Activity No. 14	
SSH Key-Based Authentication and GIT Setup	
Name: Cenar, Marqui	**Date Performed: 11/13/2025**
Course Code: CPE 201A	**Date Submitted: 11/14/2025**
Course Title: Computer System Administration and Troubleshooting	**Instructor: Mr 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

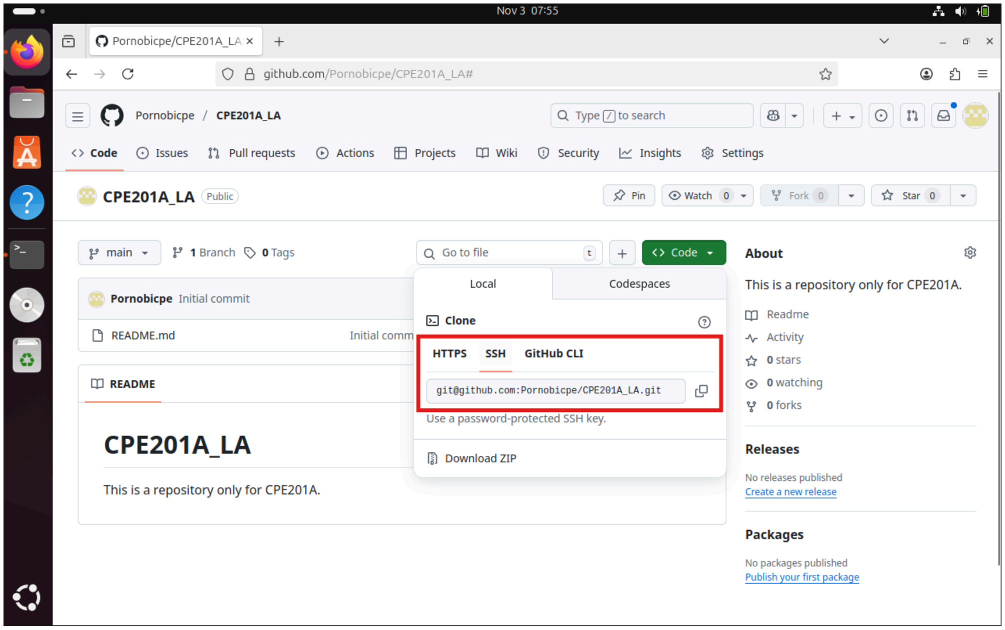
1. Open VirtualBox and start your Ubuntu virtual machine.
2. Log in using your username and password.
3. Open the Terminal.
4. Generate an SSH key pair by typing the following command and pressing Enter:
   `ssh-keygen`
5. Navigate to the SSH directory:
   `cd ~/.ssh`
6. List the files in the directory:
   `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`
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.

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.
    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.

    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.
    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.
    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
    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.

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?

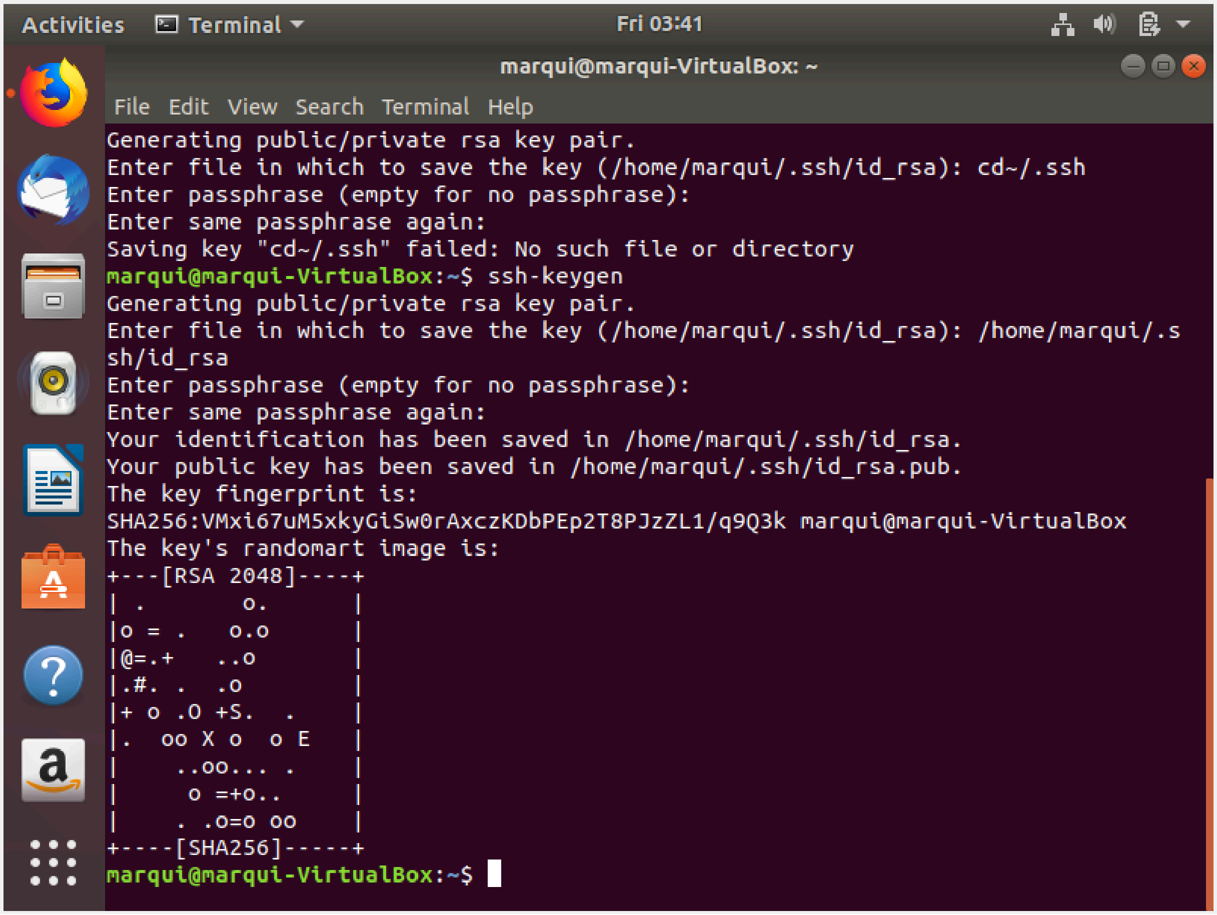
h. Use the command git add README.md to add the file into the staging area.

