Activity No. 14

SSH Key-Based Authentication and GIT Setup

Name: James Daniel M. Verano	**Date Performed: 11/12/2025**
Course Code: CPE 201A	**Date Submitted: 11/12/2025**
Course Title: Computer System Administration and Troubleshooting	**Instructor: Engineer. Jimlord M. Quejado**

1. Objective/s:

This activity aims to demonstrate students’ ability to configure secure SSH key-based authentication and perform version control operations using Git and GitHub.

2. Intended Learning Outcome/s:

By the end of this activity, the students should be able to:

- Analyze how SSH key-based authentication provides secure access.
- Evaluate the setup of SSH and Git configuration.
- Create and manage a Git repository using SSH connection.

3. Discussion:

Part 1: Discussion
It is assumed that you are already done with the last Activity (**Laboratory Activity 9 | Install Linux in a Virtual Machine and Explore the GUI**).
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.

Part 2: Discussion
Provide screenshots for each task.

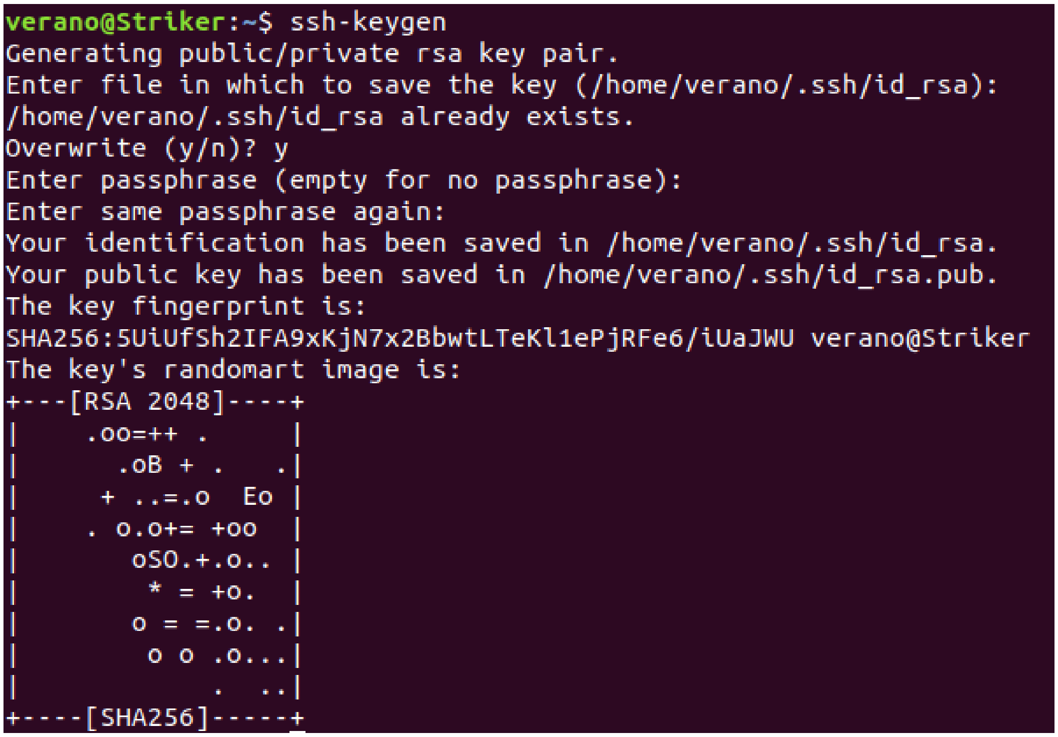
Set up Git
At the heart of GitHub is an open-source version control system (VCS) called Git. Git is responsible for everything GitHub-related that happens locally on your computer. To use Git on the command line, you'll need to download, install, and configure Git on your computer. You can also install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

4. Procedures:

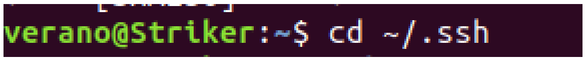
Task 1: Create an SSH Key Pair for User Authentication

1. Open VirtualBox and start your Ubuntu virtual machine.
2. Log in using your username and password.
3. Open the Terminal.
4. Generate an SSH key pair by typing the following command and pressing Enter:



