

AName: Robin E. Valenzuela	Date Performed:
Course/Section:CPE-232	Date Submitted:
Instructor: Engr. Jonathan Taylar	Semester and SY:
Activity 2: SSH Key-Based Authentication and Setting up Git	
1. Objectives: <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 	
Part 1: Discussion <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 <ul style="list-style-type: none"> 1. The simplest way to generate a key pair is to run <i>ssh-keygen</i> 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.

```
valenzuela@workstation:~$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/valenzuela/.ssh/id_rsa):
```

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.

```
valenzuela@workstation:~$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/valenzuela/.ssh/id_rsa):
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/valenzuela/.ssh/id_rsa
Your public key has been saved in /home/valenzuela/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:2KUCBrC/y+9H4GFgxYBy7Yee+0k4k7WbGq5/pDv/9rk valenzuela@workstation
The key's randomart image is:
+---[RSA 4096]-----+
|      .+0+      |
|     . 0.+..    |
|    o.o oo .    |
|     .o==+      |
|     .o=So      |
|      o=o..     |
|      B+o.      |
|      o+*.+o .   |
|     .oBXO+..E.  |
+---[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*.

```
valenzuela@workstation:~$ ls -la .ssh
total 24
drwx----- 2 valenzuela valenzuela 4096 Sep  1 19:52 .
drwxr-x--- 16 valenzuela valenzuela 4096 Sep  1 19:47 ..
-rw----- 1 valenzuela valenzuela 3389 Sep  1 19:52 id_rsa
-rw-r--r-- 1 valenzuela valenzuela 748 Sep  1 19:52 id_rsa.pub
-rw----- 1 valenzuela valenzuela 1404 Aug 27 12:08 known_hosts
-rw-r--r-- 1 valenzuela valenzuela 142 Aug 27 11:51 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.

```
valenzuela@workstation:~$ 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
```

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

```
valenzuela@workstation:~$ ssh-copy-id -i ~/.ssh/id_rsa valenzuela@workstation
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/valenzuela
/.ssh/id_rsa.pub"
The authenticity of host 'workstation (127.0.0.1)' can't be established.
ED25519 key fingerprint is SHA256:ZCaWD+liHMfWfoMqRCSgKHsInS7YoxoAQxj5zfwijjE.
This host key is known by the following other names/addresses:
  ~/.ssh/known_hosts:1: [hashed name]
  ~/.ssh/known_hosts:4: [hashed name]
  ~/.ssh/known_hosts:5: [hashed name]
  ~/.ssh/known_hosts:6: [hashed name]
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
valenzuela@workstation's password:

Number of key(s) added: 1

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

```
valenzuela@workstation:~$ cd .ssh
valenzuela@workstation:~/.ssh$ ls
authorized_keys  id_rsa  id_rsa.pub  known_hosts  known_hosts.old
valenzuela@workstation:~/.ssh$ cat authorized_key
cat: authorized: No such file or directory
cat: key: No such file or directory
valenzuela@workstation:~/.ssh$ cat authorized_key
cat: authorized_key: No such file or directory
valenzuela@workstation:~/.ssh$ cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQACc9QHh9Quds+zqF7awZeDQhUyXqc0rRgk8U/scHjp
pHZnWiF18QgMkn5Fby4bGecvMxDFkX9hN7HZSCaYEXJ7EoXu4/jZYPsvQ1X2j6ppC+7pEQMaMwxhZT
3CQ+b2tt/OVDhMVVPV5Hb7eReYNG/mPNaxv6pPp6Kn87i9ExAFPMHHIm0FFN1a1yzc/RvgvOetdaqTe
FLHD/brxzGSQR5bZ0Us8L45AeTPX8dSf+BEwHw4YGwQvySSUepP/TdUVoYgi8oN8D/mREKSJwL+7lGA
WqegnvDd7pFbyz4s7bnfLFkwBP0MC2aaE2MhERZtakv6TsV6tETL5mGRjm2MDxJdGb6rLWHqGM8BbZG
l1bCn41pLT05FBsgAgPA2ByvCgr9RwnsnGFMFo744Be9qKNM2uG6prGLvNQa0ki9H30l1B1Lcw5s56V
SNJ368QnxsJ/mDUie37H5tYDymisfAlU6XtJzQyGSjzySQqC4zAtwNmSXtUkxR8S4ndzy9k95MphuaR
o10JlH0remxyK960Tmc7cq41tfVdKKXk8CrAMUCFuAVW99oa6q955fCERlILAimsVlZeHWGSmkJKipg
ik8cJhLgGfW/yux/KgLI3p1LSbzHhY7DwPuRjzPJ2AMPvbmDcVYvHNQhM0IwwqY+B0s9RCIV8cU7dz
lTipaRP56kQ== valenzuela@workstation
```

