

<b>Name:</b> John Renzo L. Mendoza	<b>Date Performed:</b> August 26, 2023
<b>Course/Section:</b> CPE31S5	<b>Date Submitted:</b> August 29, 2023
<b>Instructor:</b> Engr. Roman Richard	<b>Semester and SY:</b> First Semester, 2023- 2024

## 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 require 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

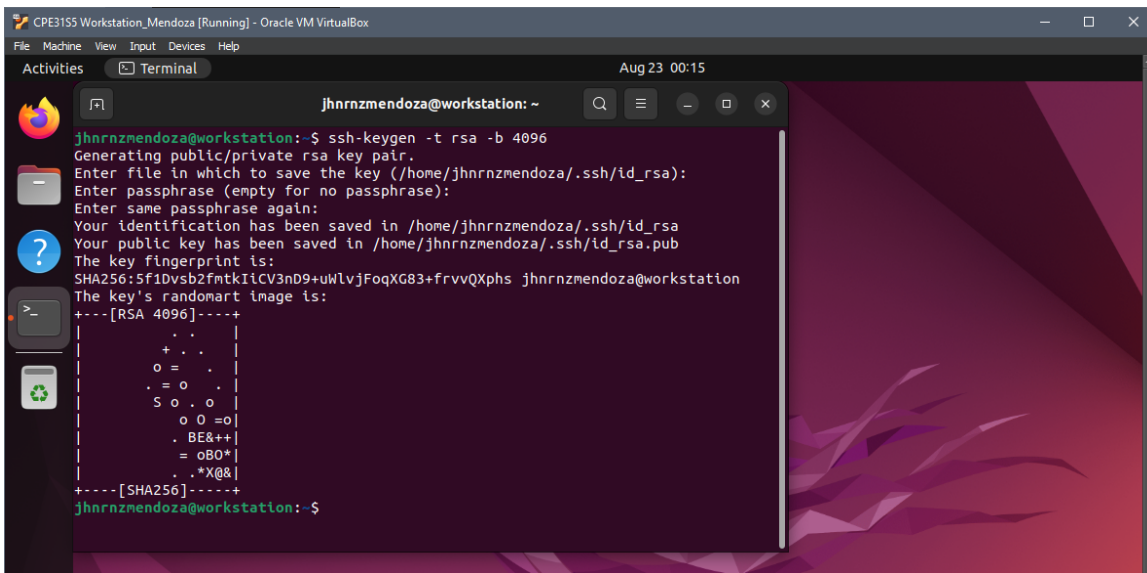
The SSH protocol uses public key cryptography for authenticating hosts and users. The authentication keys, called SSH keys, are created using the keygen program.

SSH introduced public key authentication as a more secure alternative to the older .rhosts authentication. It improved security by avoiding the need to have passwords 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.

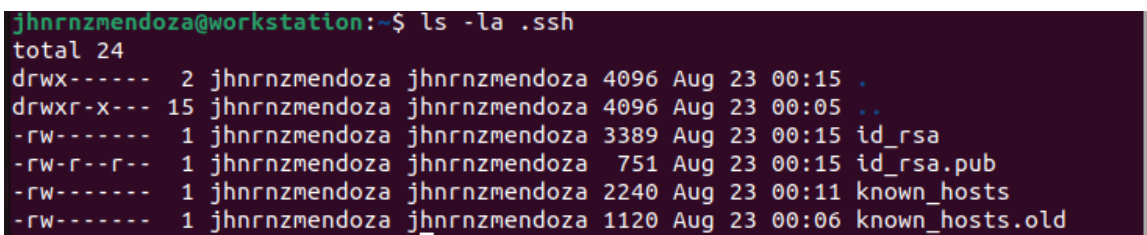
## Task 1: Create an SSH Key Pair for User Authentication

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.



```
CPE3155 Workstation_Mendoza [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Aug 23 00:15
jhnrmendoza@workstation: ~
jhnrmendoza@workstation:~$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/jhnrmendoza/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/jhnrmendoza/.ssh/id_rsa
Your public key has been saved in /home/jhnrmendoza/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:5f1Dvsb2fntkiCV3nD9+uWlvjFoqXG83+frvvQXphs jhnrmendoza@workstation
The key's randomart image is:
+---[RSA 4096]-----+
  .
  + .
  o = .
  . = o .
  S o . o
    o o =o|
  . BE&+|
  = oB0*|
  . .*X@&|
+---[SHA256]-----+
jhnrmendoza@workstation:~$
```

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`.

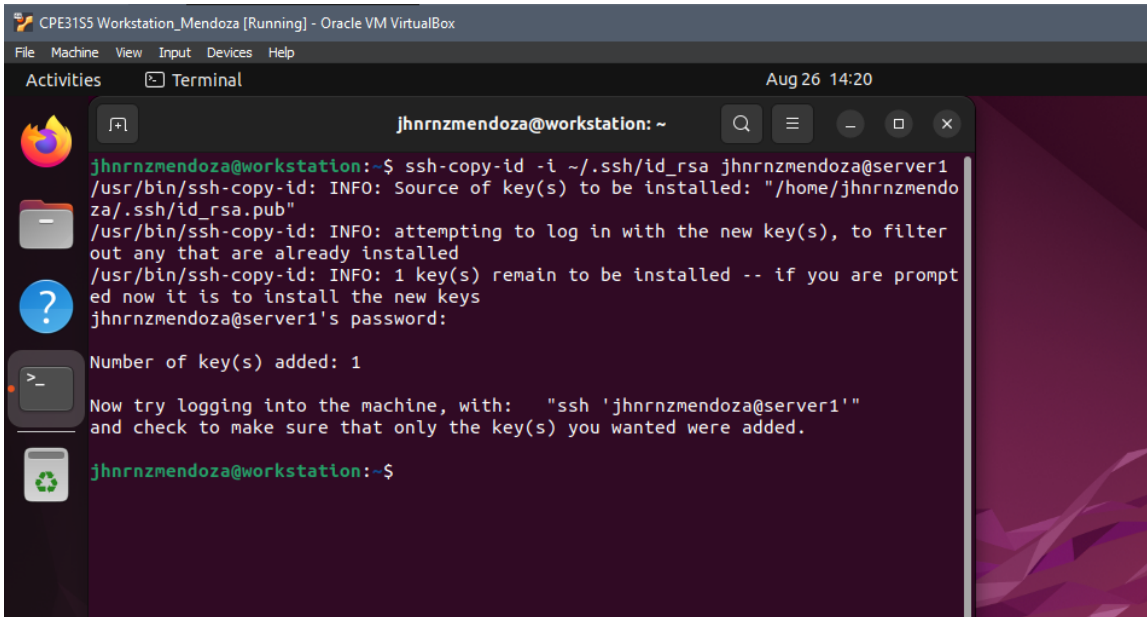


```
jhnrmendoza@workstation:~$ ls -la .ssh
total 24
drwx----- 2 jhnrmendoza jhnrmendoza 4096 Aug 23 00:15 .
drwxr-x--- 15 jhnrmendoza jhnrmendoza 4096 Aug 23 00:05 ..
-rw----- 1 jhnrmendoza jhnrmendoza 3389 Aug 23 00:15 id_rsa
-rw-r--r-- 1 jhnrmendoza jhnrmendoza 751 Aug 23 00:15 id_rsa.pub
-rw----- 1 jhnrmendoza jhnrmendoza 2240 Aug 23 00:11 known_hosts
-rw----- 1 jhnrmendoza jhnrmendoza 1120 Aug 23 00:06 known_hosts.old
```