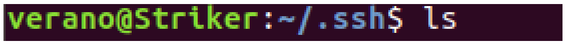
```
verano@Striker:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/verano/.ssh/id_rsa):
/home/verano/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/verano/.ssh/id_rsa.
Your public key has been saved in /home/verano/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:5UiUfSh2IFA9xKjN7x2BbwtLTeKl1ePjRFe6/iUaJWU verano@Striker
The key's randomart image is:
+---[RSA 2048]----+
|    .oo=++ .     |
|      .oB + .   .|
|     + ..=.o  Eo |
|    . o.o+= +oo  |
|       oSO.+.o.. |
|        * = +o.  |
|       o = =.o. .|
|        o o .o...|
|            .  ..|
+----[SHA256]-----+
```

5. Navigate to the SSH directory:
   cd ~/.ssh



```
verano@Striker:~$ cd ~/.ssh
```

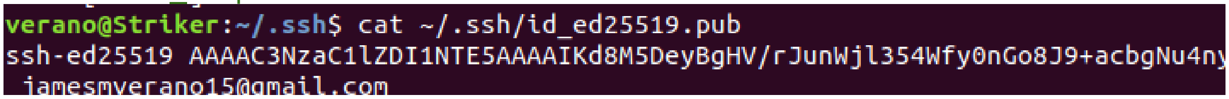
6. List the files in the directory:
   ls



```
verano@Striker:~/.ssh$ ls
```

   Look for a file ending with .pub this is your public key.

7. Display the contents of your public key file (replace id_rsa.pub with your actual filename if different):
   cat id_rsa.pub

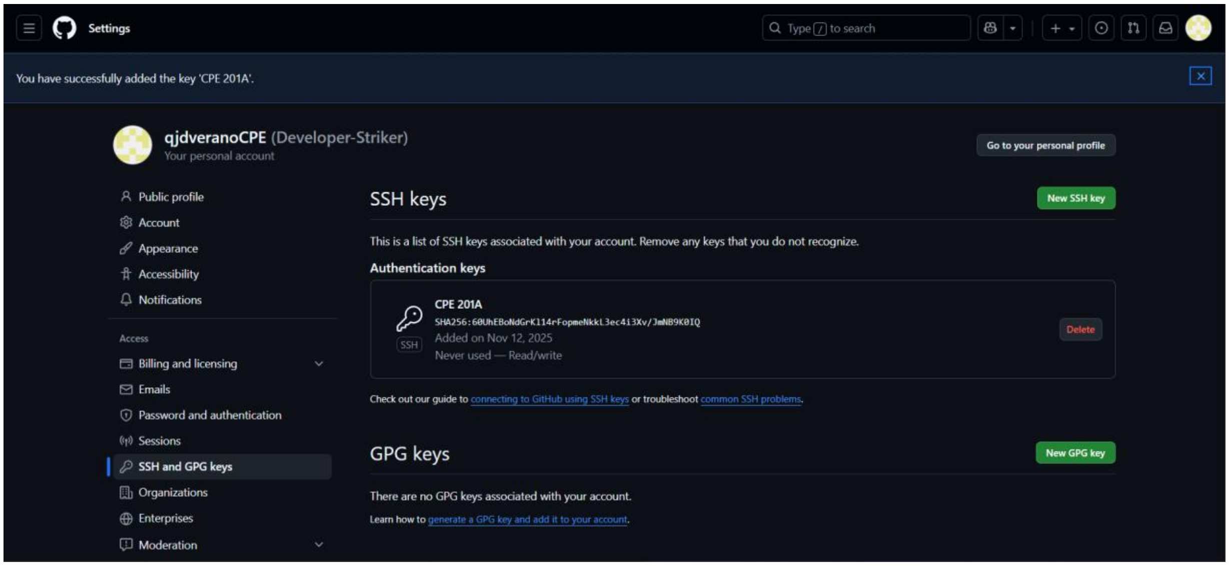


```
verano@Striker:~/.ssh$ cat ~/.ssh/id_ed25519.pub
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIKd8M5DeyBgHV/rJunWjl354Wfy0nGo8J9+acbgNu4ny
 jamesmverano15@gmail.com
```

8. Copy the entire output: this is your SSH public key, which you can use for authentication.

Task 2: Copying the Public Key to Remote Servers

1. Open your GitHub account in a web browser.
2. Click on your profile icon (upper-right corner) and go to Settings.
3. In the left sidebar, select SSH and GPG keys.
4. If there is an existing SSH key, you may delete it first.
5. Click the “New SSH key” button.
6. Enter CPE201A as the Title.
7. In the Key field, paste the SSH public key that you copied from the terminal in Task 1.
8. Click “Add SSH key” to save your new key.