```

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

Number of key(s) added: 1

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

valenzuela@workstation:~$ ssh valenzuela@server2
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: Thu Sep  1 20:36:45 2022 from 192.168.56.104
valenzuela@workstation:~$ ssh-copy-id -i ~/.ssh/id_rsa valenzuela@server1
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/valenzuela
/.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 promp
ted now it is to install the new keys
valenzuela@server1's password:

Number of key(s) added: 1

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

valenzuela@workstation:~$ ssh valenzuela@server1
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: Thu Sep  1 20:35:58 2022 from 192.168.56.104

```

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 connection did not ask for a password because it is connected to the SSH server with the private key.

Reflections:

Answer the following:

1. How will you describe the ssh-program? What does it do?
 - SSH program is a remote access device that enables two networks to communicate and encrypts the data being transferred.
2. How do you know that you already installed the public key to the remote servers?
 - By using linux commands to go into root mode to see the authorized keys in the .ssh directory.

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*

```

valenzuela@workstation:~$ which git
valenzuela@workstation:~$ sudo apt install git
[sudo] password for valenzuela:
Sorry, try again.
[sudo] password for valenzuela:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  git-man liberror-perl
Rhythmbox 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 4,110 kB of archives.
After this operation, 20.9 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ph.archive.ubuntu.com/ubuntu jammy/main amd64 liberror-perl all 0.
17029-1 [26.5 kB]
Get:2 http://ph.archive.ubuntu.com/ubuntu jammy-updates/main amd64 git-man all
1:2.34.1-1ubuntu1.4 [952 kB]
Get:3 http://ph.archive.ubuntu.com/ubuntu jammy-updates/main amd64 git amd64 1:
2.34.1-1ubuntu1.4 [3,131 kB]
Fetched 4,110 kB in 2s (1,546 kB/s)

```

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

```

valenzuela@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.

```

valenzuela@workstation:~$ git --version
git version 2.34.1

```

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 repository elsewhere? [Import a repository.](#)

Owner *

qrevalenzuela

Repository name *

CPE232_Valenzuela

Great repository names are short and memorable. Need inspiration? How about [congenial-octo-goggles?](#)

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:

Skip this step if you're importing an existing repository.

☒ Add a README file

This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore

Choose which files not to track from a list of templates. [Learn more.](#)

.gitignore template: None

Choose a license

A license tells others what they can and can't do with your code. [Learn more.](#)

License: None

- 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.

Key

ssh-rsa

```
AAAAB3NzaC1yc2EAAAADAQABAAQACc9QHh9Quds+zqF7awZeDQhUyXqcOrRgk8U/scHjpp
HZnWiF18QgMkn5Fby4bGecvMxDFkJx9hN7HZSCaYEXJ7EoXu4
/jZYPsvQ1X2j6ppC+7pEQMaMWxhZT3CQ+b2tt/OVDhMVVPV5Hb7eReYNG
/mPNAxv6pPp6Kn87i9ExAFPMHHIm0FFN1a1yzc/RvgvOetdagTeFIHD
/brxzGSQR5bZ0Us8L45AeTPX8dSf+BEwHw4YGwQvyS5UepP/TdUVoYgi8oN8D
/mREKSJwL+7lGAWqegnvDd7pFbyz4s7bNfLFkwBP0MC2aaE2MhERZtakv6TsV6tETL5mGRjM2M
DxJdGb6rlWHqGM8BbZGI1bCn41pLT05FBsgAgPA2ByvCgr9RwnsnGFMFo744Be9qKNM2uG6prG
LvNQA0ki9H30l1B1Lcw5s56VSNJ368QnxsJ
```

