Activity No. 14	
SSH Key-Based Authentication and GIT Setup	
Name: Ralph Angelov F. Braganza	**Date Performed: 11/12/2025**
Course Code: CPE 201A	**Date Submitted: 11/12/2025**
Course Title: Computer System Administration and Troubleshooting	**Instructor: Engr. 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.

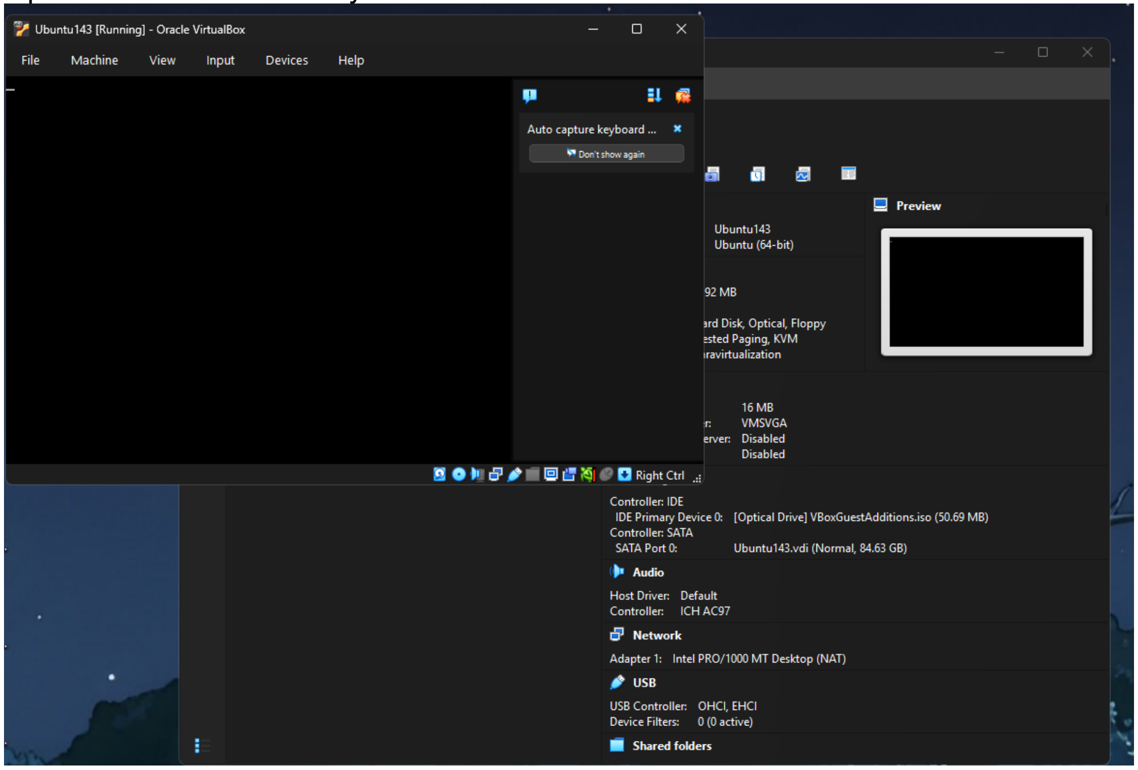
Set up Git
At the heart of GitHub is an open-source version control system (VCS) called Git. Git is responsible for everything GitHub-related that happens locally on your computer. To use Git on the command line, you'll need to download, install, and configure Git on your computer. You can also install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

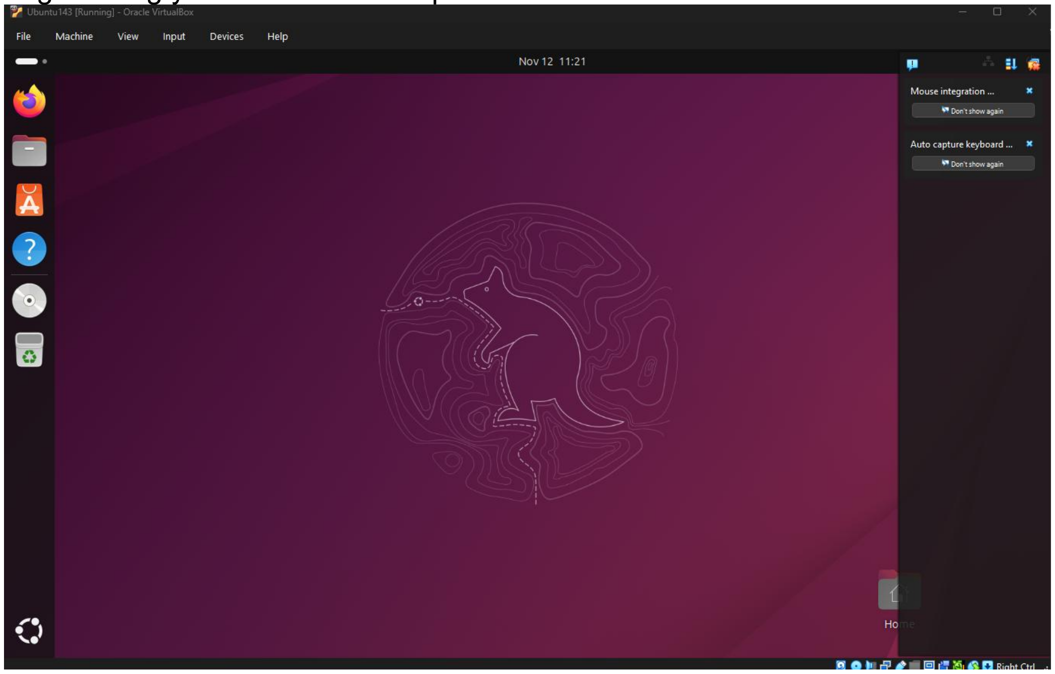
Task 1: Create an SSH Key Pair for User Authentication

1. Open VirtualBox and start your Ubuntu virtual machine.

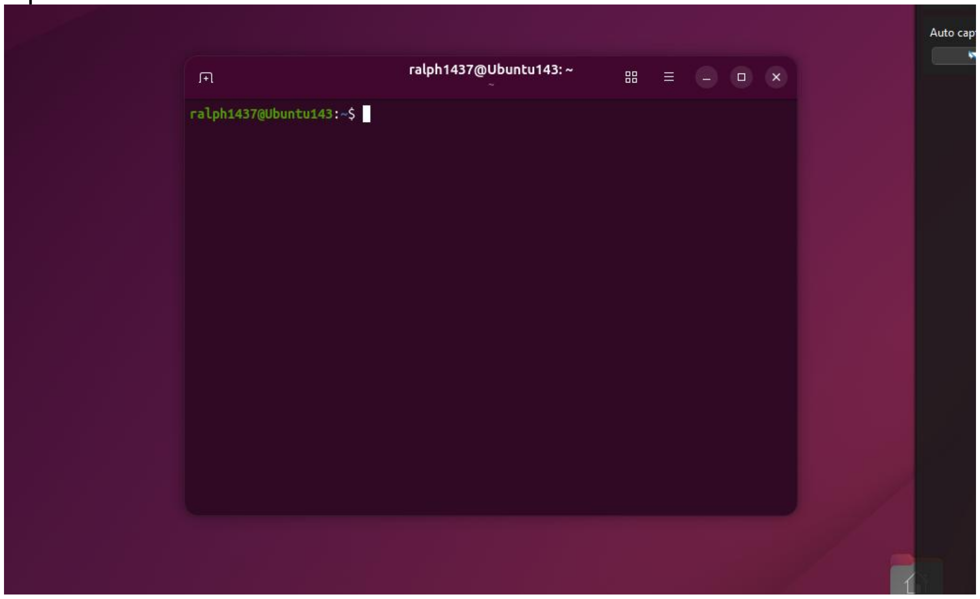

Ubuntu143 [Running] - Oracle VirtualBox
File
Machine
View
Input
Devices
Help
Auto capture keyboard ...
Don't show again
Right Ctrl
Preview

Ubuntu143
Ubuntu (64-bit)
92 MB
ard Disk, Optical, Floppy
ested Paging, KVM
aravirtualization
16 MB
r: VMSVGA
erver: Disabled
Disabled
Controller: IDE
IDE Primary Device 0: [Optical Drive] VBoxGuestAdditions.iso (50.69 MB)
Controller: SATA
SATA Port 0: Ubuntu143.vdi (Normal, 84.63 GB)
Audio
Host Driver: Default
Controller: ICH AC97
Network
Adapter 1: Intel PRO/1000 MT Desktop (NAT)
USB
USB Controller: OHCI, EHCI
Device Filters: 0 (0 active)
Shared folders