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

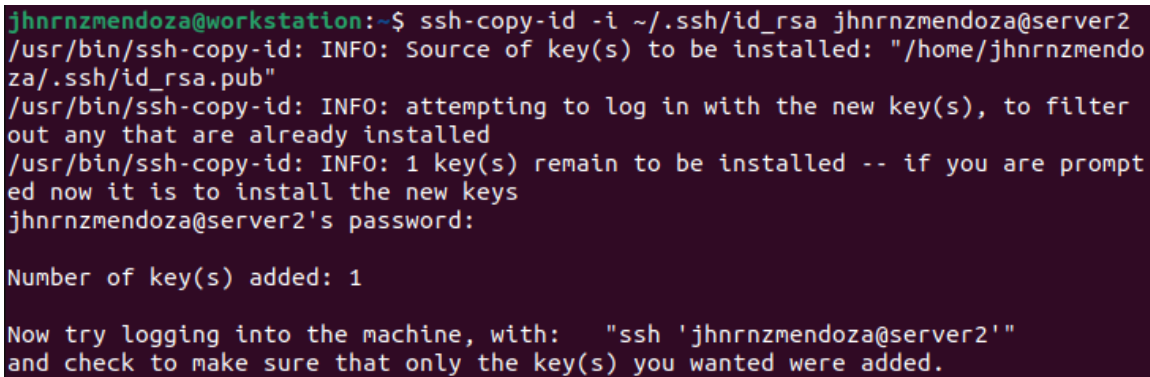
1. To use public key authentication, the public key must be copied to a server and installed in an *authorized\_keys* file. This can be conveniently done using the *ssh-copy-id* tool.
2. Issue the command similar to this: *ssh-copy-id -i ~/.ssh/id\_rsa user@host*

### Server 1



```
CPE31S5 Workstation_Mendoza [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal
Aug 26 14:20
jhnrmendoza@workstation: ~
jhnrmendoza@workstation:~$ ssh-copy-id -i ~/.ssh/id_rsa jhnrmendoza@server1
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/jhnrmendoza/.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
jhnrmendoza@server1's password:
Number of key(s) added: 1
Now try logging into the machine, with: "ssh 'jhnrmendoza@server1'"
and check to make sure that only the key(s) you wanted were added.
jhnrmendoza@workstation:~$
```

### Server 2

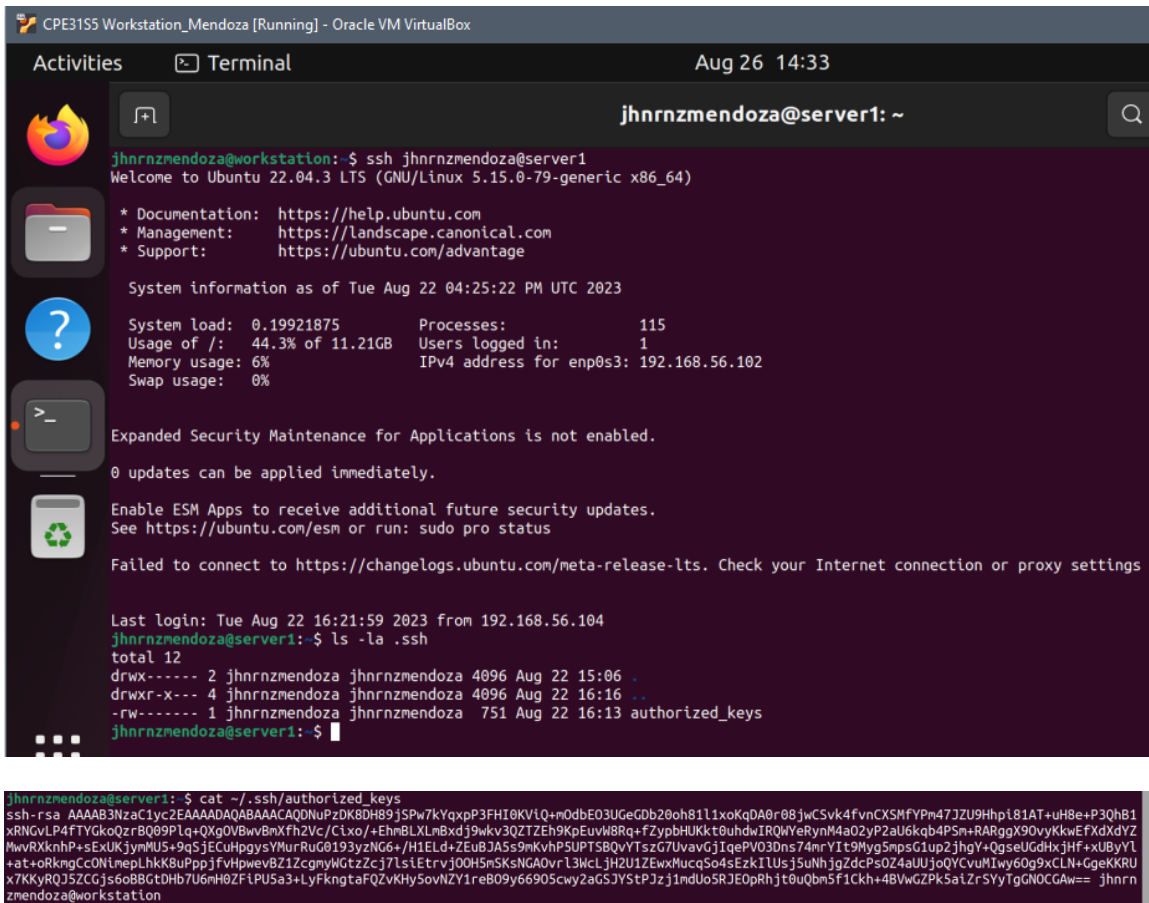


```
jhnrmendoza@workstation:~$ ssh-copy-id -i ~/.ssh/id_rsa jhnrmendoza@server2
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/jhnrmendoza/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
jhnrmendoza@server2's password:
Number of key(s) added: 1
Now try logging into the machine, with: "ssh 'jhnrmendoza@server2'"
and check to make sure that only the key(s) you wanted were added.
```

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.

