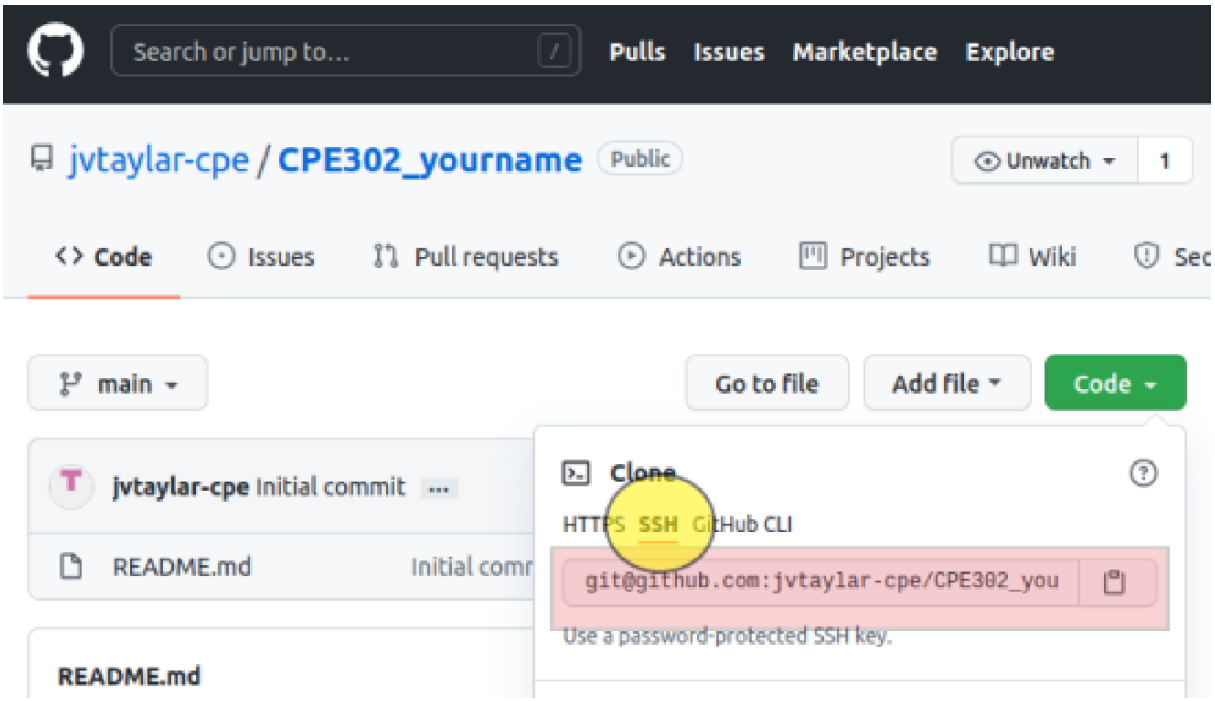
```
max@max-Server2-VirtualBox:~$ which git
/usr/bin/git
```

1.
2. The version of git installed in your device is the latest. Try issuing the command *git --version* to know the version installed.

```
max@max-Server2-VirtualBox:~$ git --version
git version 2.34.1
```

1.
2. Using the browser in the local machine, go to www.github.com.
3. Sign up in case you don't have an account yet. Otherwise, login to your GitHub account.
4. Create a new repository and name it as CPE232_yourname. Check Add a README file and click Create repository.
5. 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.
6. 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.