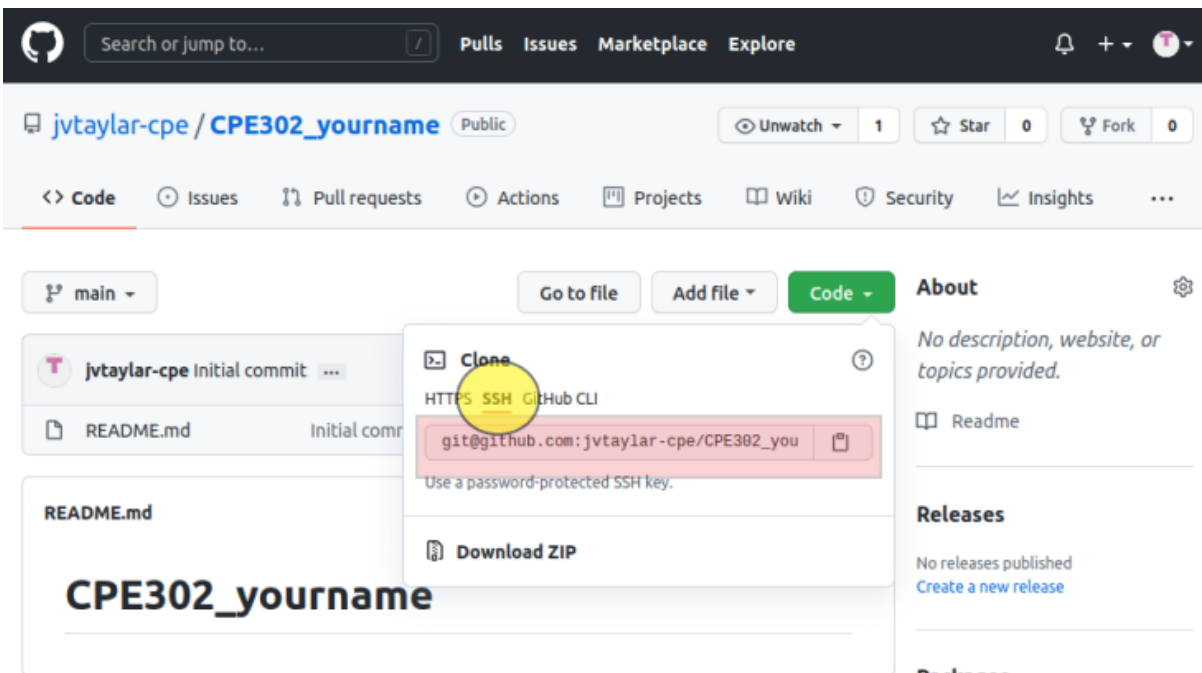
- c. On the local machine's terminal, issue the command `cat .ssh/id_rsa.pub` and copy the public key. Paste it on the GitHub key and press Add SSH key.

```

valenzuela@workstation:~$ cat .ssh/id_rsa.pub
a AAAAB3NzaC1yc2EAAAADAQABAAQCCc9QHh9Quds+zqF7awZeDQhUyXqc0Rgk8U/schjp
pHZnWiF18QgMkn5Fby4bGecvMxDfKJx9hN7HZSCaYEXJ7EoXu4/jZYPsvQ1X2j6ppC+7pEQMaMwxhZT
3CQ+b2tt/OVDhMVVPV5Hb7eReYNG/MPNaxv6pPp6Kn87i9ExAFPMHHIm0FFN1a1yzc/RvgvOetdaqTe
FLHD/brxzGSQR5bZ0Us8L45AetPX8dSf+BEwHw4YGwQvyS5UepP/TdUVoYgI8oN8D/mREKSJwL+7LGA
WqegnvDd7pFbyz4s7bNfLFkwBP0MC2aaE2MhERZtakv6TsV6tETL5mGRjM2MDxJdGb6rLWHqGM8BbZG
l1bCn41pLT05FBsgAgPA2ByvCgr9RwnsnGFMFo744Be9qKNM2uG6prGLvNQa0ki9H30l1B1Lcw5s56V
SNJ368QnxsJ/mDUIe37H5tYDYmisfALU6XtJzQyGSjzySQqC4zAtwNmSXtUkxR8S4ndzy9k95MphuaR
o10JlH0remxyK960Tmc7cq41tfVdKKXk8CrAMUCFuAVW99oa6q955fCErLlLAimsVLZeHWGSmkJKipg
ik8cJhLGdGfW/yux/KgLI3p1LSbzHhY7DwPuRjzPJ2AMPvbmdCvYvHNQhM0IwwqY+B0s9RCIV8cU7dz
lTipaRP56kQ== valenzuela@workstation