## SSH on Server 1



```
CPE3ISS Workstation_Mendoza [Running] - Oracle VM VirtualBox
Activities Terminal Aug 26 14:33
jhnrmendoza@server1: ~

jhnrmendoza@workstation:~$ ssh jhnrmendoza@server1
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-79-generic x86_64)

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

System information as of Tue Aug 22 04:25:22 PM UTC 2023

System load:  0.19921875   Processes:           115
Usage of /:   44.3% of 11.21GB   Users logged in:     1
Memory usage: 6%           IPv4 address for enp0s3: 192.168.56.102
Swap usage:  0%

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

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Last login: Tue Aug 22 16:21:59 2023 from 192.168.56.104
jhnrmendoza@server1:~$ ls -la .ssh
total 12
drwx----- 2 jhnrmendoza jhnrmendoza 4096 Aug 22 15:06 .
drwxr-x--- 4 jhnrmendoza jhnrmendoza 4096 Aug 22 16:16 ..
-rw----- 1 jhnrmendoza jhnrmendoza 751 Aug 22 16:13 authorized_keys

jhnrmendoza@server1:~$
```

```
jhnrmendoza@server1:~$ cat ~/.ssh/authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCNuPzDK8DH89j5Pw7kYqxpP3FHI8KVlQ+n0dbE03UGeGDb20oh811xoKqDA0r08jwCSvk4fVnCXSMfYpM47JZU9Hhpt81AT+uH8e+P3Qh81
xRNGvLP4fTYGkoQzrBQ99Plq+OXg0VBwvBmXfh2Vc/CiXo/+EhmBLXLnBxdj9wkv3QZTZeh9KpEuvW8Rq+fZypbHUKkt0uhdwIRQWYeRynM4a02yP2aU6kqb4P5n+RARggX90VvKkwEFXdxYz
MwvRXknHP+sExUKjymMUS+9q5jECuHpgysVMurRuG0193yzNG6+/H1ELD+ZEuBJA5s9mKvHP5UPTSBQvYTszG7UvavGjIqePV03Dns74mrYIT9Myg5mpsG1up2jhgY+QgseUGdHxjHf+xUByYl
+at+oRkngCc0NimepLhK8uPppjFvHpwvBZ1ZcgyMGtzZcj7LsiEtrvj00H5mSKsNGA0vrl3WcLjH2U1ZExwMucqSo4sEzk1LUjs5uNhjgZdcPsOZ4aUjjoQYCVuMlwy60g9xCLN+GgeKKRU
x7KKyRQJ5ZCGjs6oBBGtDhb7U6nH0ZF1PUSa3+LyFkngtaFQZvKHySovNZY1reB09y66905cwy2aGSJYStPjzj1ndUoSRJE0pRhjt0uQbm5f1Ckh+4BvWgZPk5aiZrSYyTgNOCGAw== jhnrm
endoza@workstation
```

## SSH on Server 2

```
CPE31S5 Workstation_Mendoza [Running] - Oracle VM VirtualBox
Activities Terminal Aug 26 14:34
jhnrmendoza@server2: ~
jhnrmendoza@workstation:~$ ssh jhnrmendoza@server2
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-79-generic x86_64)

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

System information as of Tue Aug 22 04:26:12 PM UTC 2023

System load: 0.00146484375   Processes:            116
Usage of /:  44.3% of 11.21GB Users logged in:             1
Memory usage: 6%            IPv4 address for enp0s3: 192.168.56.103
Swap usage:  0%

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

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Last login: Tue Aug 22 16:11:40 2023 from 192.168.56.104
jhnrmendoza@server2:~$ ls -la .ssh
total 12
drwx----- 2 jhnrmendoza jhnrmendoza 4096 Aug 22 15:06 .
drwxr-x--- 4 jhnrmendoza jhnrmendoza 4096 Aug 22 15:18 ..
-rw----- 1 jhnrmendoza jhnrmendoza 751 Aug 22 16:14 authorized_keys
jhnrmendoza@server2:~$

jhnrmendoza@server2:~$ cat ~/.ssh/authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCNuPzDK8DH89jSPw7kYqxpP3FHI0KVlQ+n0dbE03UGeGDb20oh811xoKqDA0r08jwCSvk4fVnCXSHfYpM47JZU9Hpl81AT+uH8e+P3QhB1
xRNGvLP4fTYGkoQzrBQ09PLq+QXg0VBwvBnXfh2Vc/Cix0/+EhmBLXLnBxdj9wkv3QZTZEh9KpEuvWBRq+fZypbHUKkt0uhdwIRQMYeRynM4a02yP2aU6kqb4PSn+RARggX90vyKkwEfXdxYZ
MwvRXknhP+sExUKjymMUS+9qSjECuHpgysYMurRuG0193yzNG6+/H1ELD+ZEuBJA5s9nKvHP5UPTSBQvYTszG7UvavGjIqePVO3Dns74mrYIt9Myg5mpsG1up2jhgY+QgseUGdHxjHf+xUByYL
+at+oRkmgCc0NimepLhK8uPppjfvHpwvBZ1ZcgnyWGtzZcj7LsIEtrvj00H5mSKsNGA0vrL3McLjH2U1ZEwxMucqSo4sEzkILUsj5uNhjgZdcPsOZ4aUUjoQYCVuMIwy6og9xCLN+GgeKKRU
x7KKyRQJ5ZCGjs6oBBGtdHb7U6mH0ZFtPU5a3+LyFkngtaFQZvKHySovNZY1reB09y66905cwy2aGSJYStPJzj1mdUo5R3E0pRhjt0uQbm5f1Ckh+4BWwGZPkSaIZr5YyTgGNOCGAw== jhnrm
endoza@workstation
```

What did you notice? Did the connection ask for a password? If not, why?

The remote access to the servers (1 and 2) did not ask for a password / passphrase as I did not enter any password / passphrase when initializing the private key. Moreover, the SSH command to the servers directly allowed me to access it as no password / passphrase is required to access them.