2. Log in using your username and password.

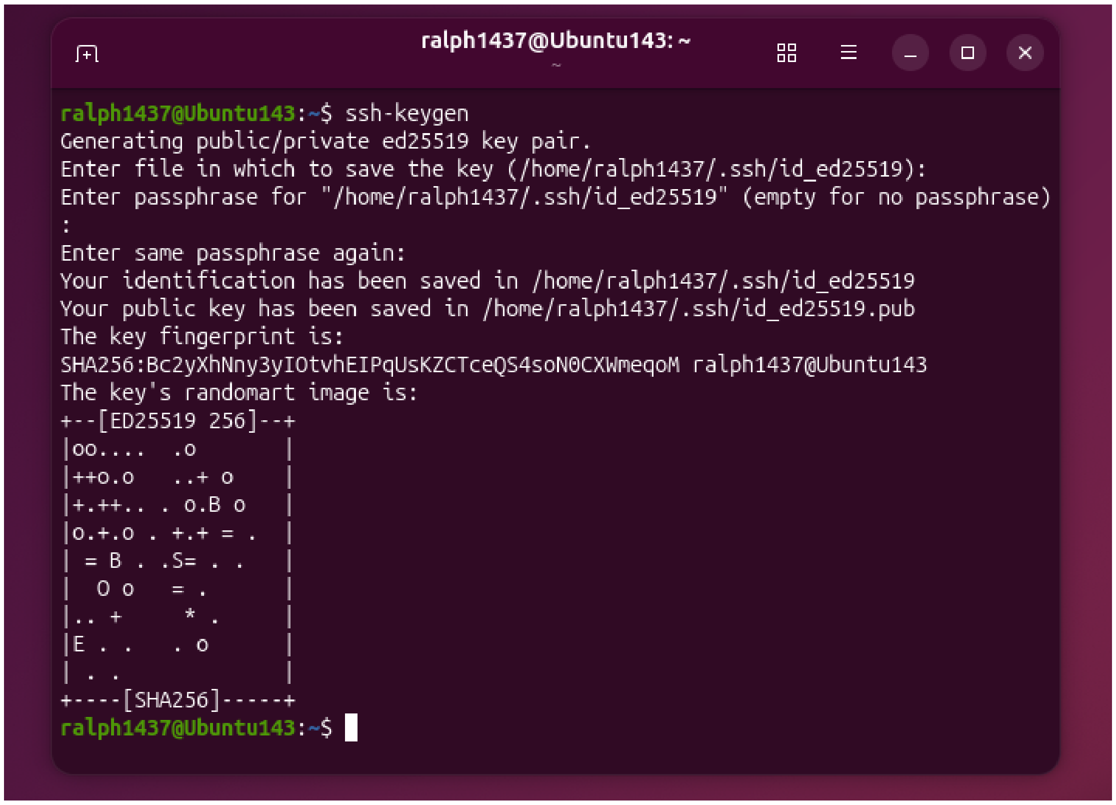


3. Open the Terminal.



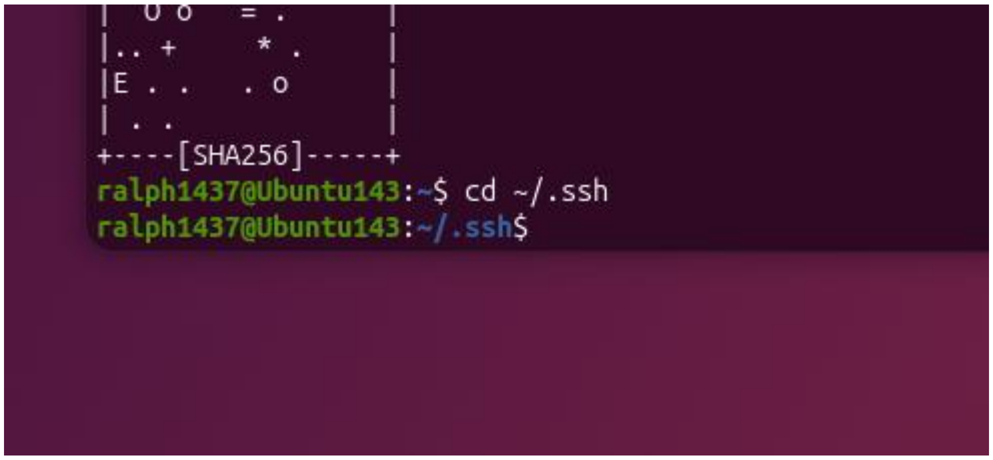
```
ralph1437@Ubuntu143:~$
```

4. Generate an SSH key pair by typing the following command and pressing Enter:
   ssh-keygen



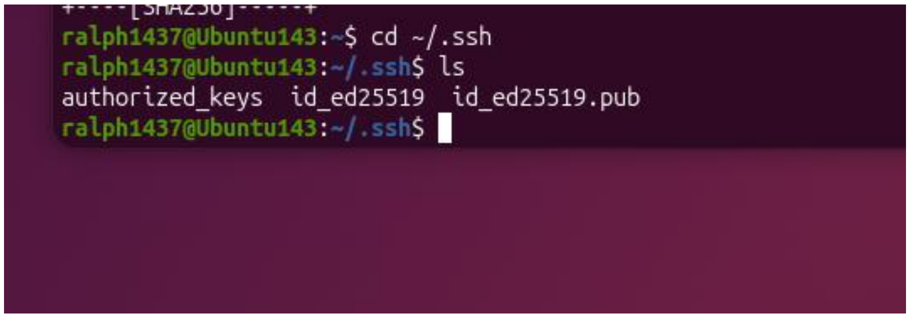
```
ralph1437@Ubuntu143:~$ ssh-keygen
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/ralph1437/.ssh/id_ed25519):
Enter passphrase for "/home/ralph1437/.ssh/id_ed25519" (empty for no passphrase)
:
Enter same passphrase again:
Your identification has been saved in /home/ralph1437/.ssh/id_ed25519
Your public key has been saved in /home/ralph1437/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:Bc2yXhNny3yIOtvhEIPqUsKZCTceQS4soN0CXWmeqoM ralph1437@Ubuntu143
The key's randomart image is:
+--[ED25519 256]--+
|oo....  .o       |
|++o.o   ..+ o    |
|+.++.. . o.B o   |
|o.+.o . +.+ = .  |
| = B . .S= . .   |
|  O o   = .      |
|.. +     * .     |
|E . .   . o      |
| . .             |
+----[SHA256]-----+
ralph1437@Ubuntu143:~$
```

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



```
|  O o   = .      |
|.. +     * .     |
|E . .   . o      |
| . .             |
+----[SHA256]-----+
ralph1437@Ubuntu143:~$ cd ~/.ssh
ralph1437@Ubuntu143:~/.ssh$
```

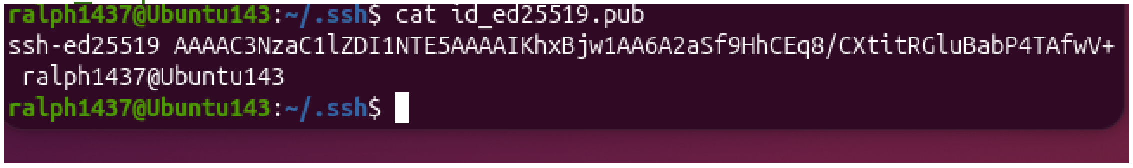
6. List the files in the directory:
   ls



```
+----[SHA256]-----+
ralph1437@Ubuntu143:~$ cd ~/.ssh
ralph1437@Ubuntu143:~/.ssh$ ls
authorized_keys  id_ed25519  id_ed25519.pub
ralph1437@Ubuntu143:~/.ssh$
```

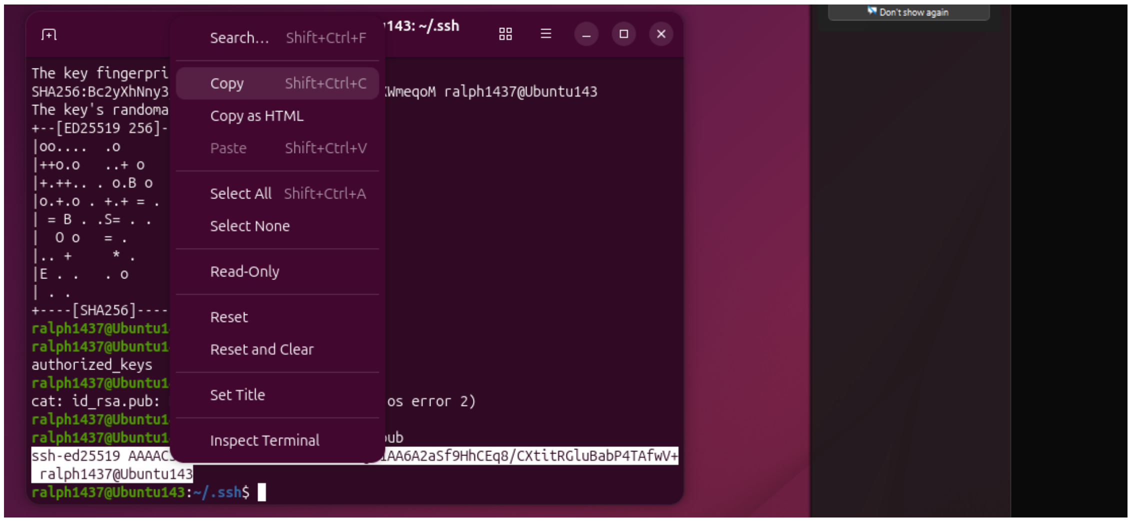
Look for a file ending with .pub this is your public key.