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.

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.

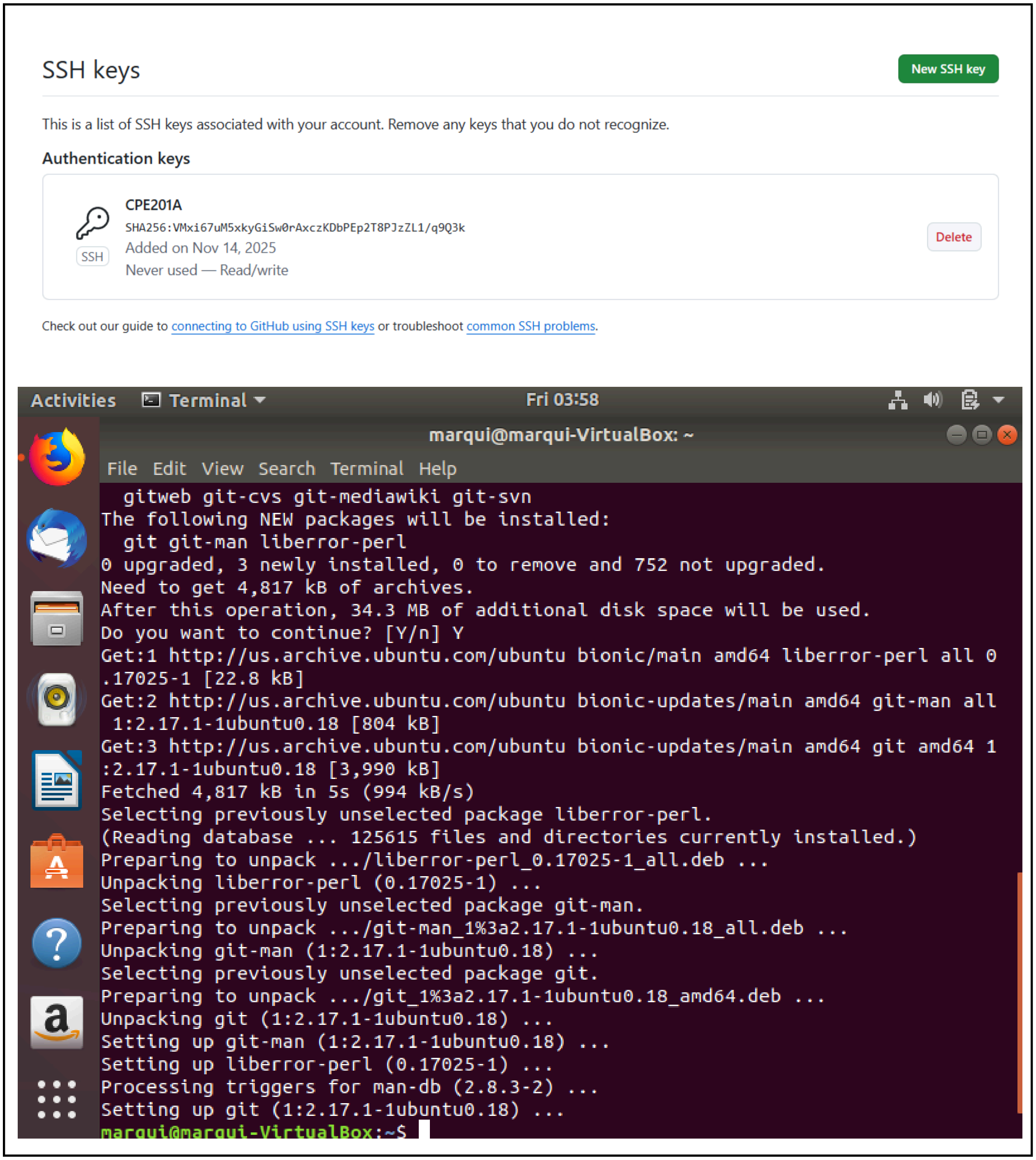
k. On the GitHub repository, verify that the changes have been made to README.md by refreshing the page. Describe the README.md file. You can notice how long was the last commit. It should be some minutes ago and the message you typed on the git commit command should be there. Also, the README.md file should have been edited according to the text you wrote.