**Additional:** Add a passphrase on server 1 for proof.

Generating key, now with a passphrase.

```
jhnrmendoza@workstation:~$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/jhnrmendoza/.ssh/id_rsa):
/home/jhnrmendoza/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Passphrases do not match. Try again.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/jhnrmendoza/.ssh/id_rsa
Your public key has been saved in /home/jhnrmendoza/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:1ksj0+KJoMKNHBfbb0SHHie98+rL+Es2VVNwbyqP7LE jhnrmendoza@workstation
The key's randomart image is:
+---[RSA 4096]-----+
|      .o.  ..o |
|      ..+.  o  |
|      .  + .. o o|
|      + . .= . .o|
|      . o.. .S B.. E|
|o =. .o+ *.+ = |
|+. . .oo+o o . |
|      .  =... . |
|      .o*o.o    |
+---[SHA256]-----+
```

```
jhnrmendoza@workstation:~$ ls -la .ssh
total 28
drwx----- 3 jhnrmendoza jhnrmendoza 4096 Aug 26 14:21 .
drwxr-x--- 15 jhnrmendoza jhnrmendoza 4096 Aug 23 00:05 ..
-rw----- 1 jhnrmendoza jhnrmendoza 3434 Aug 26 14:43 id_rsa
-rw-r--r-- 1 jhnrmendoza jhnrmendoza 751 Aug 26 14:43 id_rsa.pub
-rw----- 1 jhnrmendoza jhnrmendoza 2240 Aug 23 00:11 known_hosts
-rw----- 1 jhnrmendoza jhnrmendoza 1120 Aug 23 00:06 known_hosts.old
drwx----- 2 jhnrmendoza jhnrmendoza 4096 Aug 26 14:18 ssh-copy-id.dt7deD4Cvv
```

Copying key file to server1

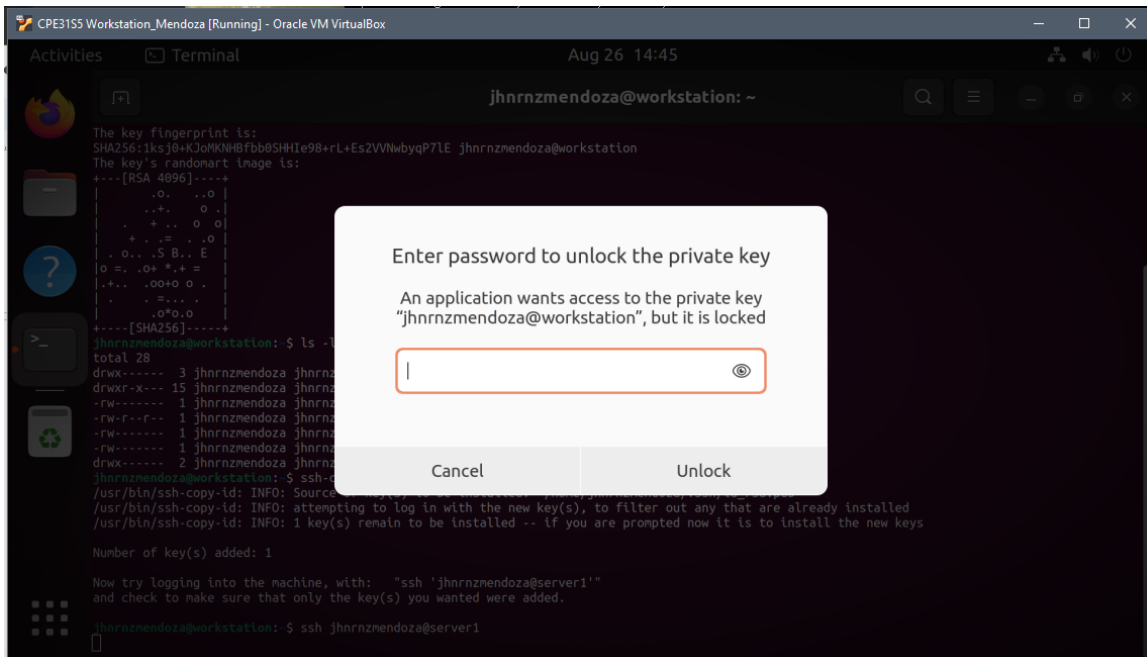
```
jhnrmendoza@workstation:~$ ssh-copy-id -i ~/.ssh/id_rsa jhnrmendoza@server1
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/jhnrmendoza/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys

Number of key(s) added: 1

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



Remotely accessing Server 1, now requiring the passphrase.



## Reflections:

Answer the following:

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

The Secure Socket Shell allows the system administrator to remotely configure a device such as a server. In this laboratory activity, the students were tasked to generate a ssh-key which will be required when remotely accessing the server. With the use of this ssh-key, it makes the access secure since only those who know the password or passphrase will be able to access the remote server.

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

By observing the files on the servers (1 and 2), I have observed that the file `~/.ssh/authorized_keys` is present whenever we have finished copying the file containing the private key from the local machine to the server.