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



```
ralph1437@Ubuntu143:~/.ssh$ cat id_ed25519.pub
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIKhxBjw1AA6A2aSf9HhCEq8/CXtitRGluBabP4TAfwV+
 ralph1437@Ubuntu143
ralph1437@Ubuntu143:~/.ssh$
```

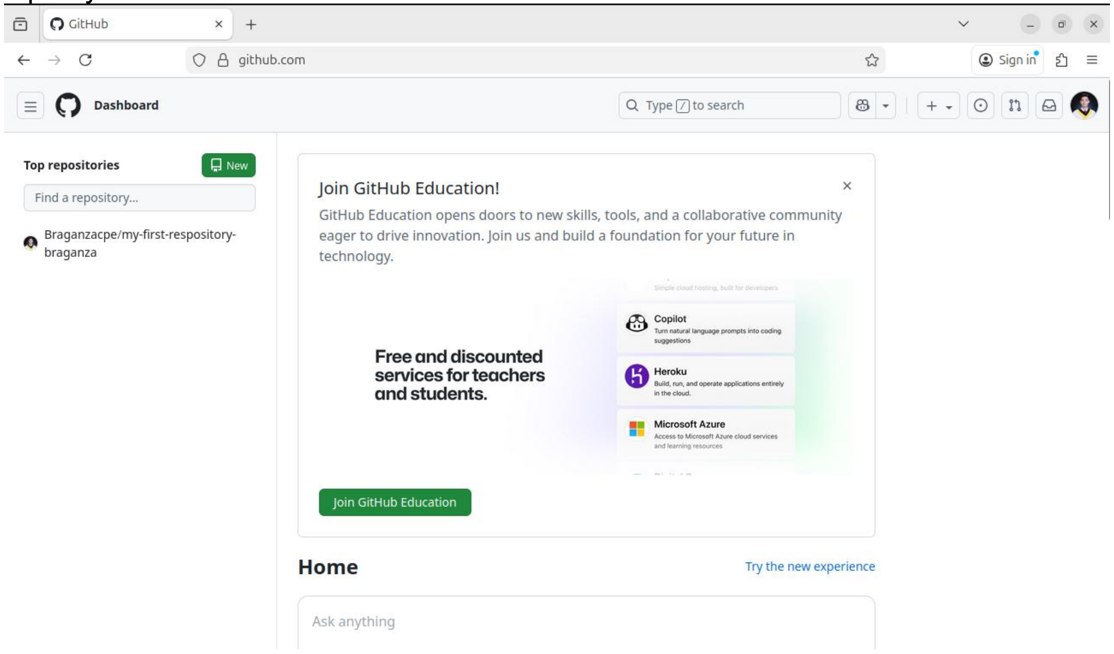
8. Copy the entire output: this is your SSH public key, which you can use for authentication.



```
The key fingerpri                                        WmeqoM ralph1437@Ubuntu143
SHA256:Bc2yXhNny3
The key's randoma
+--[ED25519 256]-
|oo....   .o
|++o.o   ..+ o
|+.++.. . o.B o
|o.+.o . +.+ = .
| = B . .S= . .
|  O o   = .
|.. +     * .
|E . .   . o
| . .
+----[SHA256]----
ralph1437@Ubuntu1
ralph1437@Ubuntu1
authorized_keys
ralph1437@Ubuntu1
cat: id_rsa.pub:                                         os error 2)
ralph1437@Ubuntu1
ralph1437@Ubuntu1                                        ub
ssh-ed25519 AAAAC                                        AA6A2aSf9HhCEq8/CXtitRGluBabP4TAfwV+
 ralph1437@Ubuntu143
ralph1437@Ubuntu143:~/.ssh$
```

Task 2: Copying the Public Key to Remote Servers

1. Open your GitHub account in a web browser.

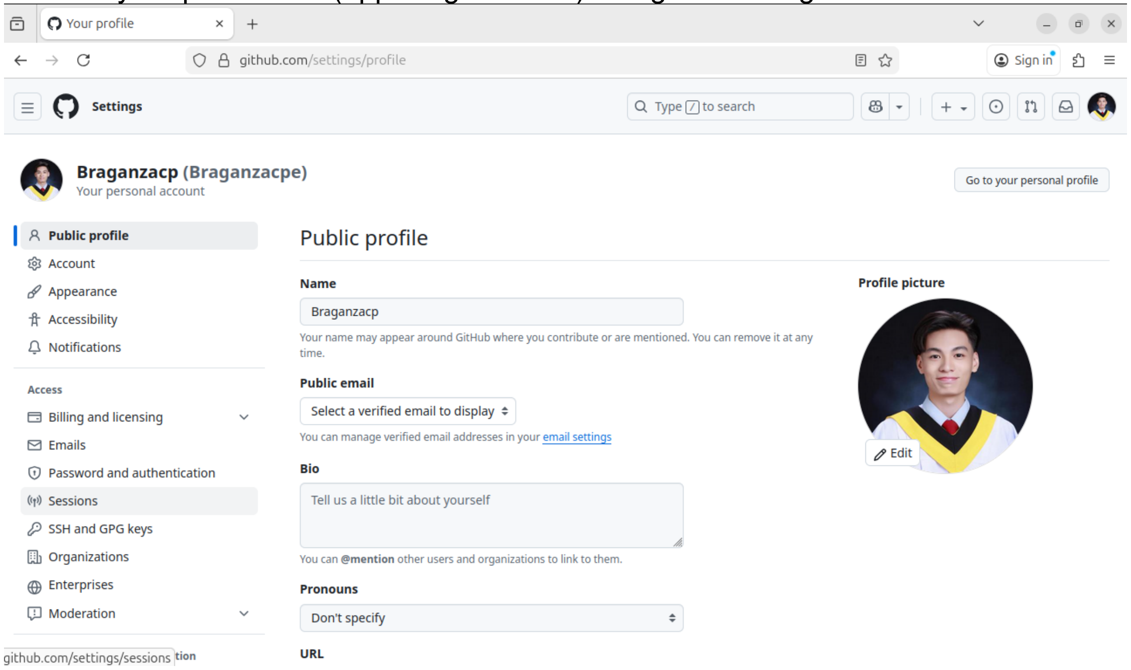


Join GitHub Education!

GitHub Education opens doors to new skills, tools, and a collaborative community eager to drive innovation. Join us and build a foundation for your future in technology.

Free and discounted services for teachers and students.

2. Click on your profile icon (upper-right corner) and go to Settings.



Your profile

github.com/settings/profile

Settings

Type / to search

Braganzacp (Braganzacpe)
Your personal account

Go to your personal profile

- Public profile
- Account
- Appearance
- Accessibility
- Notifications

Access

- Billing and licensing
- Emails
- Password and authentication
- Sessions
- SSH and GPG keys
- Organizations
- Enterprises
- Moderation

Public profile

Name
Braganzacp
Your name may appear around GitHub where you contribute or are mentioned. You can remove it at any time.

Public email
Select a verified email to display
You can manage verified email addresses in your email settings

Bio
Tell us a little bit about yourself
You can @mention other users and organizations to link to them.

Pronouns
Don't specify

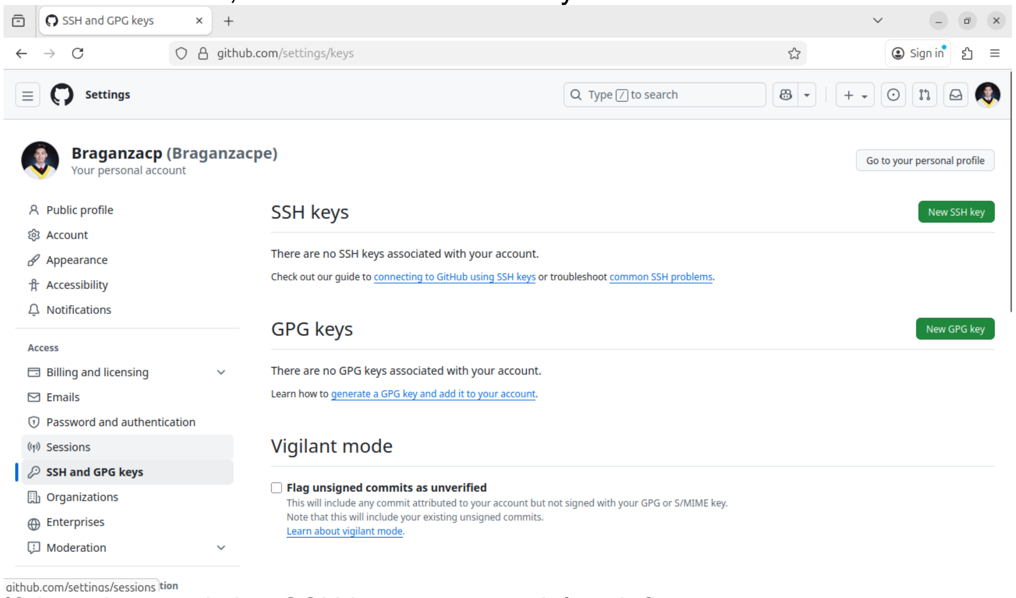
URL

Profile picture

Edit

github.com/settings/sessions

3. In the left sidebar, select SSH and GPG keys.



SSH and GPG keys

github.com/settings/keys

Settings

Type / to search

Braganzacp (Braganzacpe)
Your personal account

Go to your personal profile

- Public profile
- Account
- Appearance
- Accessibility
- Notifications

Access

- Billing and licensing
- Emails
- Password and authentication
- Sessions
- SSH and GPG keys
- Organizations
- Enterprises
- Moderation

SSH keys

New SSH key

There are no SSH keys associated with your account.

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

GPG keys

New GPG key

There are no GPG keys associated with your account.

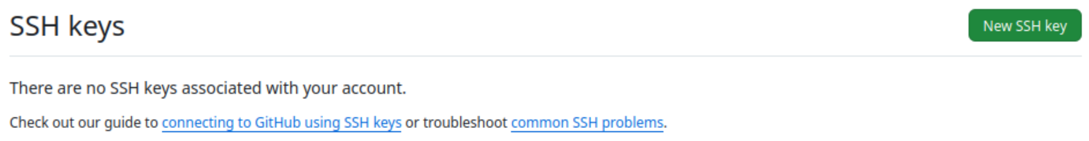
Learn how to generate a GPG key and add it to your account.

Vigilant mode

☐ Flag unsigned commits as unverified
This will include any commit attributed to your account but not signed with your GPG or S/MIME key.
Note that this will include your existing unsigned commits.
Learn about vigilant mode.

github.com/settings/sessions

4. If there is an existing SSH key, you may delete it first.



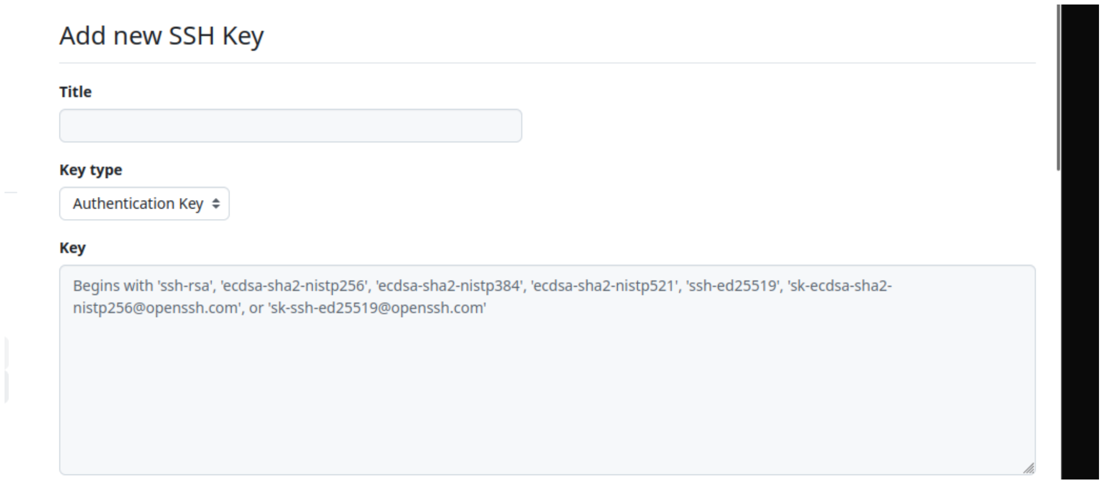
SSH keys

New SSH key

There are no SSH keys associated with your account.

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

5. Click the “New SSH key” button.



Add new SSH Key

Title

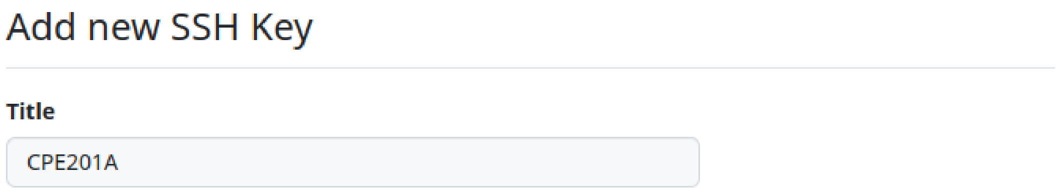
Key type

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'

6. Enter CPE201A as the Title.

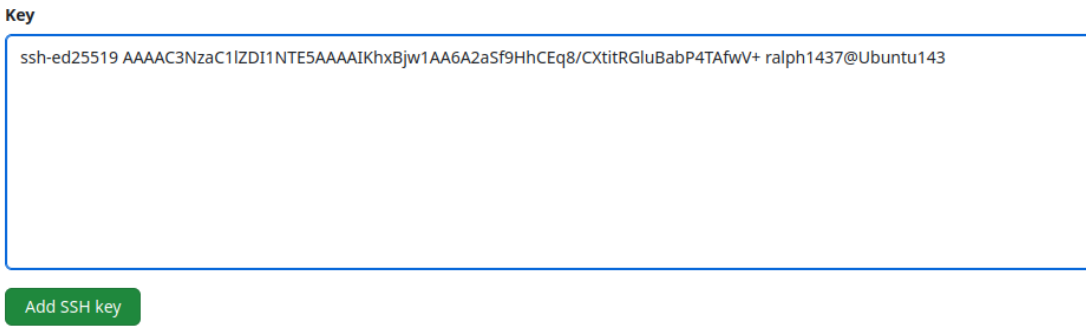


Add new SSH Key

Title

CPE201A

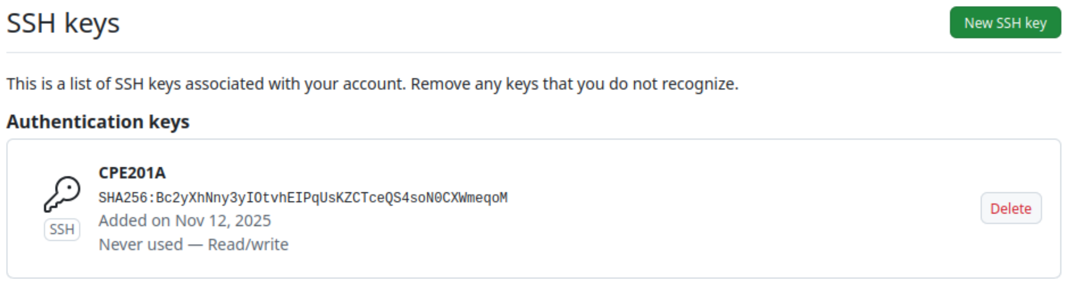
7. In the Key field, paste the SSH public key that you copied from the terminal in Task 1.



Key

```
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIKhxBjw1AA6A2aSf9HhCEq8/CXtitRGluBabP4TAfwV+ ralph1437@Ubuntu143
```

8. Click “Add SSH key” to save your new key.



SSH keys

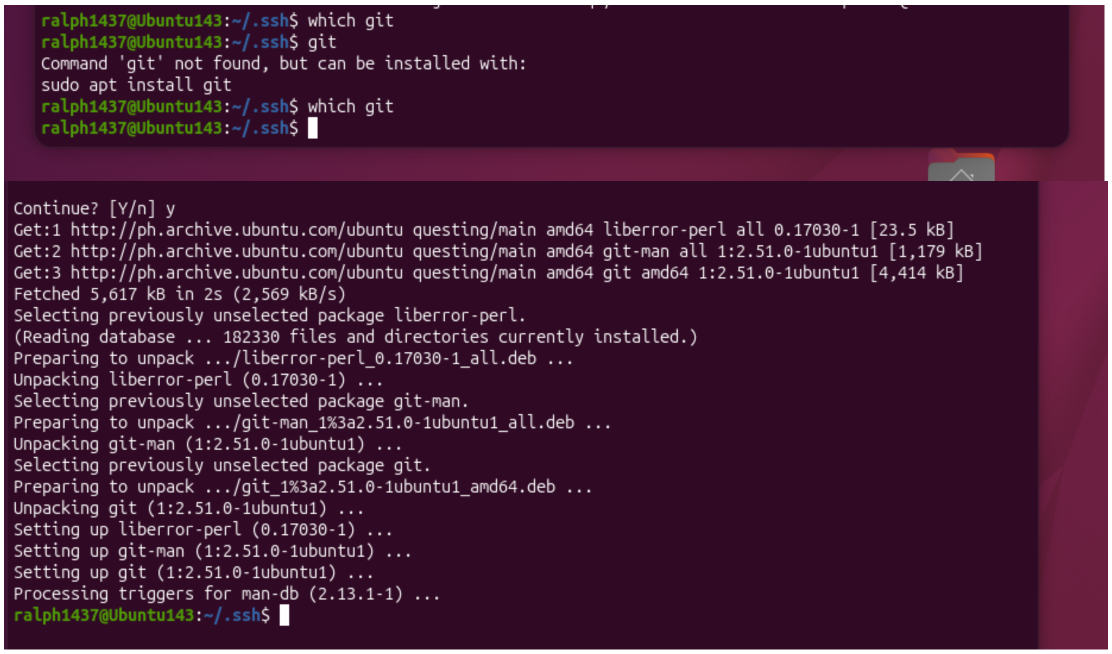
This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

Authentication keys

CPE201A
SHA256:Bc2yXhNny3yIOtvhEIPqUsKZCTceQS4soN0CXWmeqoM
Added on Nov 12, 2025
Never used — Read/write

Task 3: Set up the Git Repository

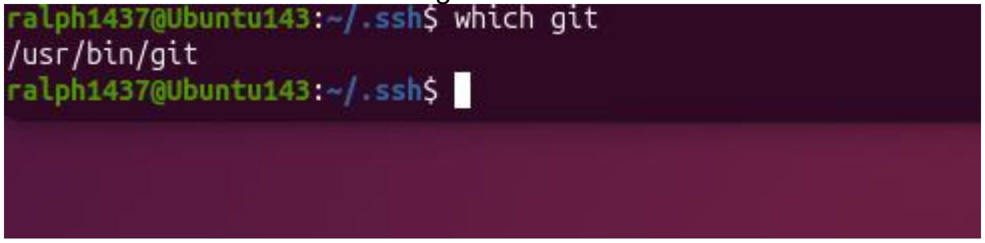
1. On the local machine, verify the version of your git using the command which git. If a directory of git is displayed, then you don’t need to install git. Otherwise, to install git, use the following command: sudo apt install git



```
ralph1437@Ubuntu143:~/.ssh$ which git
ralph1437@Ubuntu143:~/.ssh$ git
Command 'git' not found, but can be installed with:
sudo apt install git
ralph1437@Ubuntu143:~/.ssh$ which git
ralph1437@Ubuntu143:~/.ssh$
```

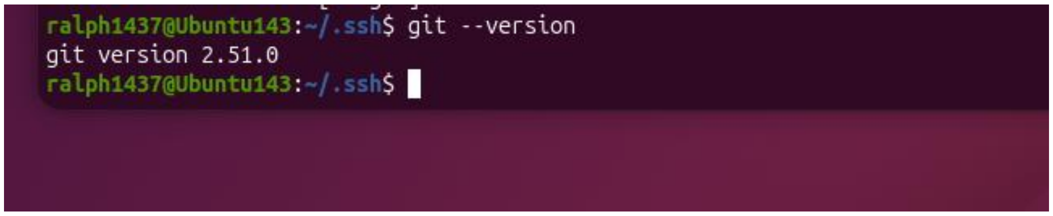
```
Continue? [Y/n] y
Get:1 http://ph.archive.ubuntu.com/ubuntu questing/main amd64 liberror-perl all 0.17030-1 [23.5 kB]
Get:2 http://ph.archive.ubuntu.com/ubuntu questing/main amd64 git-man all 1:2.51.0-1ubuntu1 [1,179 kB]
Get:3 http://ph.archive.ubuntu.com/ubuntu questing/main amd64 git amd64 1:2.51.0-1ubuntu1 [4,414 kB]
Fetched 5,617 kB in 2s (2,569 kB/s)
Selecting previously unselected package liberror-perl.
(Reading database ... 182330 files and directories currently installed.)
Preparing to unpack .../liberror-perl_0.17030-1_all.deb ...
Unpacking liberror-perl (0.17030-1) ...
Selecting previously unselected package git-man.
Preparing to unpack .../git-man_1%3a2.51.0-1ubuntu1_all.deb ...
Unpacking git-man (1:2.51.0-1ubuntu1) ...
Selecting previously unselected package git.
Preparing to unpack .../git_1%3a2.51.0-1ubuntu1_amd64.deb ...
Unpacking git (1:2.51.0-1ubuntu1) ...
Setting up liberror-perl (0.17030-1) ...
Setting up git-man (1:2.51.0-1ubuntu1) ...
Setting up git (1:2.51.0-1ubuntu1) ...
Processing triggers for man-db (2.13.1-1) ...
ralph1437@Ubuntu143:~/.ssh$
```

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



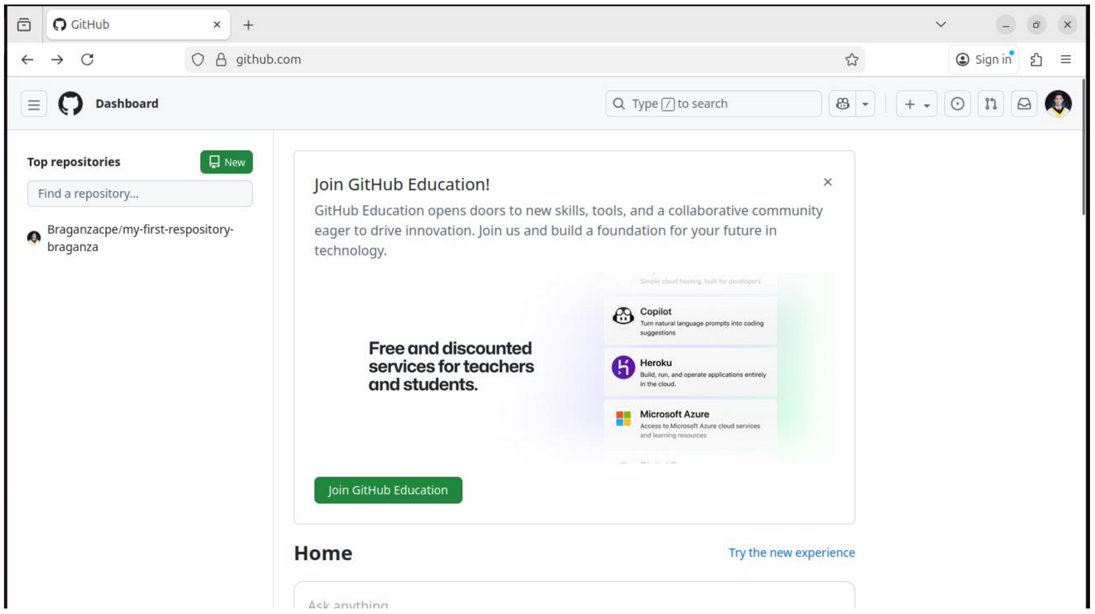
```
ralph1437@Ubuntu143:~/.ssh$ which git
/usr/bin/git
ralph1437@Ubuntu143:~/.ssh$
```

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



```
ralph1437@Ubuntu143:~/.ssh$ git --version
git version 2.51.0
ralph1437@Ubuntu143:~/.ssh$
```

4. Using the browser in the local machine, go to www.github.com.

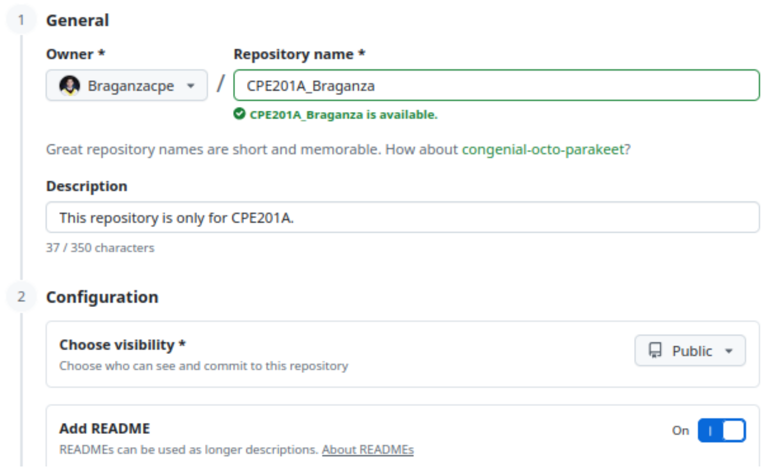


Join GitHub Education!

GitHub Education opens doors to new skills, tools, and a collaborative community eager to drive innovation. Join us and build a foundation for your future in technology.

GitHub
github.com
Dashboard
Type / to search
Sign in
Top repositories
New
Find a repository...
Braganzacpe/my-first-respository-braganza
Free and discounted services for teachers and students.
Copilot
Turn natural language prompts into coding suggestions
Heroku
Build, run, and operate applications entirely in the cloud.
Microsoft Azure
Access to Microsoft Azure cloud services and learning resources
Join GitHub Education
Home
Try the new experience
Ask anything

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.



1 General

Owner * Braganzacpe / Repository name * CPE201A_Braganza

CPE201A_Braganza is available.

Great repository names are short and memorable. How about congenial-octo-parakeet?

Description: This repository is only for CPE201A.

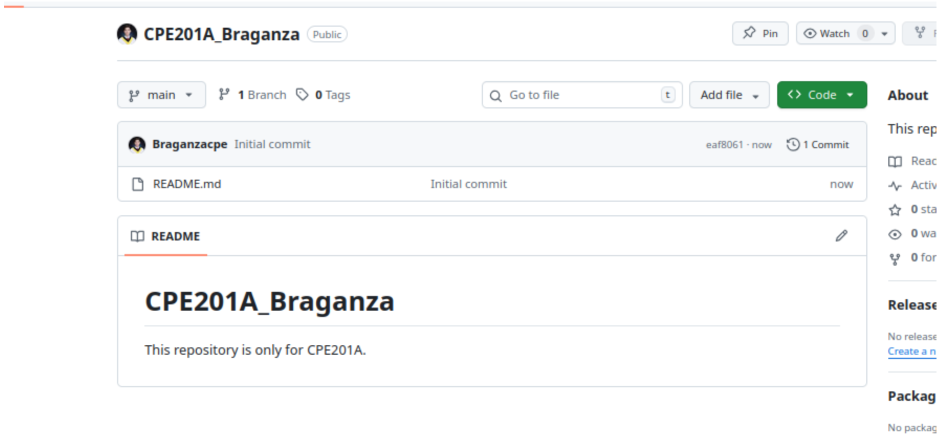
37 / 350 characters

2 Configuration

Choose visibility * — Choose who can see and commit to this repository: Public

Add README — READMEs can be used as longer descriptions. About READMEs: On

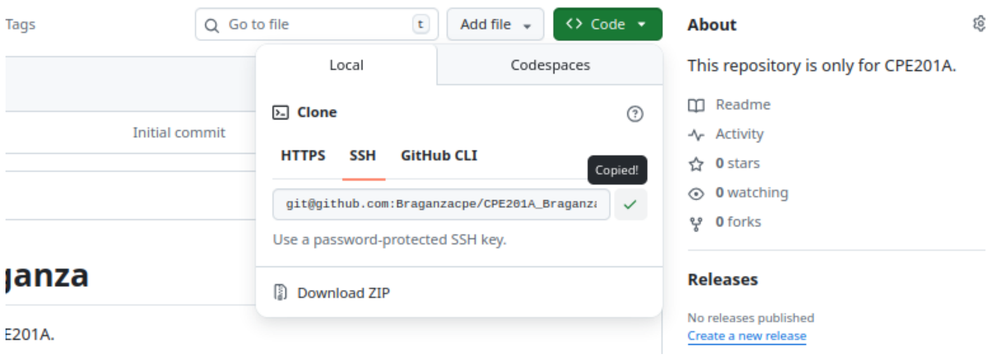
}



CPE201A_Braganza

This repository is only for CPE201A.

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.

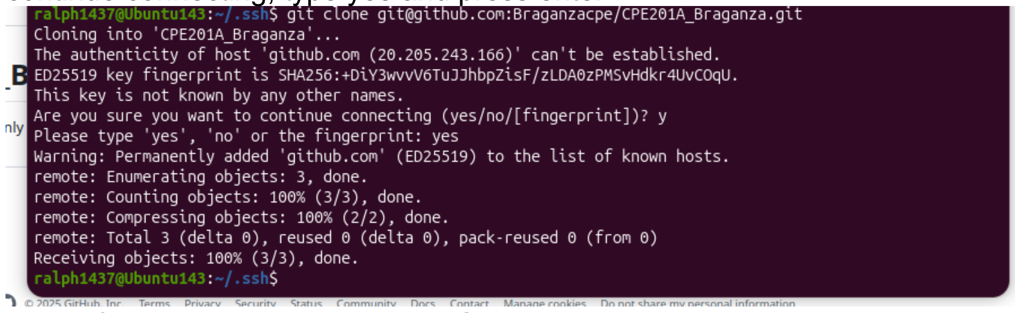


git@github.com:Braganzacpe/CPE201A_Braganz

Use a password-protected SSH key.

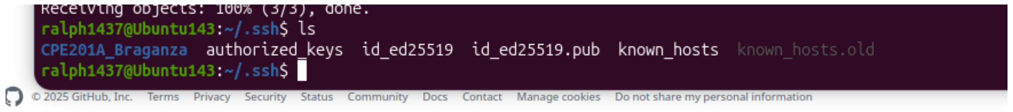
This repository is only for CPE201A.

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.



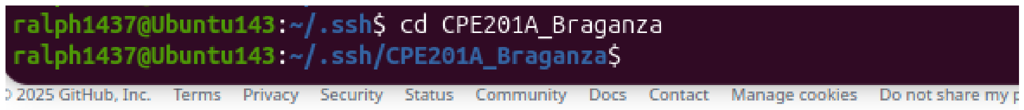
```
ralph1437@Ubuntu143:~/.ssh$ git clone git@github.com:Braganzacpe/CPE201A_Braganza.git
Cloning into 'CPE201A_Braganza'...
The authenticity of host 'github.com (20.205.243.166)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvCOqU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
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: 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.
ralph1437@Ubuntu143:~/.ssh$
```

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.



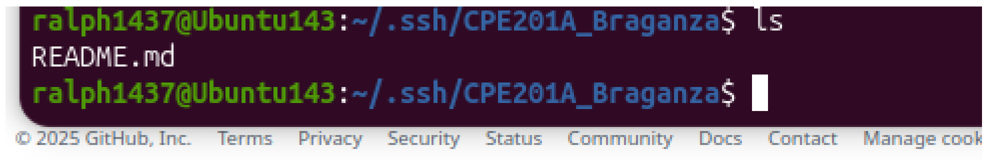
```
Receiving objects: 100% (3/3), done.
ralph1437@Ubuntu143:~/.ssh$ ls
CPE201A_Braganza  authorized_keys  id_ed25519  id_ed25519.pub  known_hosts  known_hosts.old
ralph1437@Ubuntu143:~/.ssh$
```

© 2025 GitHub, Inc. Terms Privacy Security Status Community Docs Contact Manage cookies Do not share my personal information



```
ralph1437@Ubuntu143:~/.ssh$ cd CPE201A_Braganza
ralph1437@Ubuntu143:~/.ssh/CPE201A_Braganza$
```

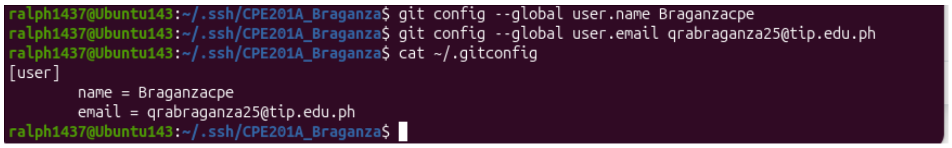
2025 GitHub, Inc. Terms Privacy Security Status Community Docs Contact Manage cookies Do not share my p



```
ralph1437@Ubuntu143:~/.ssh/CPE201A_Braganza$ ls
README.md
ralph1437@Ubuntu143:~/.ssh/CPE201A_Braganza$
```

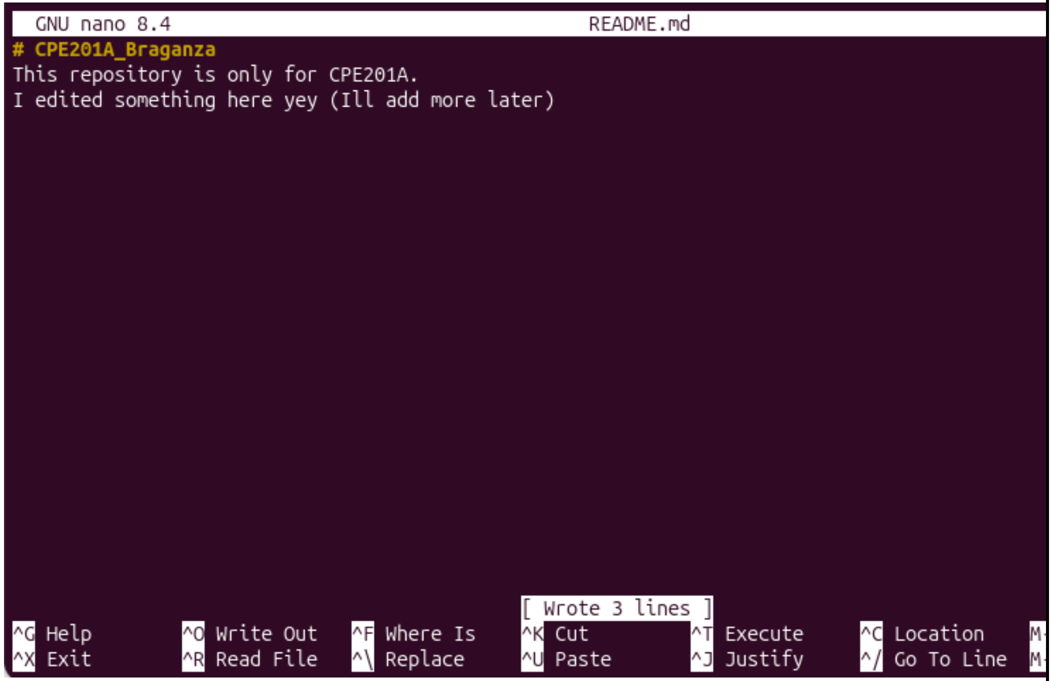
© 2025 GitHub, Inc. Terms Privacy Security Status Community Docs Contact Manage cook

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



```
ralph1437@Ubuntu143:~/.ssh/CPE201A_Braganza$ git config --global user.name Braganzacpe
ralph1437@Ubuntu143:~/.ssh/CPE201A_Braganza$ git config --global user.email qrabraganza25@tip.edu.ph
ralph1437@Ubuntu143:~/.ssh/CPE201A_Braganza$ cat ~/.gitconfig
[user]
        name = Braganzacpe
        email = qrabraganza25@tip.edu.ph
ralph1437@Ubuntu143:~/.ssh/CPE201A_Braganza$
```

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.

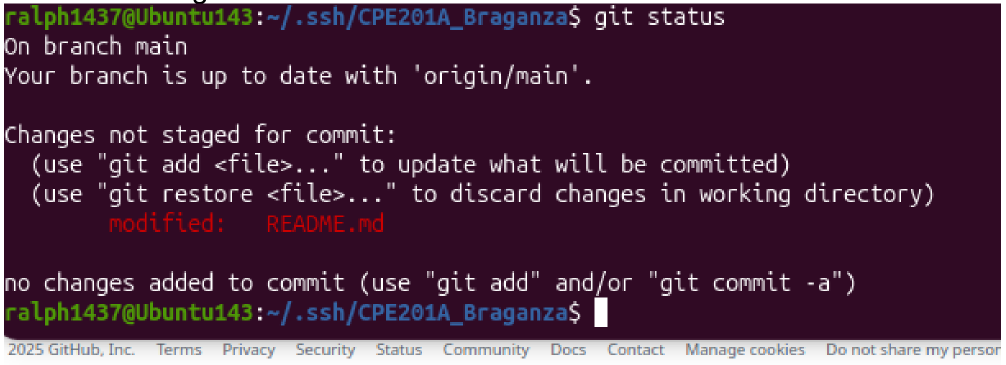


```
  GNU nano 8.4                                          README.md
# CPE201A_Braganza
This repository is only for CPE201A.
I edited something here yey (Ill add more later)




                                          [ Wrote 3 lines ]
^G Help          ^O Write Out     ^F Where Is      ^K Cut           ^T Execute       ^C Location      M-
^X Exit          ^R Read File     ^\ Replace       ^U Paste         ^J Justify       ^/ Go To Line    M-
```

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?



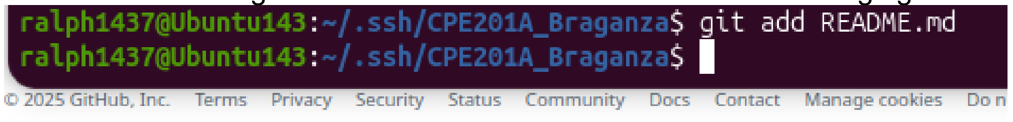
```
ralph1437@Ubuntu143:~/.ssh/CPE201A_Braganza$ git status
On branch main
Your branch is up to date with 'origin/main'.

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")
ralph1437@Ubuntu143:~/.ssh/CPE201A_Braganza$
```

2025 GitHub, Inc. Terms Privacy Security Status Community Docs Contact Manage cookies Do not share my person

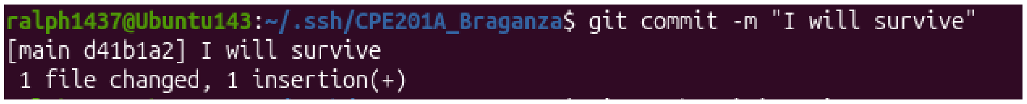
h. Use the command git add README.md to add the file into the staging area.



```
ralph1437@Ubuntu143:~/.ssh/CPE201A_Braganza$ git add README.md
ralph1437@Ubuntu143:~/.ssh/CPE201A_Braganza$
```

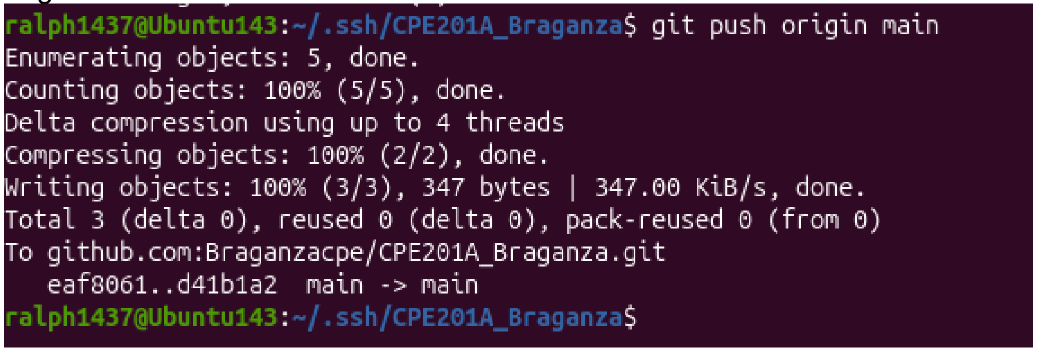
© 2025 GitHub, Inc. Terms Privacy Security Status Community Docs Contact Manage cookies Do n

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.



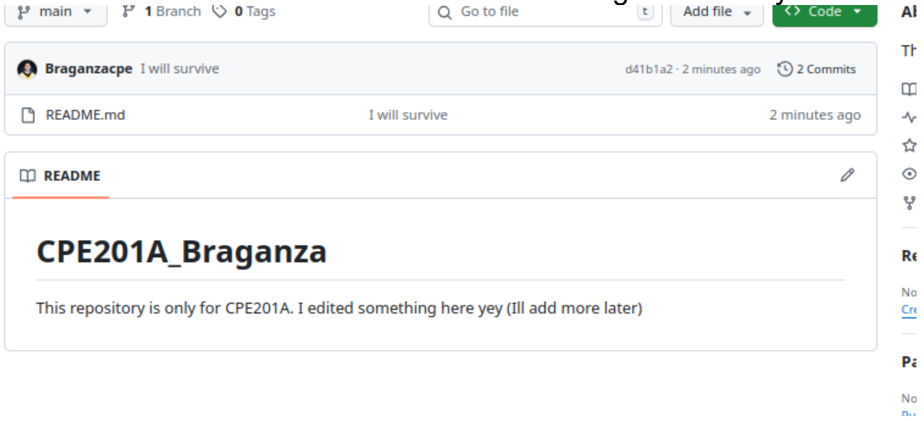
```
ralph1437@Ubuntu143:~/.ssh/CPE201A_Braganza$ git commit -m "I will survive"
[main d41b1a2] I will survive
 1 file changed, 1 insertion(+)
```

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.



```
ralph1437@Ubuntu143:~/.ssh/CPE201A_Braganza$ git push origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 347 bytes | 347.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:Braganzacpe/CPE201A_Braganza.git
   eaf8061..d41b1a2  main -> main
ralph1437@Ubuntu143:~/.ssh/CPE201A_Braganza$
```

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.



Braganzacpe I will survive d41b1a2 · 2 minutes ago 2 Commits

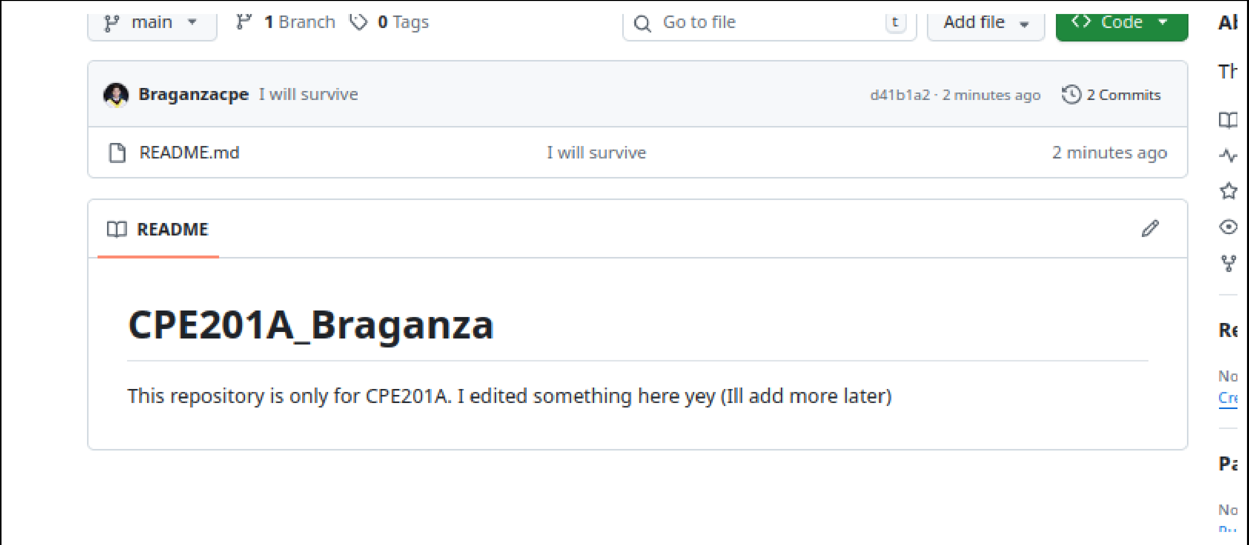
README.md I will survive 2 minutes ago

README

CPE201A_Braganza

This repository is only for CPE201A. I edited something here yey (Ill add more later)

5. Outputs:


main · 1 Branch · 0 Tags · Go to file · Add file · Code

Braganzacpe I will survive d41b1a2 · 2 minutes ago 2 Commits

README.md I will survive 2 minutes ago

README

CPE201A_Braganza

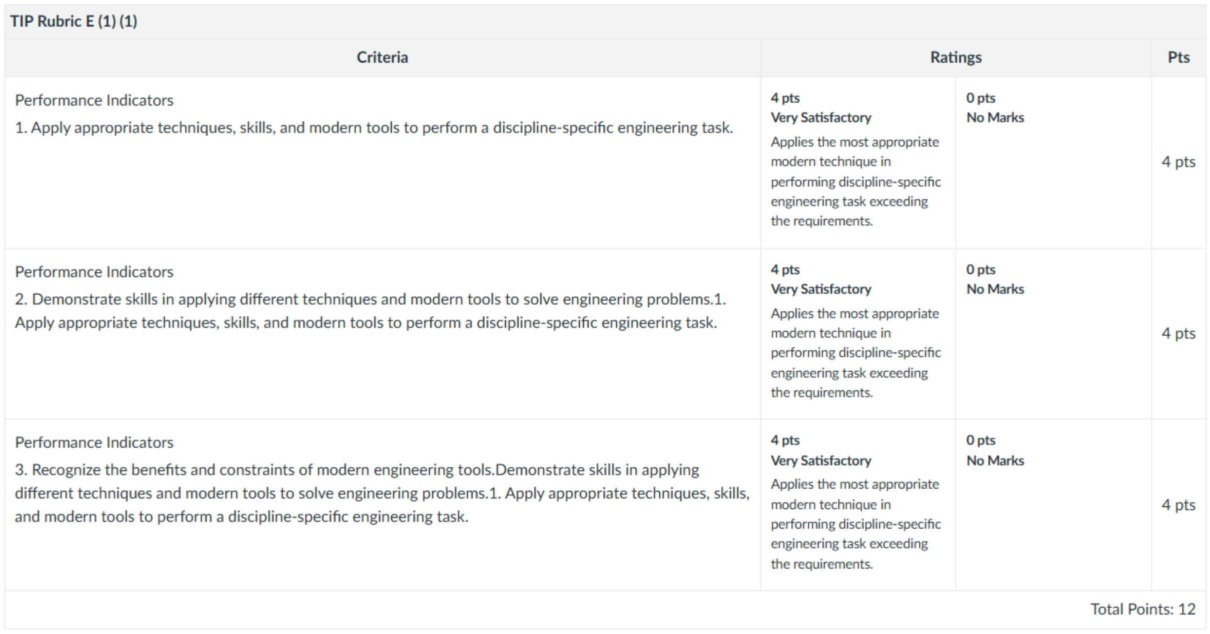
This repository is only for CPE201A. I edited something here yey (Ill add more later)

6. Conclusions/Learnings/Analysis:

This activity was quite confusing for me at first, but after analyzing it carefully, I somewhat get the gist of what I'm trying to do just by following the steps. I learned how to set up secure SSH key-based authentication and how it can replace traditional password-based logins (I did research about it to get a clearer understanding). After generating the SSH key using the ssh-keygen command using the terminal, I understood the importance of the private key staying on the local machine (or virtual machine, rather); this made me realize that I didn't need to input any sort of password for security. The key is in my local machine, which means that it doesn't send it over a network, so if someone was monitoring network traffic (specifically to get my password or personal information), there is no way they can capture a password to gain access. The only way for people to access anything from the local machine is to get ahold of the private key before they can do any malicious activity. I've also gained practical experience in configuring Git and connecting it with my GitHub using SSH. I learned how to create and clone a repository and edit files (this is where I edited the README.md file) using the basic Git commands in the terminal. The git add, git commit, and git push helped me understand how version control works and how changes in code or documentation are tracked over time. Every time I make a change and run git commit, Git takes a snapshot of the files at that moment, and each commit I do has a unique ID and message describing what I change or what other people change. I did some more research, and I can view these changes using the command git log, which shows me the development timeline, and Git also allows me to create branches, which separates lines of development. This makes it easy for me to test new features or write new documentation without affecting the main project, and when I'm done with the tested branch, I can later merge this branch back into the main version when it is ready.

Overall, this activity helped me see the connections between system security (through SSH) and how I can collaborate with other people through Git and GitHub. If I had more time to invest myself in this, I would take that chance to expand my knowledge and understand this, because I could use this as my portfolio when finding a potential job that could pay me well. This activity has given me a foundation of what it would look like if I was ever going to develop some sort of project in the future. I hope that someday I can develop something for myself and showcase the skills I've accumulated over the years, like a personal life journal.

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