5. Outputs:



Activities
Terminal ▾
Fri 03:41
marqui@marqui-VirtualBox: ~
File Edit View Search Terminal Help

```
Generating public/private rsa key pair.
Enter file in which to save the key (/home/marqui/.ssh/id_rsa): cd~/.ssh
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Saving key "cd~/.ssh" failed: No such file or directory
marqui@marqui-VirtualBox:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/marqui/.ssh/id_rsa): /home/marqui/.s
sh/id_rsa
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/marqui/.ssh/id_rsa.
Your public key has been saved in /home/marqui/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:VMxi67uM5xkyGiSw0rAxczKDbPEp2T8PJzZL1/q9Q3k marqui@marqui-VirtualBox
The key's randomart image is:
+---[RSA 2048]----+
| .       o.      |
|o = .   o.o      |
|@=.+   ..o       |
|.#. .  .o        |
|+ o .O +S.  .    |
|.  oo X o  o E   |
|    ..oo... .    |
|     o =+o..     |
|    . .o=o oo    |
+----[SHA256]-----+
marqui@marqui-VirtualBox:~$ 
```



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

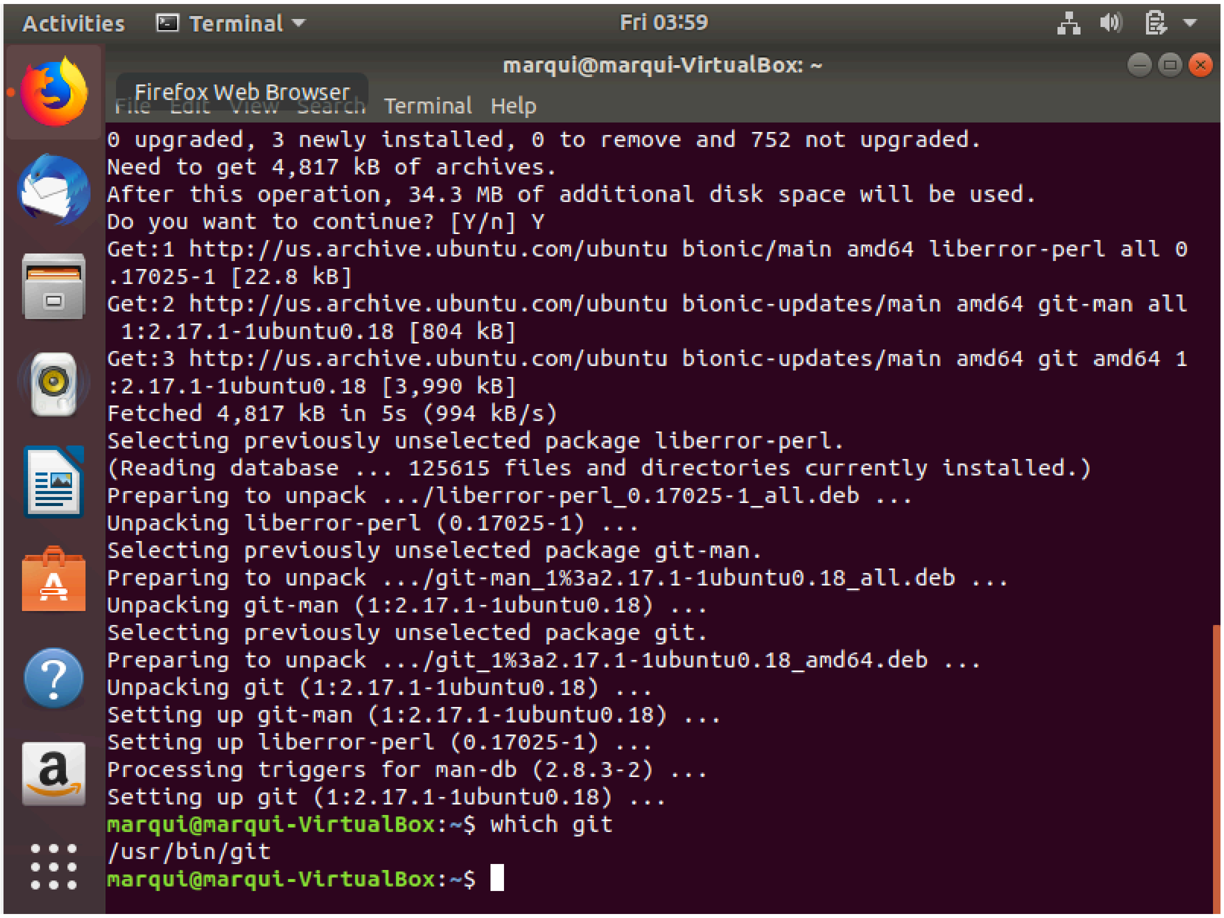
Authentication keys

CPE201A
SHA256:VMxi67uM5xkyGiSw0rAxczKDbPEp2T8PJzZL1/q9Q3k
Added on Nov 14, 2025
Never used — Read/write

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

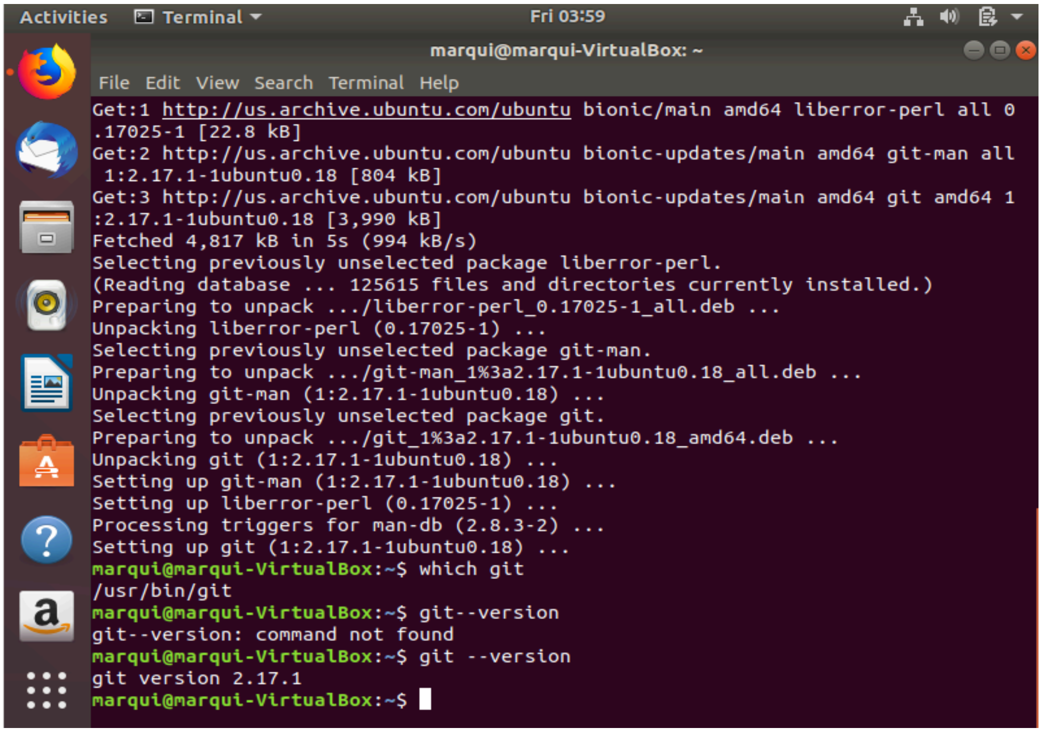
Activities
Terminal
Fri 03:58
marqui@marqui-VirtualBox: ~
File Edit View Search Terminal Help

```
  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 752 not upgraded.
Need to get 4,817 kB of archives.
After this operation, 34.3 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://us.archive.ubuntu.com/ubuntu bionic/main amd64 liberror-perl all 0
.17025-1 [22.8 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu bionic-updates/main amd64 git-man all
 1:2.17.1-1ubuntu0.18 [804 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu bionic-updates/main amd64 git amd64 1
:2.17.1-1ubuntu0.18 [3,990 kB]
Fetched 4,817 kB in 5s (994 kB/s)
Selecting previously unselected package liberror-perl.
(Reading database ... 125615 files and directories currently installed.)
Preparing to unpack .../liberror-perl_0.17025-1_all.deb ...
Unpacking liberror-perl (0.17025-1) ...
Selecting previously unselected package git-man.
Preparing to unpack .../git-man_1%3a2.17.1-1ubuntu0.18_all.deb ...
Unpacking git-man (1:2.17.1-1ubuntu0.18) ...
Selecting previously unselected package git.
Preparing to unpack .../git_1%3a2.17.1-1ubuntu0.18_amd64.deb ...
Unpacking git (1:2.17.1-1ubuntu0.18) ...
Setting up git-man (1:2.17.1-1ubuntu0.18) ...
Setting up liberror-perl (0.17025-1) ...
Processing triggers for man-db (2.8.3-2) ...
Setting up git (1:2.17.1-1ubuntu0.18) ...
marqui@marqui-VirtualBox:~$
```

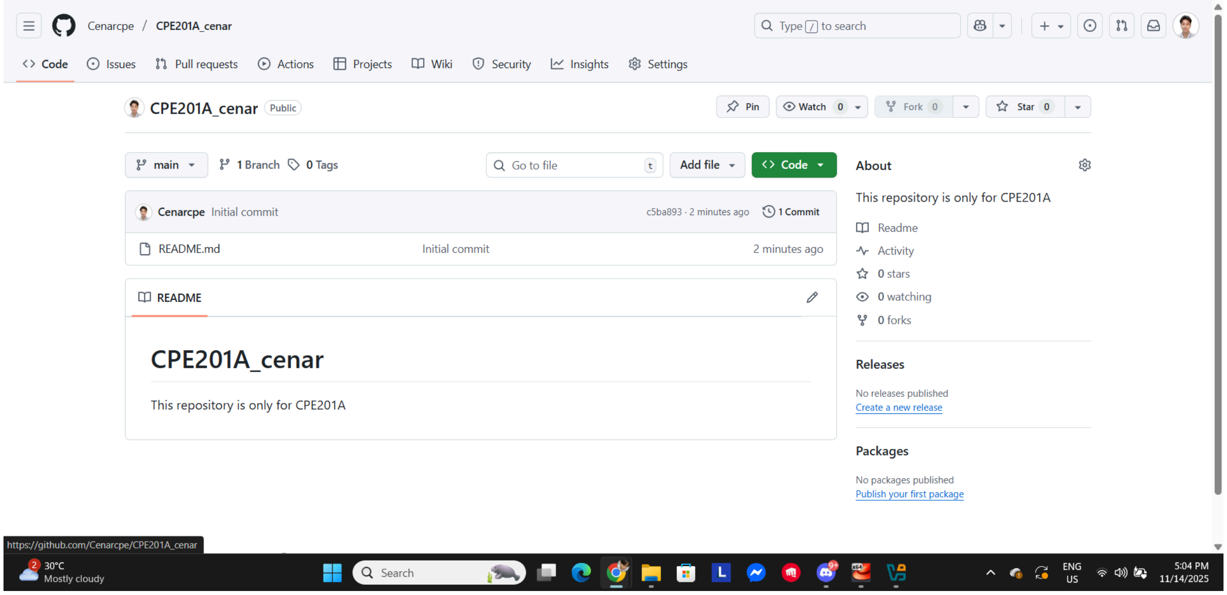


Activities
Terminal
Fri 03:59
marqui@marqui-VirtualBox: ~
Firefox Web Browser
File Edit View Search Terminal Help

```
0 upgraded, 3 newly installed, 0 to remove and 752 not upgraded.
Need to get 4,817 kB of archives.
After this operation, 34.3 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://us.archive.ubuntu.com/ubuntu bionic/main amd64 liberror-perl all 0
.17025-1 [22.8 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu bionic-updates/main amd64 git-man all
 1:2.17.1-1ubuntu0.18 [804 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu bionic-updates/main amd64 git amd64 1
:2.17.1-1ubuntu0.18 [3,990 kB]
Fetched 4,817 kB in 5s (994 kB/s)
Selecting previously unselected package liberror-perl.
(Reading database ... 125615 files and directories currently installed.)
Preparing to unpack .../liberror-perl_0.17025-1_all.deb ...
Unpacking liberror-perl (0.17025-1) ...
Selecting previously unselected package git-man.
Preparing to unpack .../git-man_1%3a2.17.1-1ubuntu0.18_all.deb ...
Unpacking git-man (1:2.17.1-1ubuntu0.18) ...
Selecting previously unselected package git.
Preparing to unpack .../git_1%3a2.17.1-1ubuntu0.18_amd64.deb ...
Unpacking git (1:2.17.1-1ubuntu0.18) ...
Setting up git-man (1:2.17.1-1ubuntu0.18) ...
Setting up liberror-perl (0.17025-1) ...
Processing triggers for man-db (2.8.3-2) ...
Setting up git (1:2.17.1-1ubuntu0.18) ...
marqui@marqui-VirtualBox:~$ which git
/usr/bin/git
marqui@marqui-VirtualBox:~$
```


Activities
Terminal
Fri 03:59
marqui@marqui-VirtualBox: ~
File Edit View Search Terminal Help

```
Get:1 http://us.archive.ubuntu.com/ubuntu bionic/main amd64 liberror-perl all 0
.17025-1 [22.8 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu bionic-updates/main amd64 git-man all
 1:2.17.1-1ubuntu0.18 [804 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu bionic-updates/main amd64 git amd64 1
:2.17.1-1ubuntu0.18 [3,990 kB]
Fetched 4,817 kB in 5s (994 kB/s)
Selecting previously unselected package liberror-perl.
(Reading database ... 125615 files and directories currently installed.)
Preparing to unpack .../liberror-perl_0.17025-1_all.deb ...
Unpacking liberror-perl (0.17025-1) ...
Selecting previously unselected package git-man.
Preparing to unpack .../git-man_1%3a2.17.1-1ubuntu0.18_all.deb ...
Unpacking git-man (1:2.17.1-1ubuntu0.18) ...
Selecting previously unselected package git.
Preparing to unpack .../git_1%3a2.17.1-1ubuntu0.18_amd64.deb ...
Unpacking git (1:2.17.1-1ubuntu0.18) ...
Setting up git-man (1:2.17.1-1ubuntu0.18) ...
Setting up liberror-perl (0.17025-1) ...
Processing triggers for man-db (2.8.3-2) ...
Setting up git (1:2.17.1-1ubuntu0.18) ...
marqui@marqui-VirtualBox:~$ which git
/usr/bin/git
marqui@marqui-VirtualBox:~$ git--version
git--version: command not found
marqui@marqui-VirtualBox:~$ git --version
git version 2.17.1
marqui@marqui-VirtualBox:~$ 
```



CPE201A_cenar

Cenarcpe Initial commit c5ba893 · 2 minutes ago 1 Commit

README.md	Initial commit	2 minutes ago

README

CPE201A_cenar

This repository is only for CPE201A

About

This repository is only for CPE201A

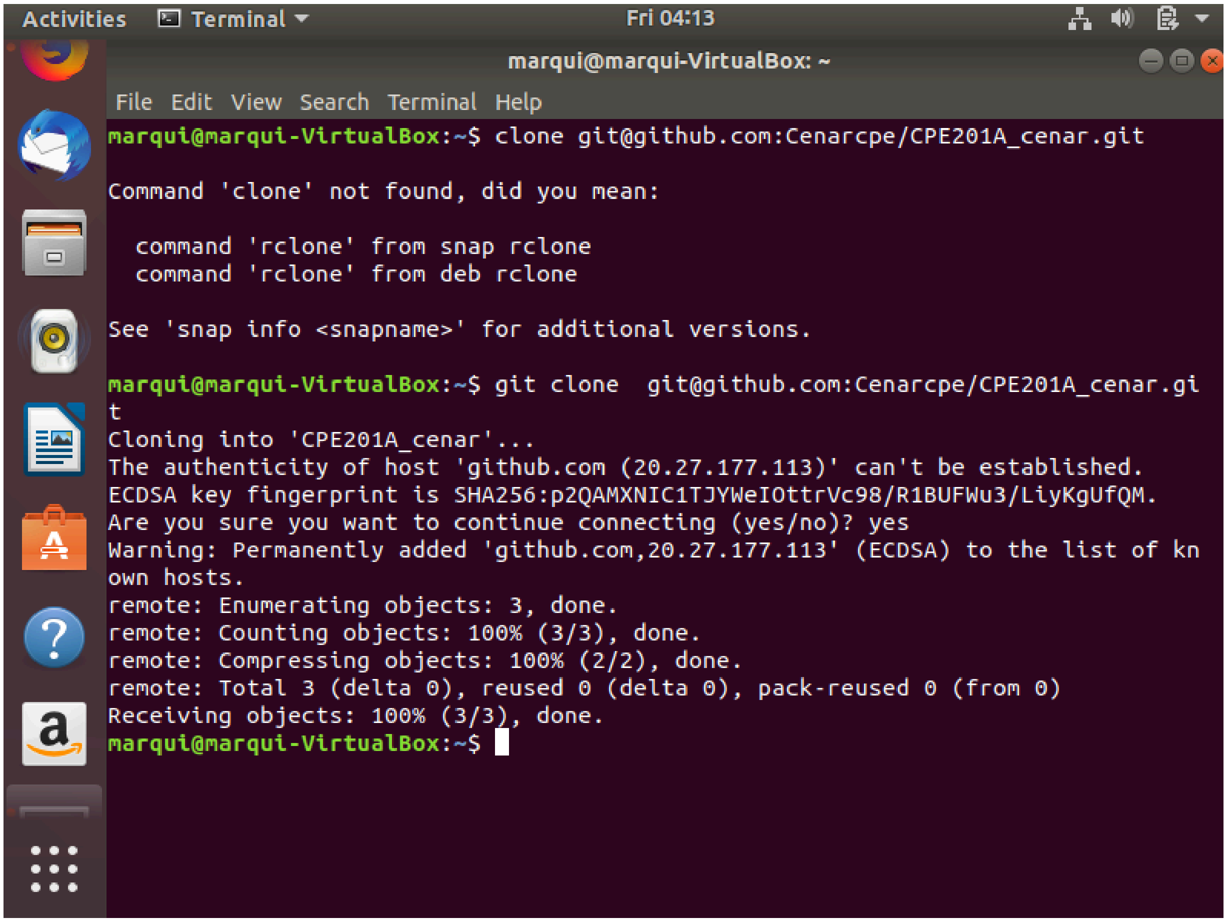
Readme
Activity
0 stars
0 watching
0 forks

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package


Activities
Terminal
Fri 04:13
marqui@marqui-VirtualBox: ~
File Edit View Search Terminal Help

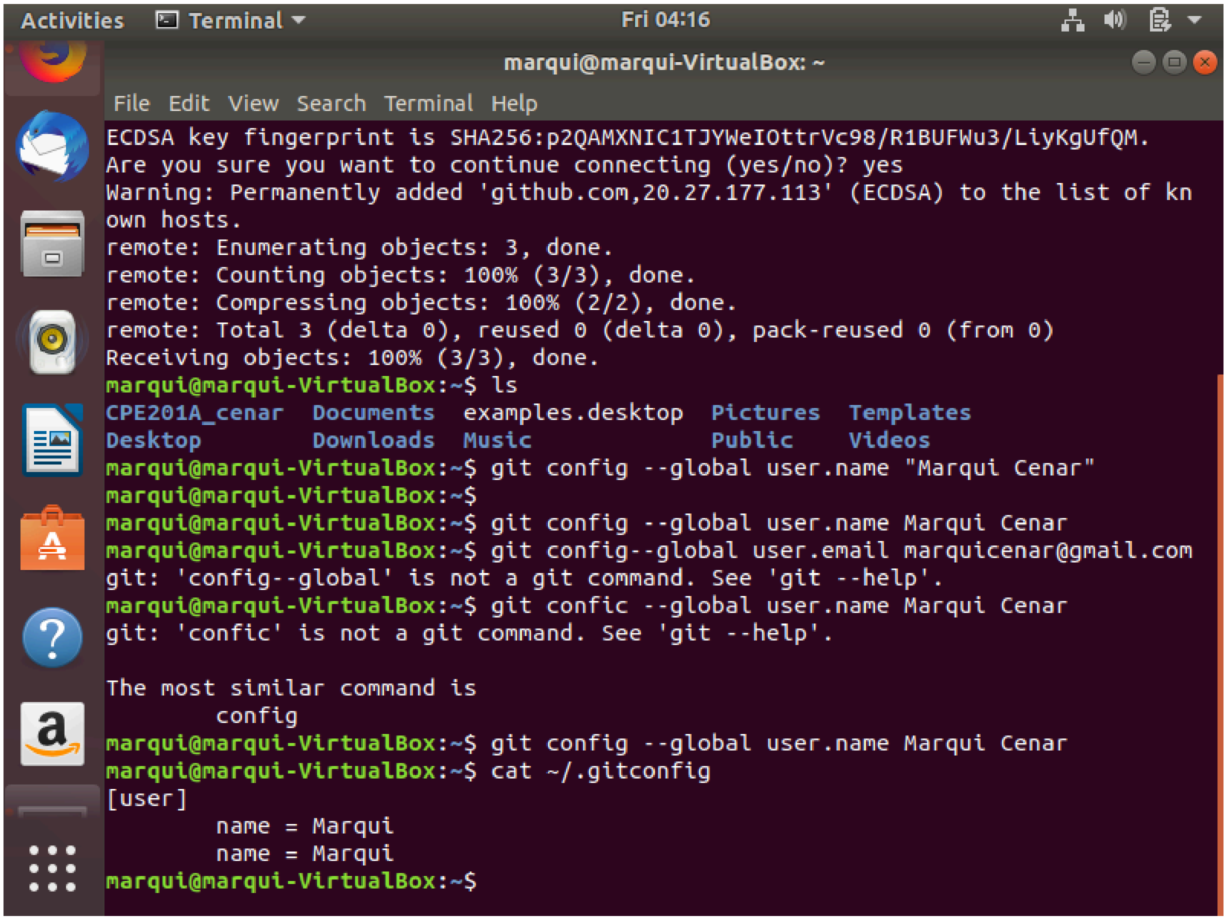
```
marqui@marqui-VirtualBox:~$ clone git@github.com:Cenarcpe/CPE201A_cenar.git

Command 'clone' not found, did you mean:

  command 'rclone' from snap rclone
  command 'rclone' from deb rclone

See 'snap info <snapname>' for additional versions.

marqui@marqui-VirtualBox:~$ git clone  git@github.com:Cenarcpe/CPE201A_cenar.gi
t
Cloning into 'CPE201A_cenar'...
The authenticity of host 'github.com (20.27.177.113)' 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,20.27.177.113' (ECDSA) to the list of kn
own hosts.
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
marqui@marqui-VirtualBox:~$ 
```


Activities
Terminal
Fri 04:16
marqui@marqui-VirtualBox: ~
File Edit View Search Terminal Help

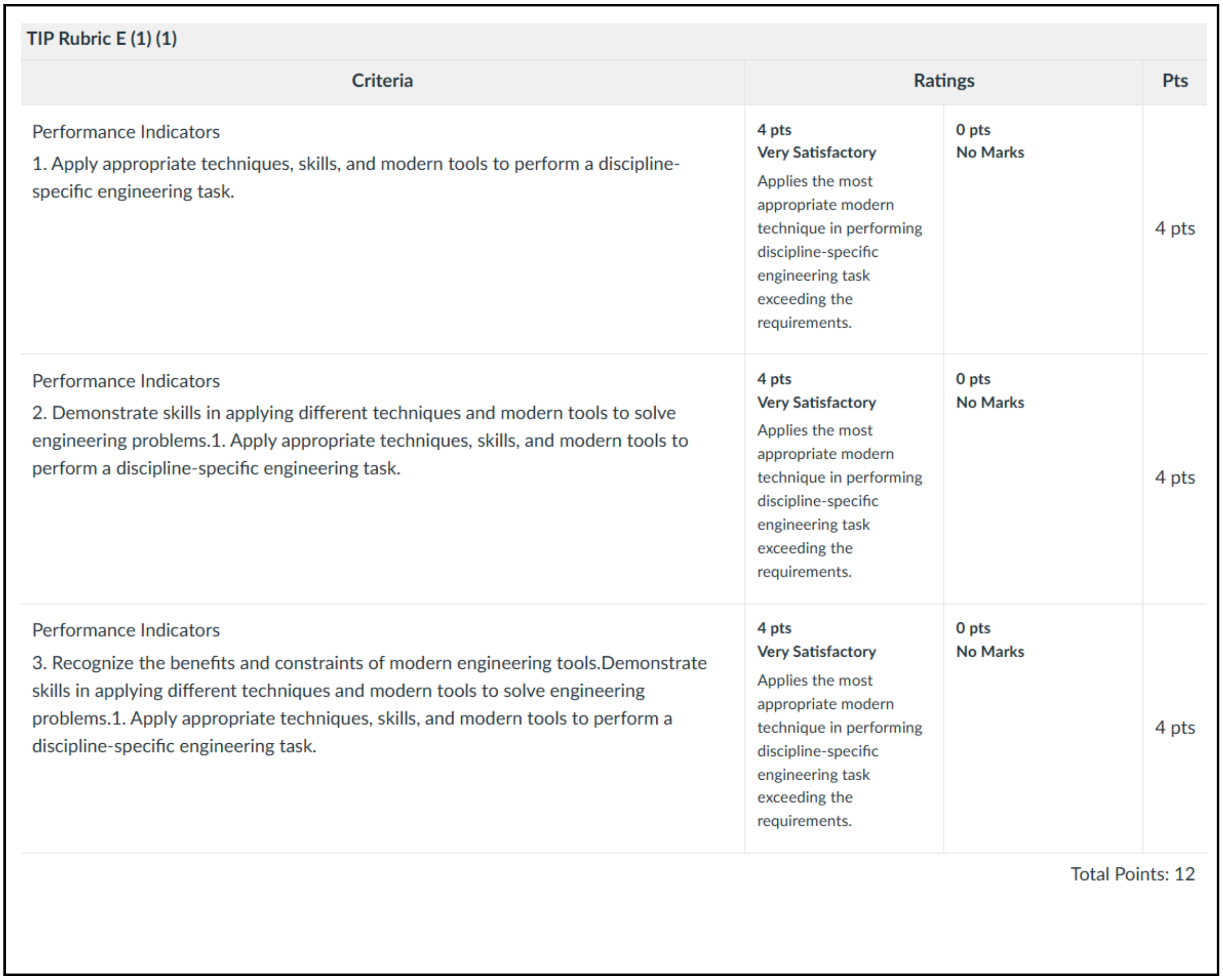
```
ECDSA key fingerprint is SHA256:p2QAMXNIC1TJYWeIOttrVc98/R1BUFWu3/LiyKgUfQM.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'github.com,20.27.177.113' (ECDSA) to the list of kn
own hosts.
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
marqui@marqui-VirtualBox:~$ ls
CPE201A_cenar  Documents  examples.desktop  Pictures  Templates
Desktop        Downloads  Music             Public    Videos
marqui@marqui-VirtualBox:~$ git config --global user.name "Marqui Cenar"
marqui@marqui-VirtualBox:~$
marqui@marqui-VirtualBox:~$ git config --global user.name Marqui Cenar
marqui@marqui-VirtualBox:~$ git config--global user.email marquicenar@gmail.com
git: 'config--global' is not a git command. See 'git --help'.
marqui@marqui-VirtualBox:~$ git confic --global user.name Marqui Cenar
git: 'confic' is not a git command. See 'git --help'.

The most similar command is
        config
marqui@marqui-VirtualBox:~$ git config --global user.name Marqui Cenar
marqui@marqui-VirtualBox:~$ cat ~/.gitconfig
[user]
        name = Marqui
        name = Marqui
marqui@marqui-VirtualBox:~$
```

6. Conclusions/Learnings/Analysis:

In this activity, I learned how to use some git commands to manage a project and make a GitHub repository. I checked the status of my files, prepared them, cloned, and personalized it. This activity helped me understand how git tracks changes and how to publish them online using github.

7. Assessment Rubric:



TIP Rubric E (1) (1)

Criteria	Ratings		Pts
Performance Indicators 1. Apply appropriate techniques, skills, and modern tools to perform a discipline-specific engineering task.	**4 pts** **Very Satisfactory** Applies the most appropriate modern technique in performing discipline-specific engineering task exceeding the requirements.	**0 pts** **No Marks**	4 pts
Performance Indicators 2. Demonstrate skills in applying different techniques and modern tools to solve engineering problems.1. Apply appropriate techniques, skills, and modern tools to perform a discipline-specific engineering task.	**4 pts** **Very Satisfactory** Applies the most appropriate modern technique in performing discipline-specific engineering task exceeding the requirements.	**0 pts** **No Marks**	4 pts
Performance Indicators 3. Recognize the benefits and constraints of modern engineering tools.Demonstrate skills in applying different techniques and modern tools to solve engineering problems.1. Apply appropriate techniques, skills, and modern tools to perform a discipline-specific engineering task.	**4 pts** **Very Satisfactory** Applies the most appropriate modern technique in performing discipline-specific engineering task exceeding the requirements.	**0 pts** **No Marks**	4 pts

Total Points: 12