You have successfully added the key 'CPE 201A'.

qjdveranoCPE (Developer-Striker)
Your personal account

SSH keys

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

Authentication keys

CPE 201A
SHA256:60UhEBoNdGrK114rFopmeNkkL3ec4i3Xv/JmNB9K0IQ
Added on Nov 12, 2025
Never used — Read/write

Check out our guide to connecting to GitHub using SSH keys or troubleshoot common SSH problems.

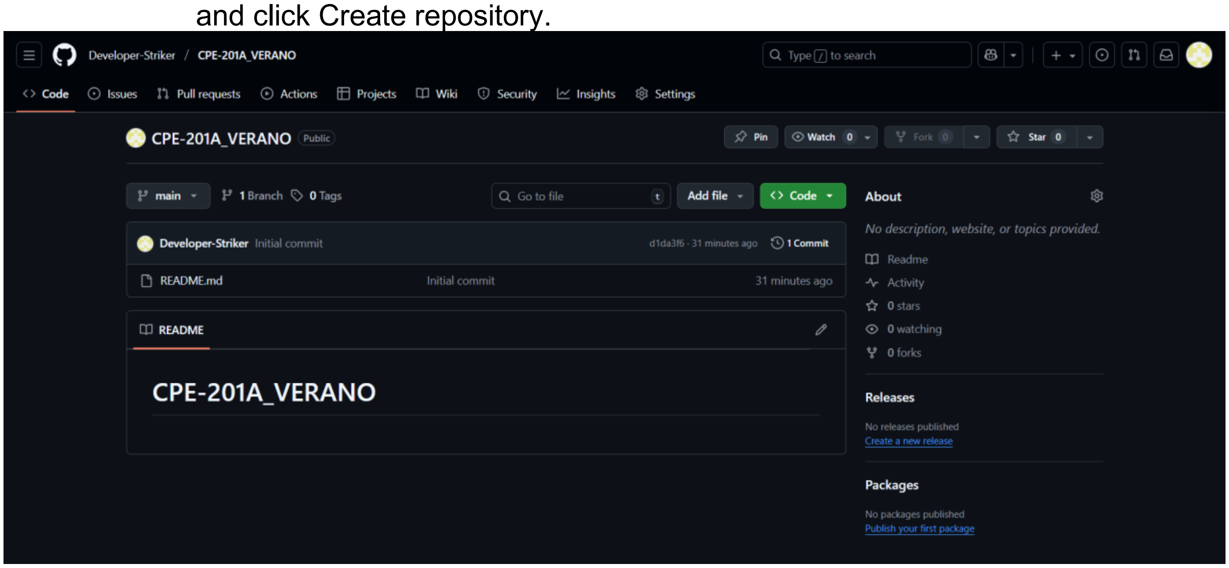
GPG keys

There are no GPG keys associated with your account.

Learn how to generate a GPG key and add it to your account.

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.
3. The version of git installed in your device is the latest. Try issuing the command git --version to know the version installed.
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 CPE201A_yourname, and add description “This repository is only for CPE201A”. Check Add a README file and click Create repository.



Developer-Striker / CPE-201A_VERANO

CPE-201A_VERANO Public

Developer-Striker Initial commit d1da3f6 · 31 minutes ago 1 Commit

README.md Initial commit 31 minutes ago

CPE-201A_VERANO

About

No description, website, or topics provided.