```

- 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.

```

valenzuela@workstation:~$ git clone git@github.com:qrevalenzuela/CPE232_Valenzuela.git
Cloning into 'CPE232_Valenzuela'...
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])? y
Please type 'yes', 'no' or the fingerprint: yes
Please type 'yes', 'no' or the 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.
valenzuela@workstation:~$

```

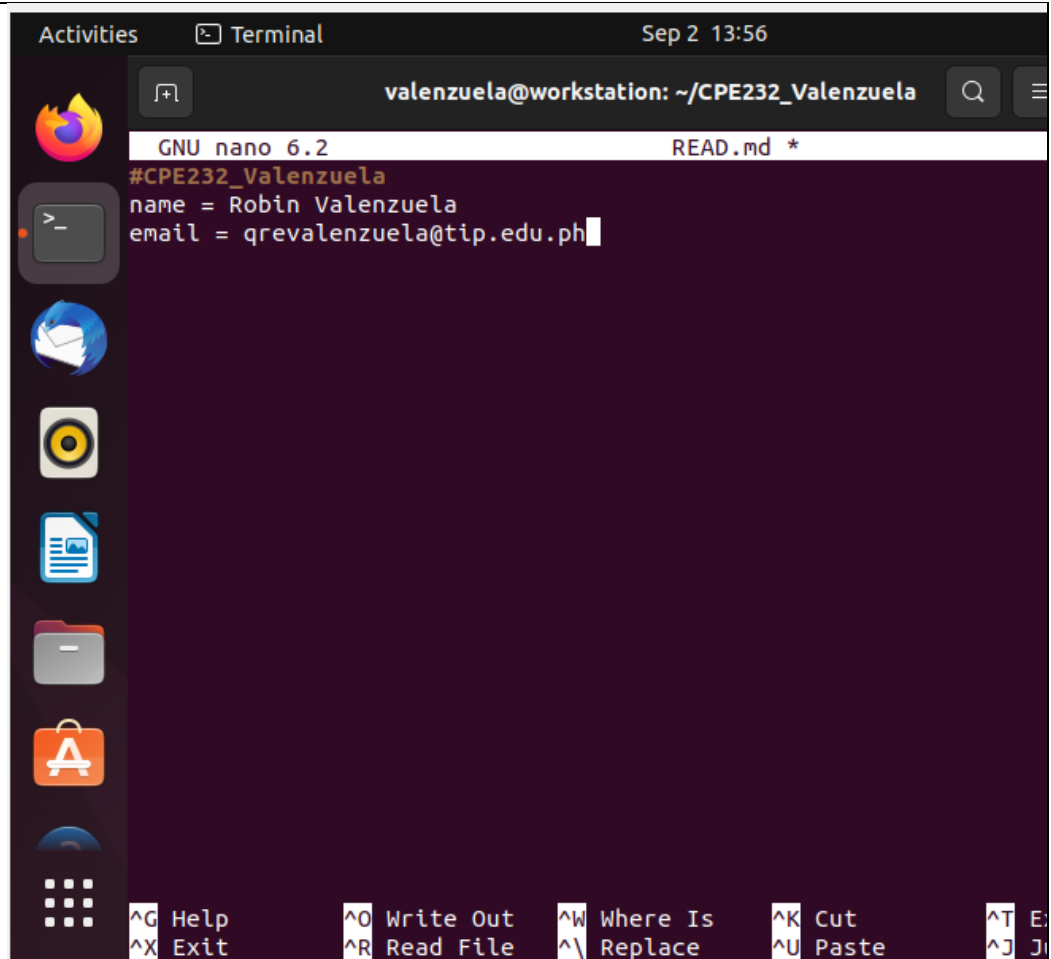

- f. To verify that you have cloned the GitHub repository, issue the command `ls`. Observe that you have the `CPE232_yourname` in the list of your directories. Use `CD` command to go to that directory and `LS` command to see the file `README.md`.

```
Receiving objects: 100% (5/5), done.
valenzuela@workstation:~$ ls
CPE232_Valenzuela  Documents  Music      Public  Templates
Desktop            Downloads  Pictures   snap    Videos
valenzuela@workstation:~$ cd CPE232_Valenzuela
valenzuela@workstation:~/CPE232_Valenzuela$ ls
README.md
valenzuela@workstation:~/CPE232_Valenzuela$
```

- 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`

```
valenzuela@workstation:~/CPE232_Valenzuela$ git config --global user.name "Robin Valenzuela"
valenzuela@workstation:~/CPE232_Valenzuela$ git config --global user.email grevalenzuela@tip.edu.ph
valenzuela@workstation:~/CPE232_Valenzuela$ cat ~/.gitconfig
[user]
  name = Robin Valenzuela
  email = grevalenzuela@tip.edu.ph
```

- 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.



```
Activities Terminal Sep 2 13:56
valenzuela@workstation: ~/CPE232_Valenzuela
GNU nano 6.2 READ.md *
#CPE232_Valenzuela
name = Robin Valenzuela
email = grevalenzuela@tip.edu.ph
^G Help ^X Exit ^O Write Out ^R Read File ^W Where Is ^_ Replace ^K Cut ^U Paste ^T E ^J J
```

- 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?

```
valenzuela@workstation:~/CPE232_Valenzuela$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
  README.md

nothing added to commit but untracked files present (use "git add" to track)
```

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

```
valenzuela@workstation:~/CPE232_Valenzuela$ git add README.md
valenzuela@workstation:~/CPE232_Valenzuela$ git commit -m "Let's go"
[main 51fb36e] Let's go
1 file changed, 5 insertions(+), 1 deletion(-)
```

- 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

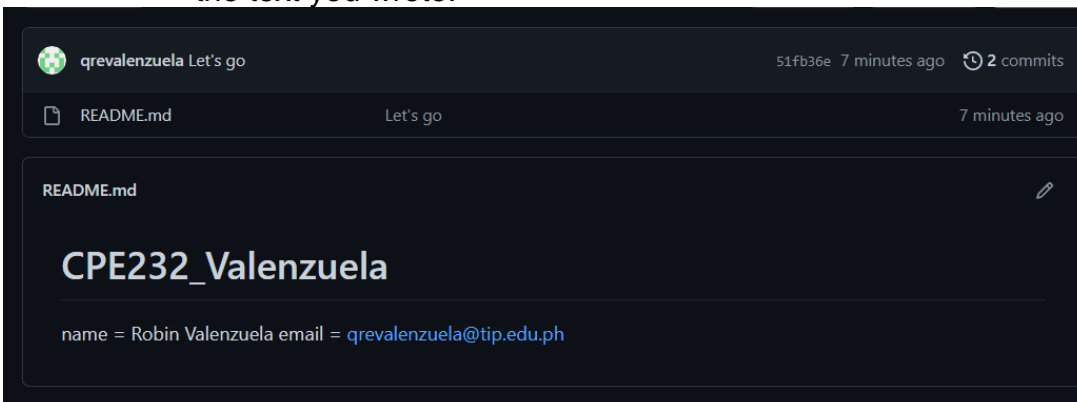
command is required to select the changes that will be staged for the next commit.

```
valenzuela@workstation:~/CPE232_Valenzuela$ git add README.md
valenzuela@workstation:~/CPE232_Valenzuela$ git commit -m "Let's go"
[main 51fb36e] Let's go
1 file changed, 5 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`.

```
valenzuela@workstation:~/CPE232_Valenzuela$ git push origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 311 bytes | 311.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:qrevalenzuela/CPE232_Valenzuela.git
8f0eb50..51fb36e 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 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.



The title reflects on the message that was being committed. The README.md is edited in the linux server terminal using nano command.

Reflections:

Answer the following:

3. What sort of things have we so far done to the remote servers using ansible commands?
 - The things that we have done is creating and using SSH for remote servers so that we know that the data is encrypted. Creating private networks and servers for personal use especially for the GITHUB.
4. How important is the inventory file?
 - **It is very important because it can list all of the things in your system and can be used to store important data as well.**

Conclusions/Learnings:

In conclusion, learning about remote servers and networks is a very important part in becoming a system administrator. It is because of the nature of the job that hints many servers at once and not only one or two. We are challenged with creating our encrypted SSH keys to teach us about cybersecurity. Over all, it is a fun activity to do and learn.

I affirm that I shall not give or receive any unauthorized help on this assignment and that all work shall be my own.