

Name: Repani, Justin Jello J.	Date Performed: August 24, 2023
Course/Section: CPE31S6	Date Submitted: August 24, 2023
Instructor: Dr. Jonathan V, Taylar	Semester and SY: 1st, SY 2023-2023
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>	
<p>Task 1: Create an SSH Key Pair for User Authentication</p> <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.

```
jello@workstation: ~  
File Edit View Search Terminal Help  
jello@workstation:~$ ssh-keygen -t rsa -b 4096  
Generating public/private rsa key pair.  
Enter file in which to save the key (/home/jello/.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.

```
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in /home/jello/.ssh/id_rsa.  
Your public key has been saved in /home/jello/.ssh/id_rsa.pub.  
The key fingerprint is:  
SHA256:B4ImY5ndkFMrOGR+9sCjWJK4merMXUsI9Ag1uYFFNIY jello@workstation  
The key's randomart image is:  
+---[RSA 4096]---+  
| .+=o. |  
|o*EO+= . |  
|=o@+@.+ . |  
|o0=0 = . . |  
|=o.. . S . |  
|. . . . |  
|. . 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`.

```
jello@workstation:~$ ls -la .ssh  
total 20  
drwx----- 2 jello jello 4096 Aug 24 17:33 .  
drwxr-xr-x 16 jello jello 4096 Aug 24 17:19 ..  
-rw----- 1 jello jello 3243 Aug 24 17:33 id_rsa  
-rw-r--r-- 1 jello jello 743 Aug 24 17:33 id_rsa.pub  
-rw-r--r-- 1 jello jello 888 Aug 17 18:07 known_hosts
```

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`

```
jello@workstation:~/.ssh$ ssh-copy-id -i ~/.ssh/id_rsa jello@server1
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/jello/.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
jello@server1's password:

Number of key(s) added: 1

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

```
jello@workstation:~/.ssh$
```

```
jello@workstation:~/.ssh$ ssh-copy-id -i ~/.ssh/id_rsa jello@server2
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/jello/.ssh
/id_rsa.pub"
The authenticity of host 'server2 (192.168.56.103)' can't be established.
ECDSA key fingerprint is SHA256:7BrewEemwrzk/jFucX235tXhbyUu9o1M+iQ/8cv5I0M.
Are you sure you want to continue connecting (yes/no)? 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
jello@server2's password:

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh 'jello@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.

```
jello@server1:~$ ls
Desktop  Downloads  Music      Public      Videos
Documents  examples.desktop  Pictures  Templates
jello@server1:~$ cd .ssh
jello@server1:~/.ssh$ ls
authorized_keys
jello@server1:~/.ssh$ cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQADibLS10sJS3as9uYJavC2caL03QSpdwgFyYpbV0wX
pxsSlx0r1uEEBiMJCuCk3jBnS6QPS/oe1tcgeT7vgMgogQohsdeAo+aRnIzEuFjYs7qohTca0xPSfga
mzdijgQYBKK0e4QPGH6wELheb7u2noq9WGbMpw53II+7uTVfsU38C2BOSDj0D0wwRJWt2Z9Qg1bM6U6
ogE7MpULjME9EguM/ow80bocUhBeTrBeLfNepT6mYW/6m0JisG1+zouUKJCFJ0pKQaF2njErgBq/6/e
KaN78VzxfxIeIBsRWIyx+ev5A7bNtqrj0eepQNi7dvGWEEaxYiPqIEMSRJooR8ejcKp5FIMnrAT6hee
30ktbJHGU6Xs8365rT5AR3/UbqWf/UYmL2R2HskDfWzYKzd6gTfMWAfWyl5zKKVvwV6nq4tb/7REBNe
nC4UcK0etk5z6n+AAtwLgXiKHhw/cqLWqZKvt1Jadzv+6/xmgSUW99vhatzjuLY68z10AA1VcUHysC9
WG5wcUdiHRS/5PXtKXk+yJ8A0htg03x2culHIJhDv6PlawIs5CkjlFGyj/6fiK0uMRhANI0LDruPkfh
5aJMJNJ9N+SMbfHdW5fnMwZbmiFFumFfaZcWFrXMqJbXm25cctqWSYkuTzZ/gZMw60mKfyq/s7ciBo
1bxhsjcICcw== jello@workstation
jello@server1:~/.ssh$
```

```
jello@server2:~$  
jello@server2:~$ cd .ssh  
jello@server2:~/.ssh$ ls  
authorized_keys  
jello@server2:~/.ssh$ cat authorized_keys  
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQAdibLS10sJS3as9uYJavC2caL03QSpdwgFyYpbV0wX  
pxsSlxOr1uEEBiMJCuKc3jBnS6QPS/oe1tcgeT7vgMgogQohsdeAo+aRnIzEuFjYs7qohTca0xPSfga  
mzdijgQYBKK0e4QPGH6wELheb7u2noq9WGbMpw53II+7uTVfsU38C2B0SDj0D0wwRJWt2Z9Qg1bM6U6  
ogE7MpULjME9EguM/oW80bocUhBeTrBeLfNepT6mYW/6m0JisG1+zouUKJCFJ0pKQaF2njErgBq/6/e  
KaN78VzxfxIeIBsRWIyx+ev5A7bNtqrj0eepQNi7dvGWEEaxYiPqIEMSRJooR8ejcKp5FIMnrAT6hee  
30ktbJHGU6Xs8365rT5AR3/UbqwF/UYmL2R2HskDfWzYKzd6gTfMWAfWyl5zKKVwwV6nq4tb/7REBNe  
nC4UcK0etk5z6n+AAwLgXiKHhw/cqLWqZKvt1Jadzv+6/xmgSUW99vhatzjuLY68z10AA1VcUHysC9  
WG5wcUdiHRS/5PXtKXk+yJ8A0htg03x2culHIJhDv6PlawIs5CkjlFCgyj/6fik0uMRhANI0lDruPkfh  
5aJMJNJ9N+SMbfHdW5fnMwZbmiFFumFfaZcWFrXMqJbXM25cctqWSYkuTzZ/gZMw60mKfyq/s7ciBo  
1bxhsjcICCw== jello@workstation  
jello@server2:~/.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?

```
jello@workstation:~$ ssh jello@server1
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 5.4.0-150-generic x86_64)

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

Expanded Security Maintenance for Infrastructure is not enabled.

0 updates can be applied immediately.

85 additional security updates can be applied with ESM Infra.
Learn more about enabling ESM Infra service for Ubuntu 18.04 at
https://ubuntu.com/18-04

New release '20.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Your Hardware Enablement Stack (HWE) is supported until April 2023.
Last login: Thu Aug 24 17:39:55 2023 from 192.168.56.101
jello@server1:~$ ls

jello@workstation:~$ ssh jello@server2
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 5.4.0-150-generic x86_64)

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

Expanded Security Maintenance for Infrastructure is not enabled.

0 updates can be applied immediately.

85 additional security updates can be applied with ESM Infra.
Learn more about enabling ESM Infra service for Ubuntu 18.04 at
https://ubuntu.com/18-04

New release '20.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Your Hardware Enablement Stack (HWE) is supported until April 2023.
Last login: Thu Aug 24 17:41:24 2023 from 192.168.56.101
jello@server2:~$
```

Reflections:

Answer the following:

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

The SSH program gives encryption to the user login by adding a username and a password which is encrypted, making it more secure. Additionally it can generate keys that the users can use in order to login to the network without having to enter any username or password.

2. How do you know that you already installed the public key to the remote servers?
When the user connects to a network without needing to enter any passwords.

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*

```
jello@workstation:~$ sudo apt install git
[sudo] password for jello:
Sorry, try again.
[sudo] password for jello:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
```

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

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

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