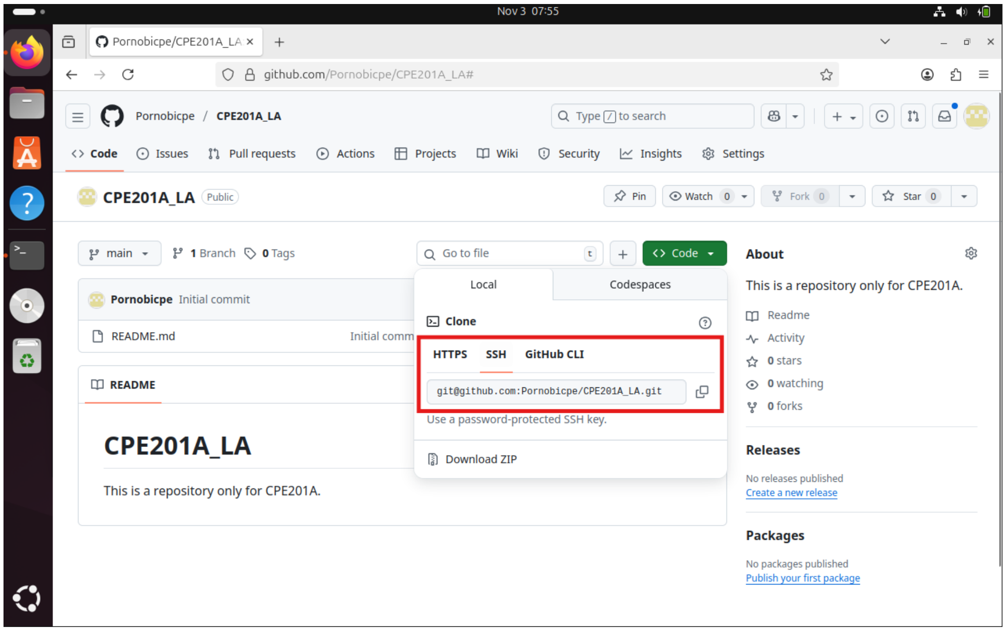
Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

    b. 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.



```
git@github.com:Pornobicpe/CPE201A_LA.git
```

Use a password-protected SSH key.

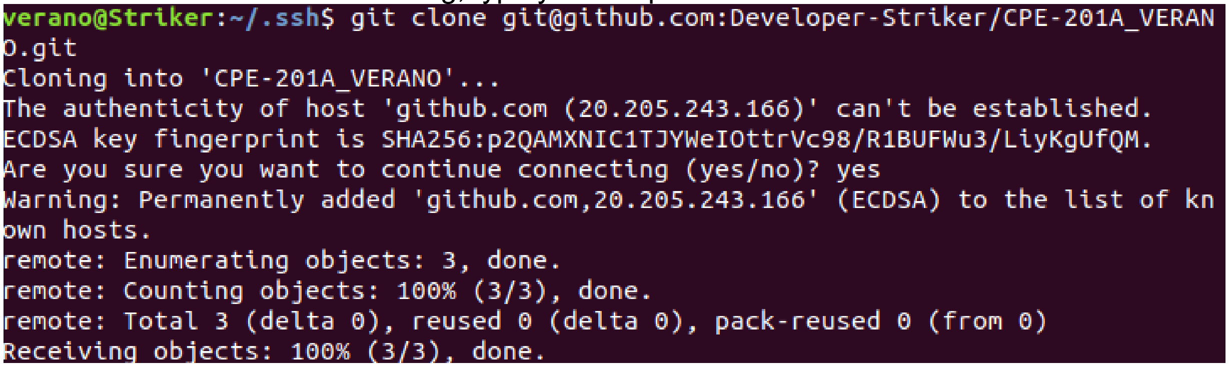
CPE201A_LA

This is a repository only for CPE201A.

No releases published
Create a new release

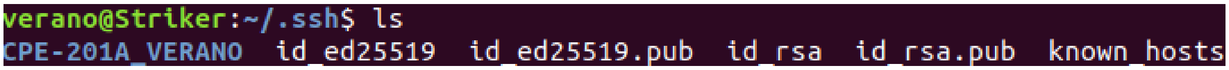
No packages published
Publish your first package

c. Issue the command git clone followed by the copied link. For example, git clone git@github.com:Pornobicpe/CPE201A_yourname.git. When prompted to continue connecting, type yes and press enter.



```
verano@Striker:~/.ssh$ git clone git@github.com:Developer-Striker/CPE-201A_VERAN
O.git
Cloning into 'CPE-201A_VERANO'...
The authenticity of host 'github.com (20.205.243.166)' can't be established.
ECDSA key fingerprint is SHA256:p2QAMXNIC1TJYWeIOttrVc98/R1BUFWu3/LiyKgUfQM.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'github.com,20.205.243.166' (ECDSA) to the list of kn
own hosts.
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
```