7. 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.



a 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.

```
max@max-VirtualBox:~$ git clone git@github.com:MaxinePulao/CPE232_PULAO.git
Cloning into 'CPE232_PULAO'...
The authenticity of host 'github.com (140.82.114.3)' 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
Receiving objects: 100% (3/3), done.
```

a

b 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.

```
max@max-VirtualBox:~$ ls
CPE232_PULAO  Documents  Music     Public  Templates
Desktop       Downloads  Pictures  snap    Videos
```

a

b 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*

```
max@max-VirtualBox:~$ git config --global user.name MAXINE
max@max-VirtualBox:~$ git config --global user.email qmadpulao@tip.edu.ph
max@max-VirtualBox:~$ cat ~/.gitconfig
[user]
        name = MAXINE
        email = qmadpulao@tip.edu.ph
```

-

a 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 6.2                          README.md
# CPE232_PULAO

Wow Hotdog!
```

```
max@max-VirtualBox:~$ cd CPE232_PULAO
max@max-VirtualBox:~/CPE232_PULAO$ ls
README.md
max@max-VirtualBox:~/CPE232_PULAO$ nano README.md
max@max-VirtualBox:~/CPE232_PULAO$
```

a

b 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? Gi

```
max@max-VirtualBox:~/CPE232_PULAO$ git status
On branch main
Your branch is up to date with 'origin/main'.

LibreOffice Writer
ed for commit:
<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")
max@max-VirtualBox:~/CPE232_PULAO$
```

a

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

c 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.

```
max@max-VirtualBox:~/CPE232_PULAO$ git add README.md
max@max-VirtualBox:~/CPE232_PULAO$ git commit -m "Wow Hotdog!"
[ c9c7402] Wow Hotdog!
 1 file changed, 3 insertions(+), 1 deletion(-)
max@max-VirtualBox:~/CPE232_PULAO$
```

Trash

a

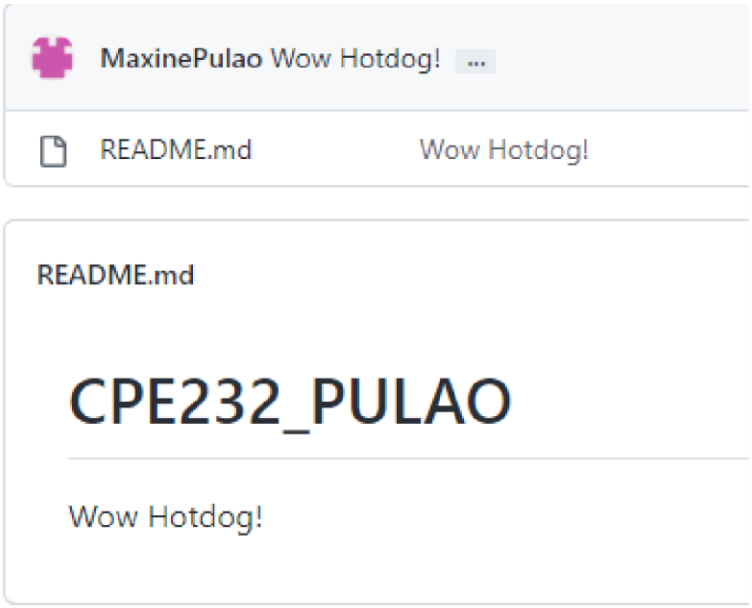
b Use the command *git push <remote><branch>* to upload the local repository content to GitHub repository. Pushing means to transfer commits from the local repository to the remote repository. As an example, you may issue *git push origin main*.

```
max@max-VirtualBox:~/CPE232_PULAO$ git push origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Writing objects: 100% (3/3), 275 bytes | 275.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:MaxinePulao/CPE232_PULAO.git
   2d3499d..c9c7402  main -> main
max@max-VirtualBox:~/CPE232_PULAO$
```

a

b 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.

c



a

Reflections:

Answer the following:

1 What sort of things have we so far done to the remote servers using ansible commands?

- By using the ansible commands, this configures, manages and facilitates tasks to maintain the remote servers. Through that, the user can control hundreds of systems from a central location.

2 How important is the inventory file?

- It provides an orderly way of managing and planning files.

Conclusions/Learnings:

- **After performing the activity, I have successfully introduced how the SSH key works and how it is significant to use it in servers. By employing a safe management in a system, we can make a system safer and more orderly to use. Now that I have seen how useful the SSH keys are, I can use this as a key skill to in my future endeavors as a computer engineer and a system administrator.**