```
jello@workstation:~$ git --version
git version 2.17.1
```

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.](#)

Required fields are marked with an asterisk (*).

Owner * JelzLow / Repository name * CPE232_Repani
✓ CPE232_Repani is available.

Great repository names are short and memorable. Need inspiration? How about [bookish-system](#) ?


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.](#)

 **CPE232_Repani** Public Pin Unwatch 1

 **main**  **1 branch**  **0 tags** Go to file Add file Code

 JelzLow Initial commit	a559487 now	 1 commit
 README.md	Initial commit	now

README.md 

CPE232_Repani

SysAdS6

- 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.

SSH keys

[New SSH key](#)

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

Authentication Keys

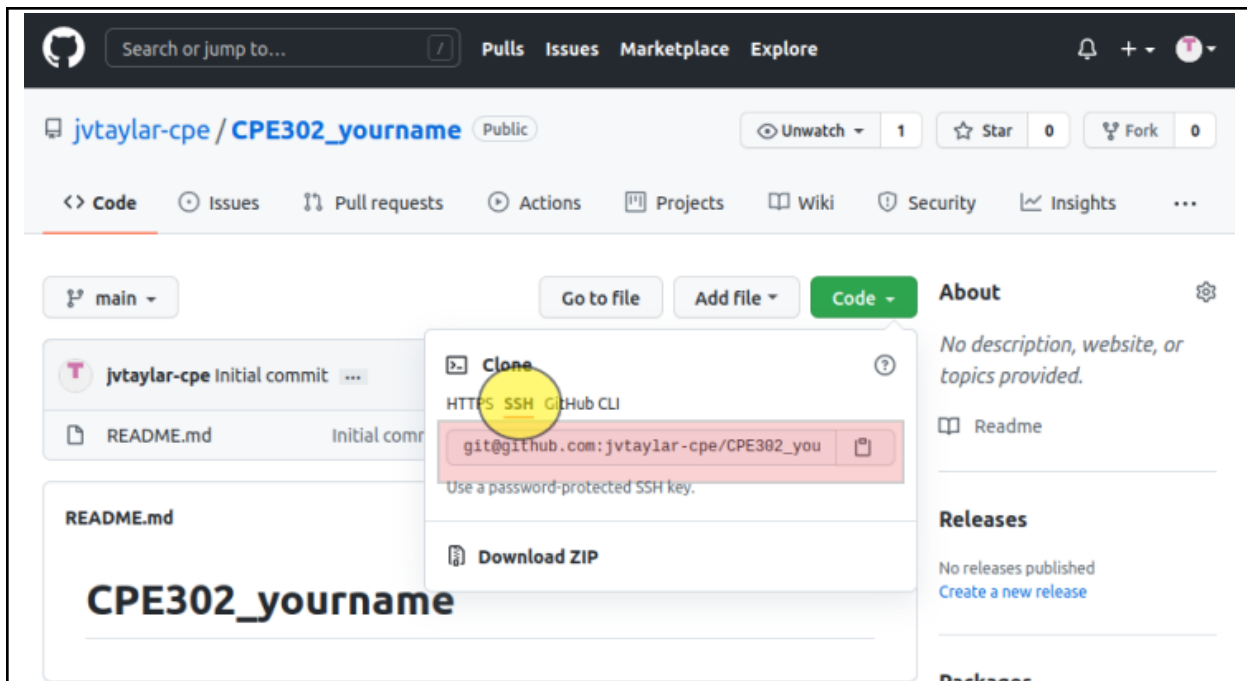
**CPE232**
SHA256:B4ImY5ndkFMR0GR+9sCjWJK4merMXUsI9Ag1uYfFNIY
SSH Added on Aug 24, 2023
Never used — Read/write

Delete

- 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.

```
jello@workstation:~$ cd .ssh
jello@workstation:~/.ssh$ cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQADiBLS10sJS3as9uYJavC2caL03QSpdwgFyYpbVOWX
pxsSlx0r1uEEBiMJCuKc3jBnS6QPS/oe1tcgeT7vgMgogQohsdeAo+aRnIzEuFjYs7qohTca0xPSfga
mzdiJgQYBKK0e4QPGH6wELheb7u2noq9WGbMpw53II+7uTVfsU38C2B0SDj0D0wwRJWt2Z9Qg1bM6U6
ogE7MpULjME9EguM/ow80bocUhBeTrBeLfNepT6mYW/6m0JisG1+zouUKJCFJ0pKQaF2njErgBq/6/e
KaN78VzxfxIeIBsRWIyx+ev5A7bNtqrj0eepQNi7dvGWEEaxYiPqIEMSRJooR8ejcKp5FIMnrAT6hee
30ktbJHGU6Xs8365rT5AR3/UbqwF/UymL2R2HskDfWzYKzd6gTfMWAfWyl5zKKVwwV6nq4tb/7REBNe
nC4UcK0etk5z6n+AAtwLgXiKHhw/cqLWqZKvt1Jadzv+6/xmgSUW99vhatzjuLY68z10AA1VcUHysC9
WG5wcUdiHRS/5PXtKXk+yJ8A0htg03x2culHIJhDv6PlawIs5CkjlFGyj/6fiK0uMRhANI0ldruPkfh
5aJMjNJ9N+SmbfnHdW5fnMwZbmiFFumFfaZcWFrXMqJbXM25cctqWSYkuTZZ/gZMw60mKfyq/s7ciBo
1bxhsjcICcw== jello@workstation
jello@workstation:~/.ssh$
```

- 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.

```
jello@workstation:~/Documents$ git clone https://github.com/JelzLow/CPE232_Repa
ni.git
Cloning into 'CPE232_Repani'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking 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`.

```
jello@workstation:~/Documents$ ls
CPE232_Repani
jello@workstation:~/Documents$ ls CPE232_Repani
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`

```
jello@workstation:~/Documents$ cd CPE232_Repani
jello@workstation:~/Documents/CPE232_Repani$ cat README.md
# CPE232_Repani
SysAdS6
jello@workstation:~/Documents/CPE232_Repani$
```

- 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.

```
jello@workstation: ~/Documents/CPE232_Repani
File Edit View Search Terminal Help
GNU nano 2.9.3 README.md Modified

# CPE232_Repani
SysAdS6

This is my first commit

File Name to Write: README.md
^G Get Help      M-D DOS Format   M-A Append      M-B Backup File
^C Cancel        M-M Mac Format   M-P Prepend     ^T To Files
```

- 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?

```
jello@workstation:~/Documents/CPE232_Repani$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")
```

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

```
jello@workstation:~/Documents/CPE232_Repani$ git add README.md
jello@workstation:~/Documents/CPE232_Repani$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    modified:   README.md
```

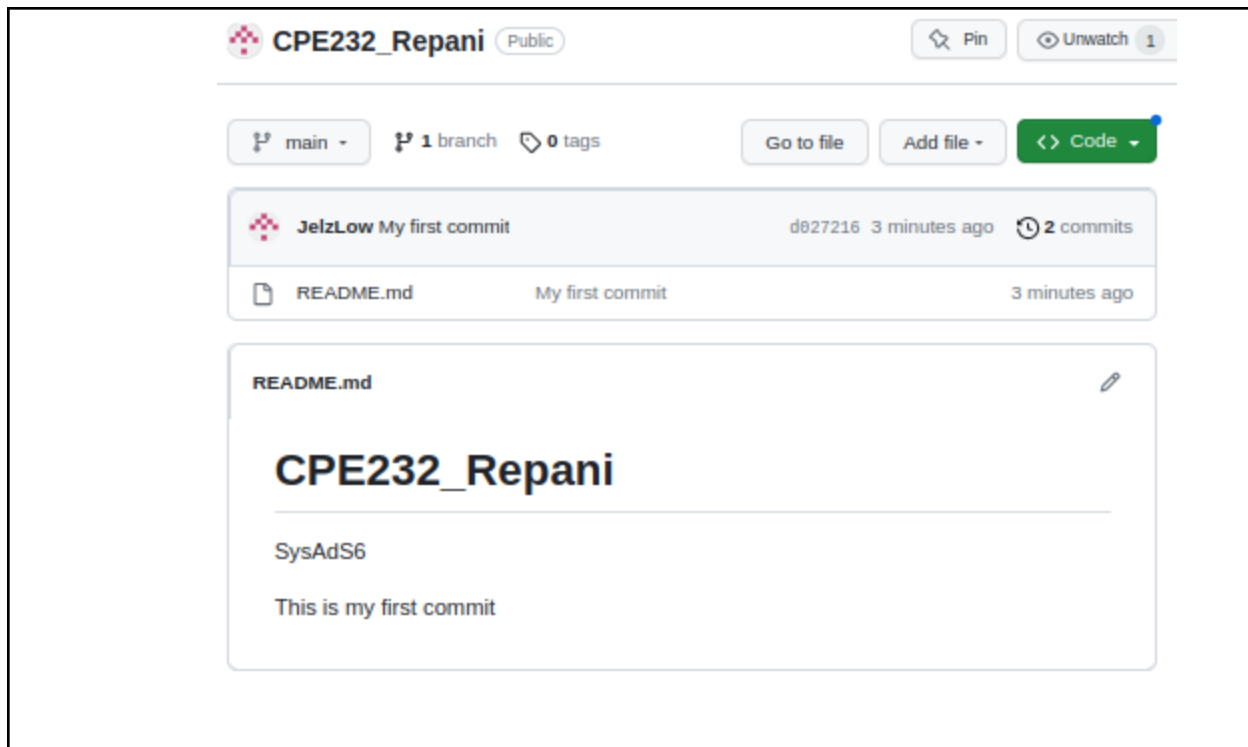
- k. Use the *git commit -m "your message"* to create a snapshot of the staged changes along the timeline of the Git projects history. The use of this command is required to select the changes that will be staged for the next commit.

```
jello@workstation:~/Documents/CPE232_Repani$ git commit -m "My first commit"
[main d027216] My first commit
1 file changed, 2 insertions(+)
```

- 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*.

```
jello@workstation:~/Documents/CPE232_Repani$ git push origin main
Username for 'https://github.com': JelzLow
Password for 'https://JelzLow@github.com':
Counting objects: 3, done.
Writing objects: 100% (3/3), 296 bytes | 296.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/JelzLow/CPE232_Repani.git
a559487..d027216  main -> main
jello@workstation:~/Documents/CPE232_Repani$
```

- 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?

The commands are used to connect and verify the connections within the SSH network. This allows use to remotely configure the servers using the workstation without having to access the PC itself.

4. *How important is the inventory file? (Hindi muna kasama)*

Conclusions/Learnings:

In this hands-on-activity 2. We are tasked to generate a key on the workstation to make it secure with the public and private key. The public key is then copied to server1 and server2. This allows the connection to the servers without having to enter a password. In the second part of the activity a git repository is set-up in order to enable the synchronization of files between the github cloud storage and the local system which allows for better collaboration between different people from different places.

"I affirm that I have not given or received any unauthorized help on this assignment, and that this work is my own."