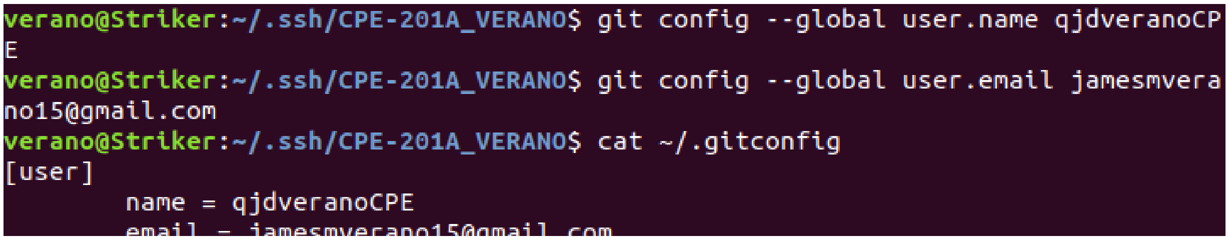
d. To verify that you have cloned the GitHub repository, issue the command ls. Observe that you have the CPE201A_yourname in the list of your directories. Use CD command to go to that directory and LS command to see the file README.md.



```
verano@Striker:~/.ssh$ ls
CPE-201A_VERANO  id_ed25519  id_ed25519.pub  id_rsa  id_rsa.pub  known_hosts
```

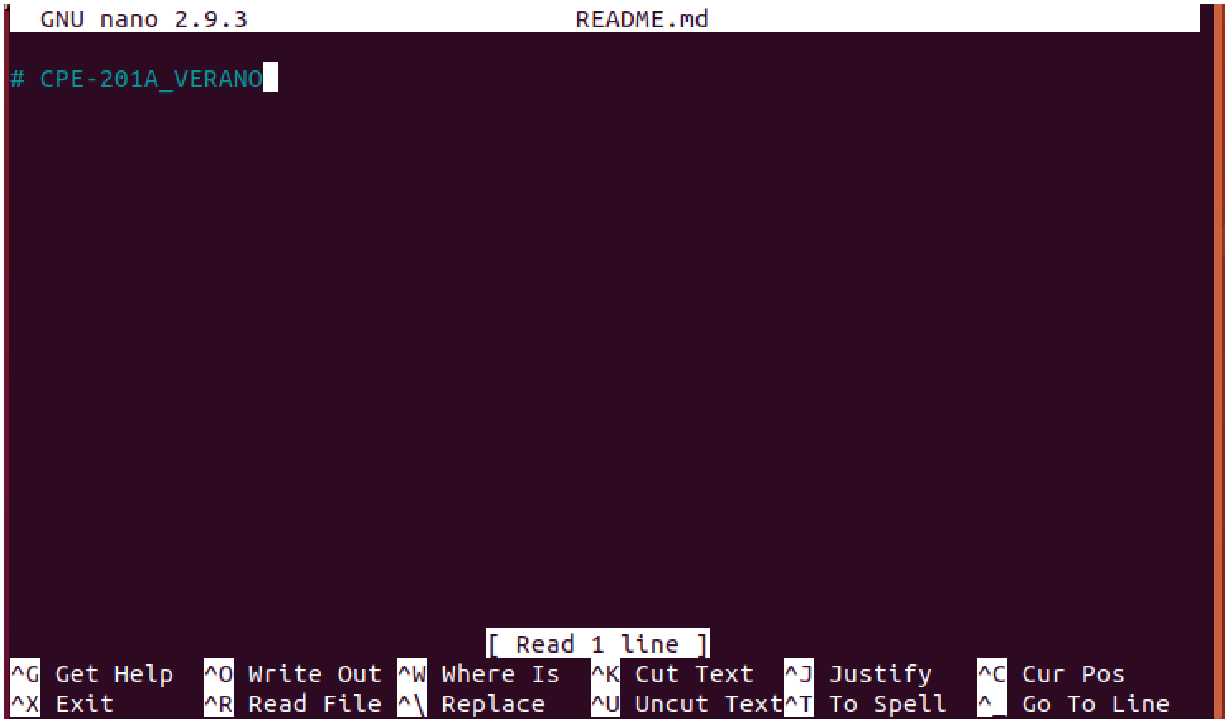
e. Use the following commands to personalize your git.