```
jhnrmendoza@server2:~$ cat ~/.ssh/authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDU4pRC/TyDcZAMUpjuYm1m7Cvdqazasx1BYqBb7B53
bTAqeAk8fC1i30owq9Fgrq9qr7PH0/+be+auVJfDYmogDg5CPa+059icDtWXMswbGA0+ecTkAPXsf1B/
ByZ5A0v0Sof+a6JhGaINBh3HThM6eopSSZwdqjSc9XCcRuEL9bUzS0giM0Fj7JLryqpECZqGgxy8MqSX
RL6ncF+sSKNwslz0uY6Ykm42dsZRVit5skIIATyBaILIHfv3E0S6No8f9YIyZy0+rk6BDc0lRLz5XT86
WkNewQZ98becjfuDvzprmpTldfdgEH32CZSzdWCNqAKodYBj4QKYndVSnofTHw6VbFLS4Dvq3bbN5td1
+dnJY2A8N93mm92Ls10b6DZy3godhLB2TFtapPfPq/7A/hq3fRv0Df3ilIsuHpGa0XIqk0VEuWpgy9X3
OawS4r19dsijzGoTzKV9jxL9YuCxmwk3F8pGTEDNq1nN5AyUkh1vGKPTaSoLwbDSSN/y1PvVnx60qoNL
ab2qXUkWyLTyr/wOzgVZGhkQEGmQpIubSpKsQTVqpUoP4UTVLixMxMJ8+XiQHvQni6Qa+VipwEGC1nw
UYtfdLCHbAU9HIMYGrPFw6TD02BN9CA+baJcLCSOyOuphbiw1sSCT2DW21sVKkjmdYXC+hE87KfLorf
Pw== jhnrmendoza@workstation
```

By also concatenating the contents of the file, it also shows a relation with the local machine denoted by `jhnrmendoza@workstation`.

## 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

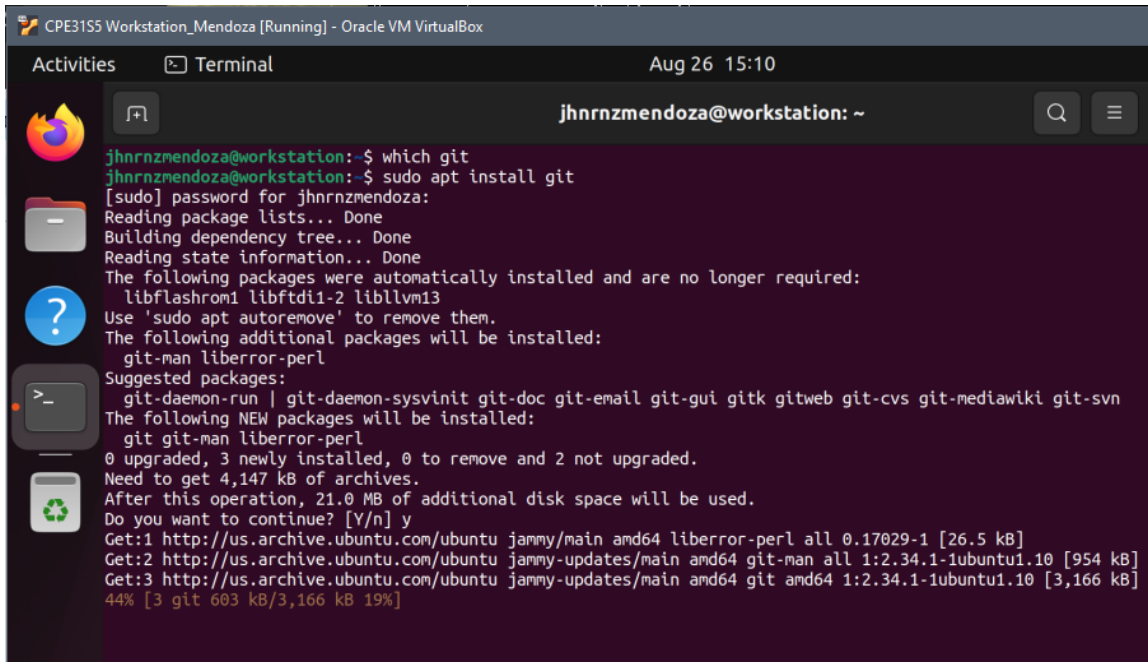
At the heart of GitHub is an open-source version control system (VCS) called Git. Git is responsible for everything GitHub-related that happens locally on your computer. To use Git on the command line, you'll need to download, install, and configure Git on your computer. You can also install GitHub CLI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



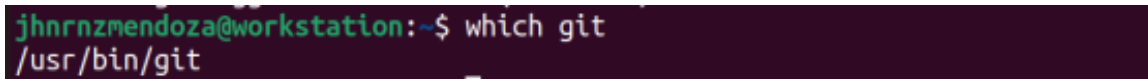
### Task 3: Set up the Git Repository

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*

A screenshot of a terminal window titled "CPE3155 Workstation\_Mendoza [Running] - Oracle VM VirtualBox". The terminal shows the user "jhnrmendoza@workstation" running the command "which git", which returns no output. Then, the user runs "sudo apt install git". The terminal displays the password prompt, package lists, dependency tree, and state information. It lists packages to be installed (git, git-man, liberror-perl) and additional packages (git-daemon-run, git-daemon-sysvinit, git-doc, git-email, git-gui, gitk, gitweb, git-cvs, git-mediawiki, git-svn). It shows the disk space requirements and asks for confirmation to continue. The user responds with 'y'. The terminal then shows the progress of the installation, including the download of git and its dependencies.

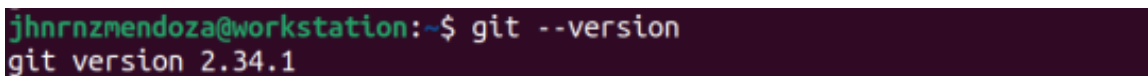
```
jhnrmendoza@workstation:~$ which git
jhnrmendoza@workstation:~$ sudo apt install git
[sudo] password for jhnrmendoza:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libflashrom1 libftdi1-2 libllvm13
Use 'sudo apt autoremove' to remove them.
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 2 not upgraded.
Need to get 4,147 kB of archives.
After this operation, 21.0 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us.archive.ubuntu.com/ubuntu jammy/main amd64 liberror-perl all 0.17029-1 [26.5 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 git-man all 1:2.34.1-1ubuntu1.10 [954 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 git amd64 1:2.34.1-1ubuntu1.10 [3,166 kB]
44% [3 git 603 kB/3,166 kB 19%]
```

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

A screenshot of a terminal window showing the user "jhnrmendoza@workstation" running the command "which git". The output is "/usr/bin/git".

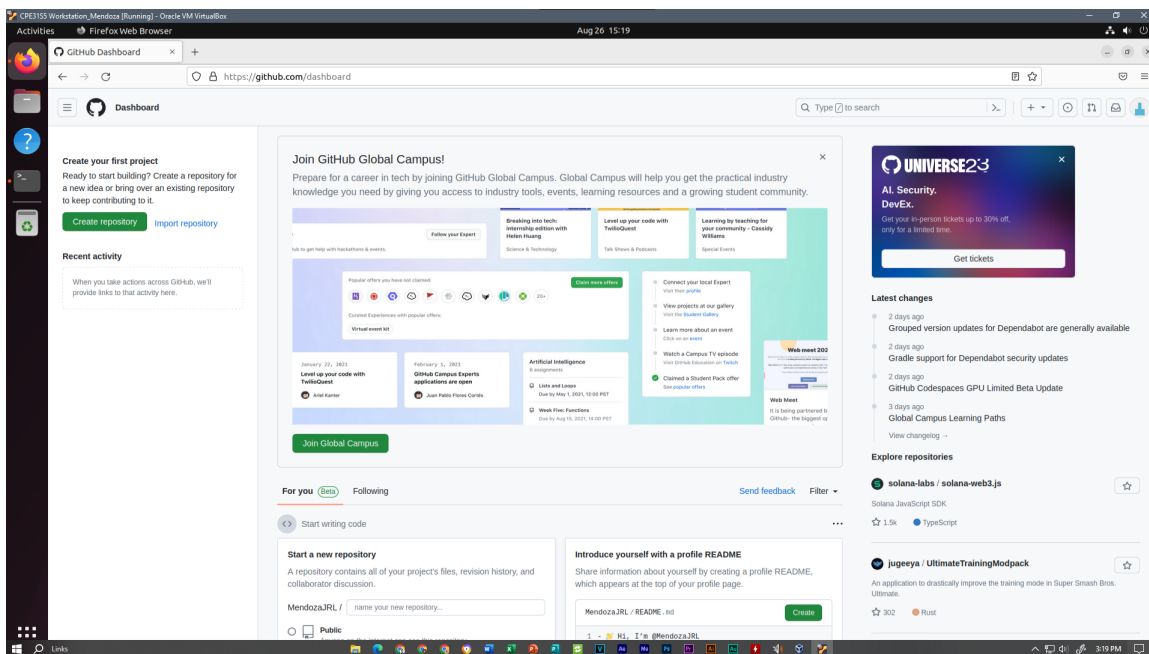
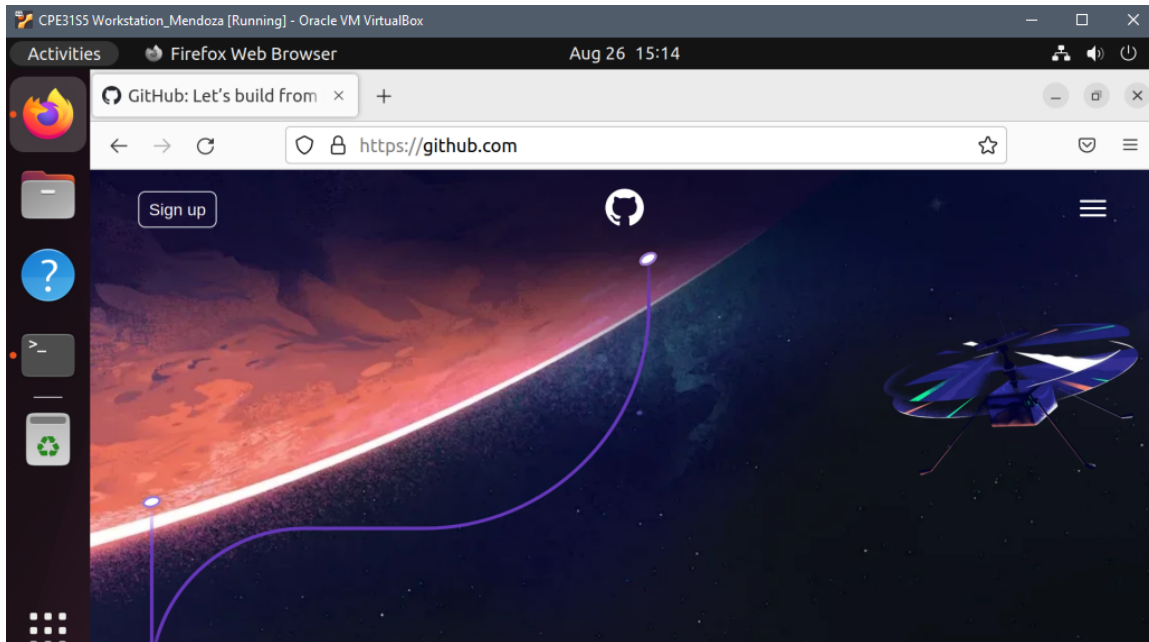
```
jhnrmendoza@workstation:~$ which git
/usr/bin/git
```

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

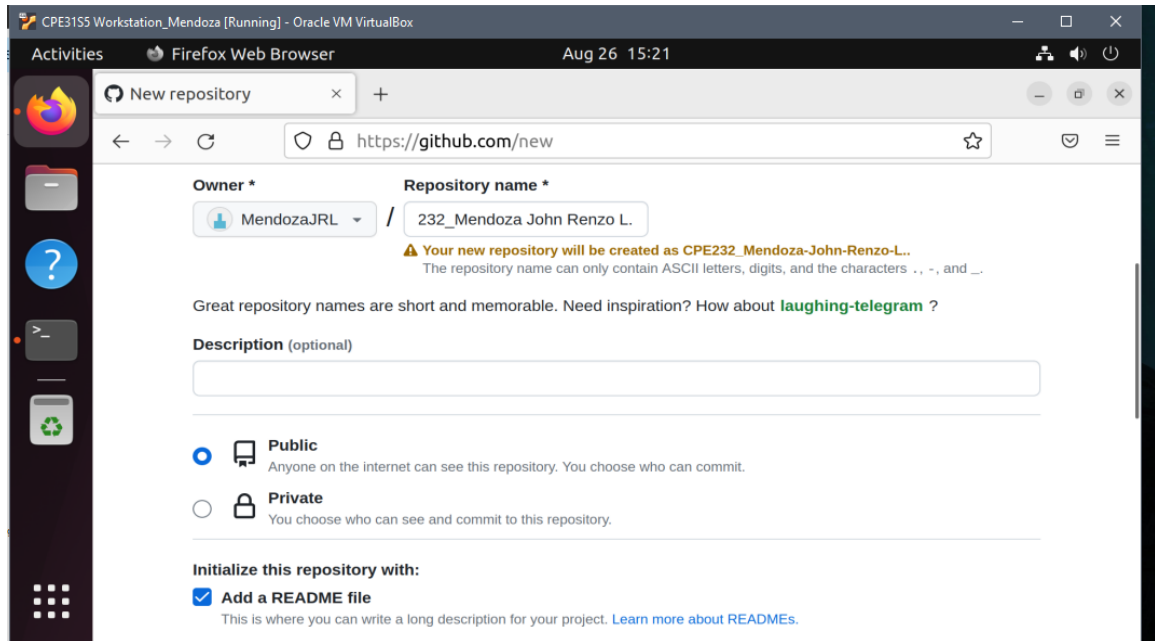
A screenshot of a terminal window showing the user "jhnrmendoza@workstation" running the command "git --version". The output is "git version 2.34.1".

```
jhnrmendoza@workstation:~$ git --version
git version 2.34.1
```

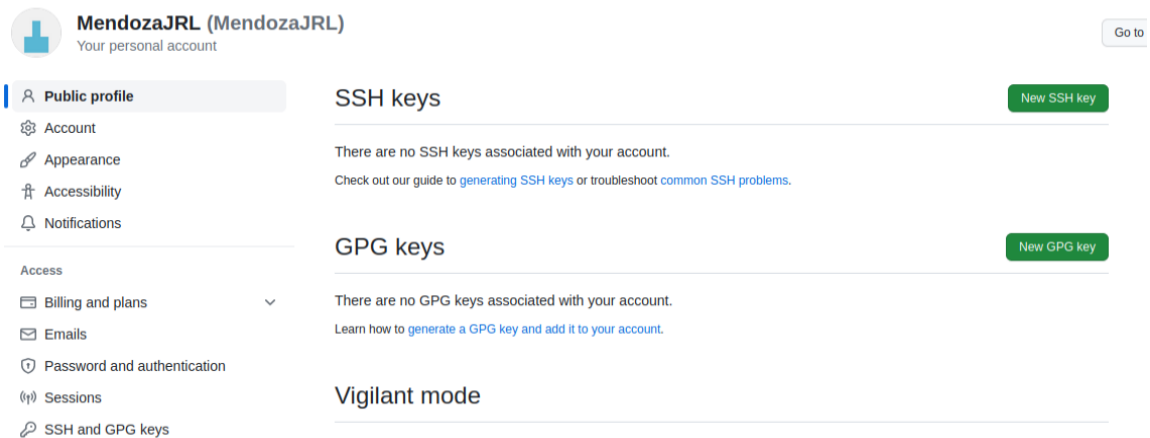
4. Using the browser in the local machine, go to [www.github.com](https://www.github.com).



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 to your profile's settings and click SSH and GPG keys. If there is an existing key, make sure to delete it. To create new SSH keys, click New SSH Key. Write CPE232 key as the title of the key.



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-ecdsa-sha2-nistp256@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.

```
jhnrmendoza@workstation:~$ cat .ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCAQDU4pRC/TyDcZAMUpjuYm1m7CvdqazaxlBYqBb7B53bTAqeAk8fCl30owq9Fgrq9r7PH0/+be+auVJfDYm
ogDg5CPa+059icDthXMSwbGA0+ecTkAPXsf1B/ByZ5AOv0Sof+a6JhGaINBh3HTHm6eopSSZwdqj5c9XCcRuEL9bUzS0giMOFj7JLRyqpECZqGgxy8MqSXRl6n
cF+sSKNwslz0uY6Ykm42dsZRVit5skIIATYBaILIHfV3E0S6No8f9YIyZy0+rk6BDc0RLz5XT86WkNewqZ98becjfuDvzprpPtLDfdgEH32CZSzdWcNqAKodY
Bjj4QKYndVsn0fTHw6vFLS4Dvq3bbN5td1+dnJY2A8N93mm92Ls1Ob6DZy3godhLB2TFtapPfPq/7A/hq3frv0Df3i1IsuHpGa0XIqk0VEuWpgy9X30awS4r19
ds1jzGoTzKV9jxL9YuCxmWk3F8pGTEDNQ1nN5AyUkh1vGKPTa50lwbDSSN/y1PvVnx6OqoNLab2qXUkWyLTyr/wOzgVZGhkQEGmQpIubSpKsQTVqUp0P4UTVLi
xMXmJ8+XiQHvQnii6Qa+VipwEGC1nwUYtfdLCHBAU9HIMYGrPFW6TD02BN9CA+baJcLCoY0uphbiw1sSCT2DW21sVKKjmdIYXC+hE87KfLorFPw== jhnrm
endoza@workstation
```

Public profile

Account

Appearance

Accessibility

Notifications

Access

Billing and plans

Emails

Password and authentication

Sessions

SSH and GPG keys

Organizations

Enterprises

Moderation

## Add new SSH Key

Title

CPE232

Key type

Authentication Key

Key

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQCAQDU4pRC/TyDcZAMUpjuYm1m7CvdqazaxlBYqBb7B53bTAqeAk8fCl30ow
q9Fgrq9r7PH0/+be+auVJfDYmogDg5CPa+059icDthXMSwbGA0+ecTkAPXsf1B
/ByZ5AOv0Sof+a6JhGaINBh3HTHm6eopSSZwdqj5c9XCcRuEL9bUzS0giMOFj7JLRyqpECZqGgxy8MqSXRl6ncF+s
SKNwslz0uY6Ykm42dsZRVit5skIIATYBaILIHfV3E0S6No8f9YIyZy0+rk6BDc0RLz5XT86WkNewqZ98becjfuDvzprpPtL
DfdgEH32CZSzdWcNqAKodYBjj4QKYndVsn0fTHw6vFLS4Dvq3bbN5td1+dnJY2A8N93mm92Ls1Ob6DZy3godhLB2
TFtapPfPq
/7A/hq3frv0Df3i1IsuHpGa0XIqk0VEuWpgy9X30awS4r19ds1jzGoTzKV9jxL9YuCxmWk3F8pGTEDNQ1nN5AyUkh1vGK
PTa50lwbDSSN/y1PvVnx6OqoNLab2qXUkWyLTyr
```

Add SSH key

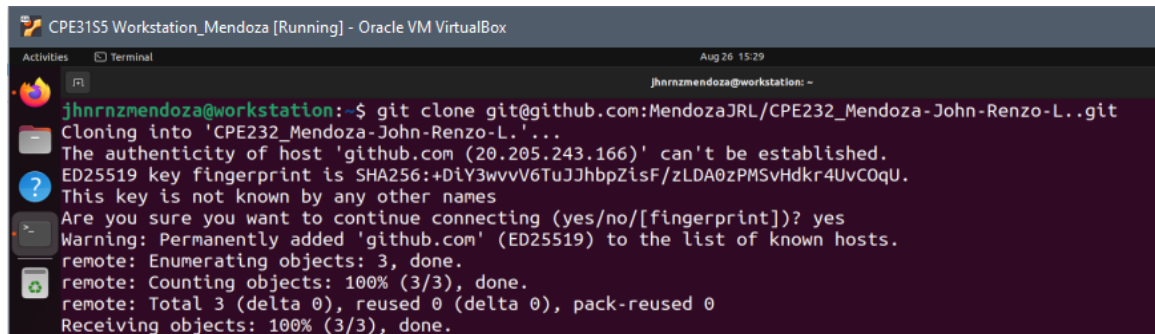
- d. Clone the repository that you created. In doing this, you need to get the link from GitHub. Browse to your repository as shown below. Click on the Code drop down menu. Select SSH and copy the link.

The image displays two screenshots of GitHub repository pages, illustrating the process of cloning a repository using SSH.

**Top Screenshot:** The repository is `jvtaylor-cpe / CPE302_yourname`. The `Code` dropdown menu is open, showing options: `HTTPS`, `SSH` (highlighted with a yellow circle), and `GitHub CLI`. The SSH link is `git@github.com:jvtaylor-cpe/CPE302_you`. Below the link, it says "Use a password-protected SSH key." and there is a "Download ZIP" button.

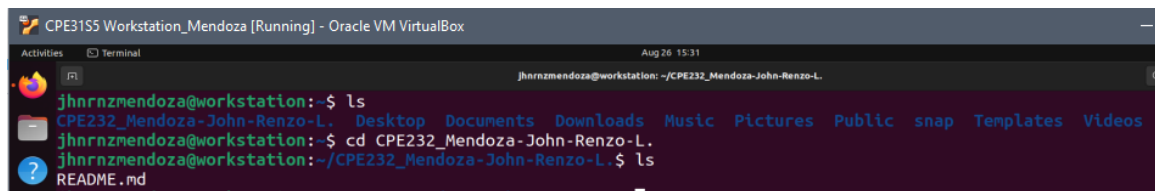
**Bottom Screenshot:** The repository is `CPE232_Mendoza-John-Renzo-L.`. The `Code` dropdown menu is open, showing options: `Local`, `Codespaces` (with a "New" button), `HTTPS`, `SSH` (highlighted with a red underline), and `GitHub CLI` (with a "New" button). The SSH link is `git@github.com:MendozaJRL/CPE232_Mendoza-`. Below the link, it says "Use a password-protected SSH key." and there is a "Download ZIP" button.

- 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.



```
CPE31S5 Workstation_Mendoza [Running] - Oracle VM VirtualBox
Aug 26 15:29
jhnrmendoza@workstation: ~
jhnrmendoza@workstation:~$ git clone git@github.com:MendozaJRL/CPE232_Mendoza-John-Renzo-L..git
Cloning into 'CPE232_Mendoza-John-Renzo-L.'...
The authenticity of host 'github.com (20.205.243.166)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvV6TuJJhpZisF/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.
```

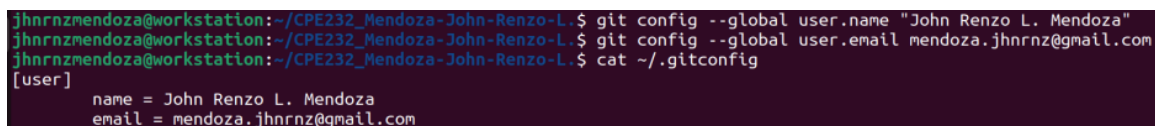
- f. To verify that you have cloned the GitHub repository, issue the command `ls`. Observe that you have the `CPE232_yourname` in the list of your directories. Use `CD` command to go to that directory and `LS` command to see the file `README.md`.



```
CPE31S5 Workstation_Mendoza [Running] - Oracle VM VirtualBox
Aug 26 15:31
jhnrmendoza@workstation: ~
jhnrmendoza@workstation:~$ ls
CPE232_Mendoza-John-Renzo-L. Desktop Documents Downloads Music Pictures Public snap Templates Videos
jhnrmendoza@workstation:~$ cd CPE232_Mendoza-John-Renzo-L.
jhnrmendoza@workstation:~/CPE232_Mendoza-John-Renzo-L.$ ls
README.md
```

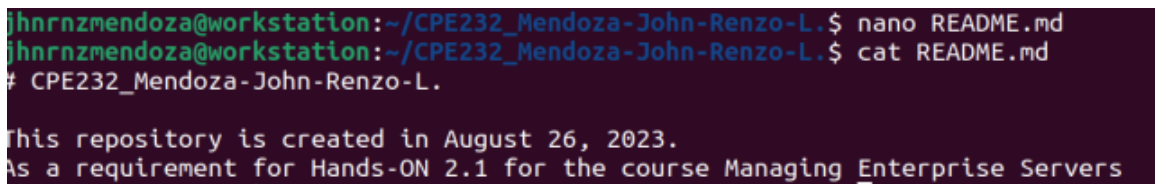
- g. Use the following commands to personalize your git.

- `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`



```
jhnrmendoza@workstation:~/CPE232_Mendoza-John-Renzo-L.$ git config --global user.name "John Renzo L. Mendoza"
jhnrmendoza@workstation:~/CPE232_Mendoza-John-Renzo-L.$ git config --global user.email mendoza.jhnrm@gmail.com
jhnrmendoza@workstation:~/CPE232_Mendoza-John-Renzo-L.$ cat ~/.gitconfig
[user]
  name = John Renzo L. Mendoza
  email = mendoza.jhnrm@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.



```
jhnrmendoza@workstation:~/CPE232_Mendoza-John-Renzo-L.$ nano README.md
jhnrmendoza@workstation:~/CPE232_Mendoza-John-Renzo-L.$ cat README.md
# CPE232_Mendoza-John-Renzo-L.

This repository is created in August 26, 2023.
As a requirement for Hands-ON 2.1 for the course Managing Enterprise Servers
```

- 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?

```
jhnrmendoza@workstation:~/CPE232_Mendoza-John-Renzo-L.$ 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")
```

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

```
jhnrmendoza@workstation:~/CPE232_Mendoza-John-Renzo-L.$ git add README.md
jhnrmendoza@workstation:~/CPE232_Mendoza-John-Renzo-L.$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   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.

```
jhnrmendoza@workstation:~/CPE232_Mendoza-John-Renzo-L.$ git commit -m "Edited README.md contents"
[main fb0d7ac] Edited README.md contents
1 file changed, 4 insertions(+), 1 deletion(-)
```

- 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`.

```
jhnrmendoza@workstation:~/CPE232_Mendoza-John-Renzo-L.$ git push origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 2 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 405 bytes | 405.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:MendozaJRL/CPE232_Mendoza-John-Renzo-L..git
5ccfa01..fb0d7ac  main -> 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 how long the last commit is. 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.

The screenshot shows a GitHub repository page for the user **CPE232\_Mendoza-John-Renzo-L.** The repository is public and has 1 branch (main) and 0 tags. The README.md file is highlighted, showing it was edited by JRLMendoza 1 minute ago. The commit history shows 2 commits. The README.md file content is displayed, showing the repository name and a description: "This repository is created in August 26, 2023. As a requirement for Hands-ON 2.1 for the course Managing Enterprise Servers". Below the README.md content, the commit history is shown, indicating 1 changed file with 4 additions and 1 deletion. The diff view shows the changes to README.md, with the new content being added.

**CPE232\_Mendoza-John-Renzo-L.**

This repository is created in August 26, 2023. As a requirement for Hands-ON 2.1 for the course Managing Enterprise Servers

**Edited README.md contents**

main

JRLMendoza committed 1 minute ago

Showing 1 changed file with 4 additions and 1 deletion.

```
@@ -1,4 @@
1 - # CPE232_Mendoza-John-Renzo-L.
1 + # CPE232_Mendoza-John-Renzo-L.
2 +
3 + This repository is created in August 26, 2023.
4 + As a requirement for Hands-ON 2.1 for the course Managing Enterprise Servers
```

**Reflections:**

Answer the following:

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

For the third task, the students were able to create a git repository in the github. By using ansible commands installed on the local machine, the students are able to import the created git from the website by using the SSH key in order to access it. Moreover, using the unique SSH key, the system administrator are able to access remote servers / directories. The imported git can also be modified using the terminal and the changes on it can be uploaded back to the github repository using ansible (git) commands.

4. How important is the inventory file?

An inventory file stores the list of the connected servers or clients in a network. It also contains the commands or scripts stored in a playbook that can be imported on the remote servers or clients to be executed. By using the inventory file, the system administrator will be able to configure multiple devices to have the same configuration without manually configuring them one by one.

**Conclusions/Learnings:**

In this hands-on activity, the students were enlightened about the secured socket shell (SSH) basic configuration as well as setting up a Git in which connects to the remote servers or clients.

The SSH enables the system administrator to remotely access a server or client by using its IP address in order to link with them. As observed on part 1 of this activity, the students used the IP address of the servers (1 and 2) and used them to remotely connect with them. The students have also added a private key (public key in server) in order to authenticate the appropriate user to access the server. This was made possible by copying the file containing the private key to the remote server.

The Git is a control system which enables the system administrator to organize the network. As observed in this activity (part 2), the students have created a repository and imported it to the local machine using git commands in the terminal. In addition, the git commands can also be used to configure the repository in the local machine. In turn, git commands can also be used to push the changes to the cloud repository in GitHub.

Using these two new concepts, the students would be able to create cloud repositories in which they can import on their local machine to simplify or to make configuration much accessible and much faster.