- 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



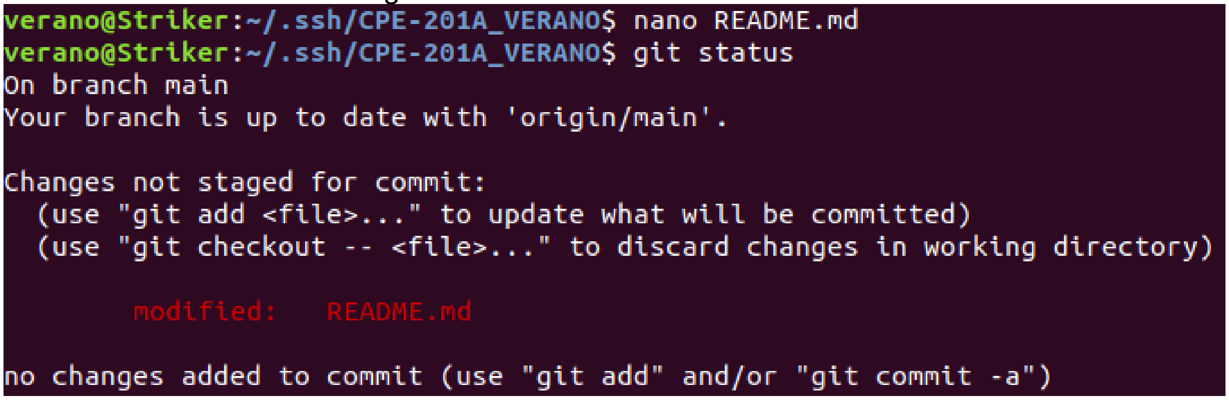
```
verano@Striker:~/.ssh/CPE-201A_VERANO$ git config --global user.name qjdveranoCP
E
verano@Striker:~/.ssh/CPE-201A_VERANO$ git config --global user.email jamesmvera
no15@gmail.com
verano@Striker:~/.ssh/CPE-201A_VERANO$ cat ~/.gitconfig
[user]
        name = qjdveranoCPE
        email = jamesmverano15@gmail.com
```

f. 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.



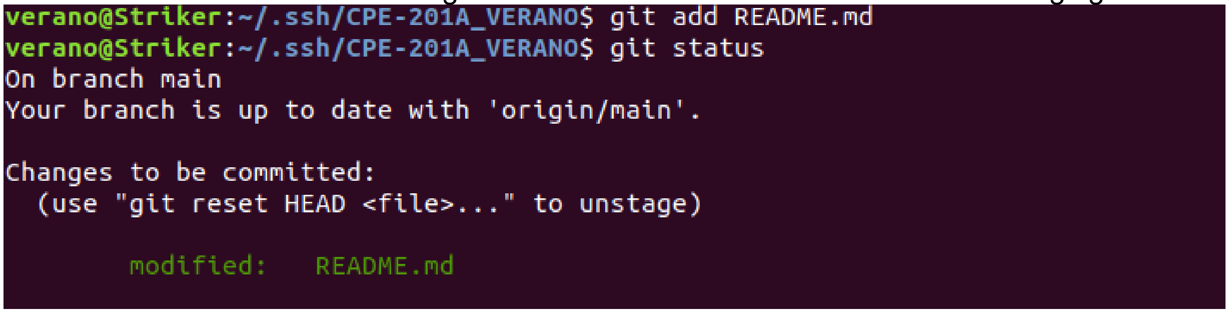
```
  GNU nano 2.9.3                    README.md

# CPE-201A_VERANO

                                 [ Read 1 line ]
^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify     ^C Cur Pos
^X Exit        ^R Read File   ^\ Replace     ^U Uncut Text^T To Spell      ^_ Go To Line
```

g. 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?



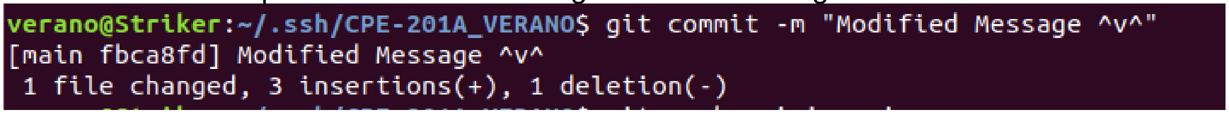
```
verano@Striker:~/.ssh/CPE-201A_VERANO$ nano README.md
verano@Striker:~/.ssh/CPE-201A_VERANO$ git status
On branch main
Your branch is up to date with 'origin/main'.

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")
```

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



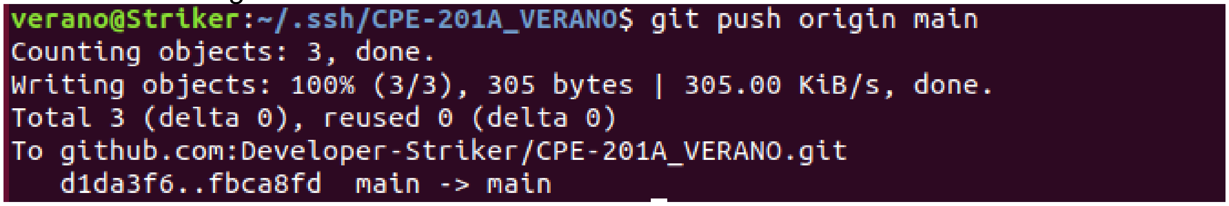
```
verano@Striker:~/.ssh/CPE-201A_VERANO$ git add README.md
verano@Striker:~/.ssh/CPE-201A_VERANO$ git status
On branch main
Your branch is up to date with 'origin/main'.

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

        modified:   README.md
```

i. 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.



```
verano@Striker:~/.ssh/CPE-201A_VERANO$ git commit -m "Modified Message ^v^"
[main fbca8fd] Modified Message ^v^
 1 file changed, 3 insertions(+), 1 deletion(-)
```

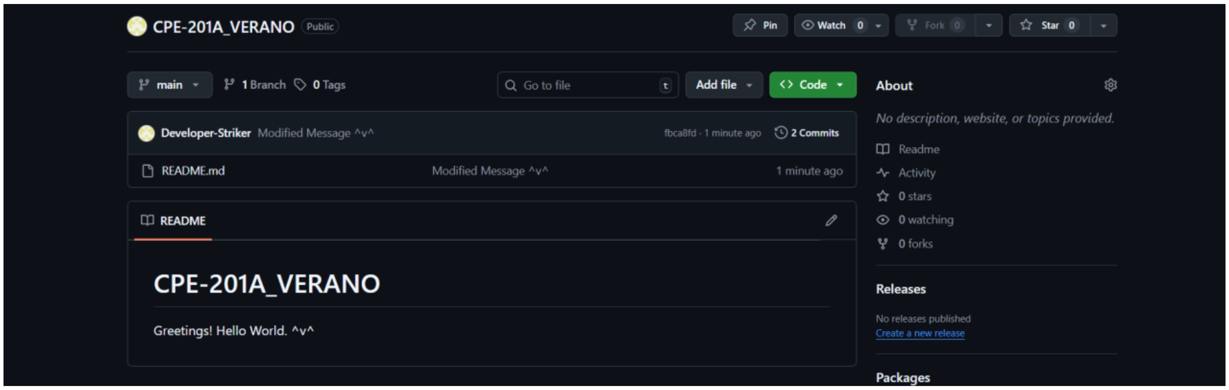
j. Use the command git push <remote><branch> to upload the local repository content to GitHub repository. Pushing means to transfer commits from the local repository to the remote repository. As an example, you may issue git push origin main.



```
verano@Striker:~/.ssh/CPE-201A_VERANO$ git push origin main
Counting objects: 3, done.
Writing objects: 100% (3/3), 305 bytes | 305.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To github.com:Developer-Striker/CPE-201A_VERANO.git
   d1da3f6..fbca8fd  main -> main
```

k. 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 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.

5. Outputs:



Developer-Striker Modified Message ^v^ fbca8fd · 1 minute ago 2 Commits

README.md Modified Message ^v^ 1 minute ago

CPE-201A_VERANO

Greetings! Hello World. ^v^

No description, website, or topics provided.

No releases published

Create a new release

6. Conclusions/Learnings/Analysis:
This activity introduced me to various git and GitHub relationship between modifying files. The commands have guided me to successfully implement the intended outcome. I was able to locate and edit and access using the SSH key, and access folders and files. This activity was a new learning and difficult to execute without proper knowledge and guidance.
7. Assessment Rubric: