| Activity No. 14 | |
|---|---|
| **SSH Key-Based Authentication and GIT Setup** | |
| **Name: Niel Vincent B. Condino** | **Date Performed: 11/11/25** |
| **Course Code: CPE 201A** | **Date Submitted: 11/12/25** |
| **Course Title: Computer System Administration and Troubleshooting** | **Instructor: Jimlord 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.

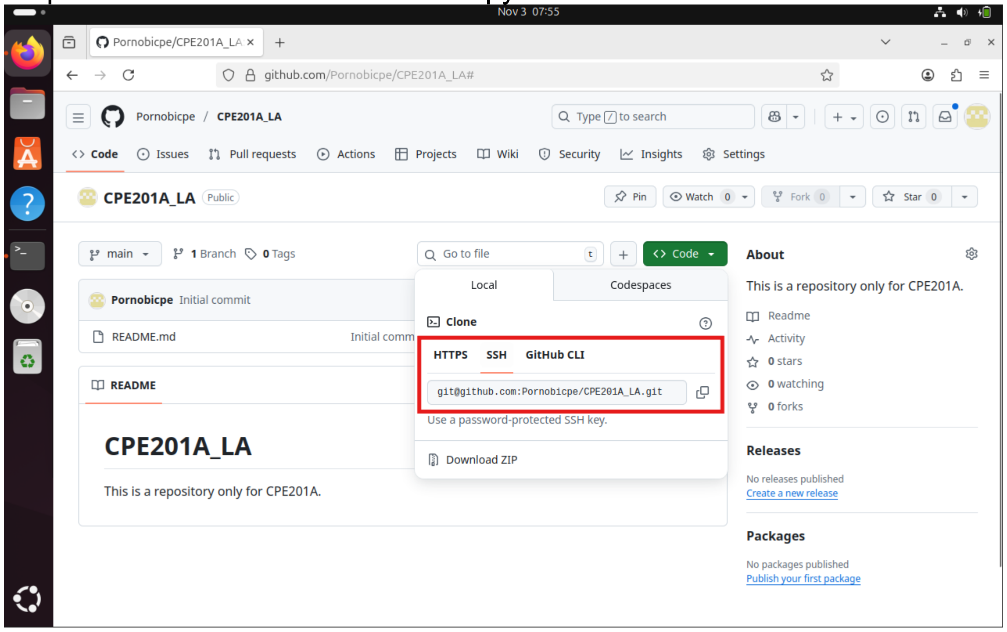
**Task 2: Copying the Public Key to Remote Servers**

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

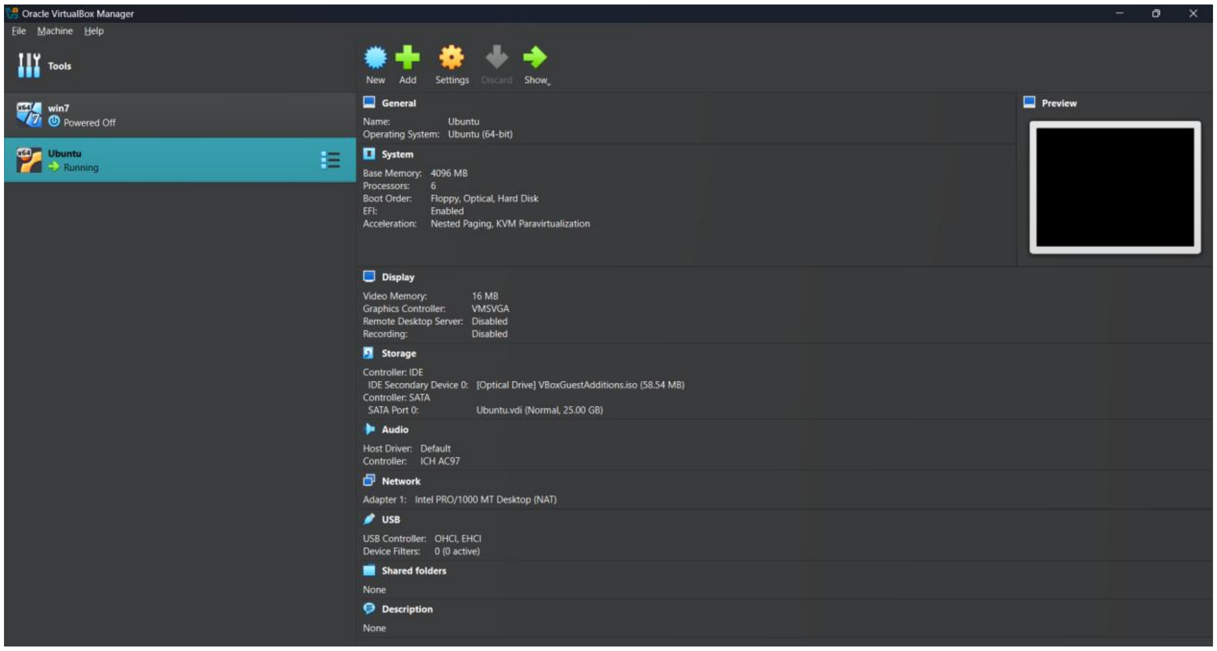
~/.gitconfig

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:**

Task 1.

1.



General

```
Name:               Ubuntu
Operating System:   Ubuntu (64-bit)
```

System

```
Base Memory:   4096 MB
Processors:    6
Boot Order:    Floppy, Optical, Hard Disk
EFI:           Enabled
Acceleration:  Nested Paging, KVM Paravirtualization
```

Display

```
Video Memory:             16 MB
Graphics Controller:      VMSVGA
Remote Desktop Server:    Disabled
Recording:                Disabled
```

Storage

```
Controller: IDE
 IDE Secondary Device 0:  [Optical Drive] VBoxGuestAdditions.iso (58.54 MB)
Controller: SATA
 SATA Port 0:             Ubuntu.vdi (Normal, 25.00 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)
```

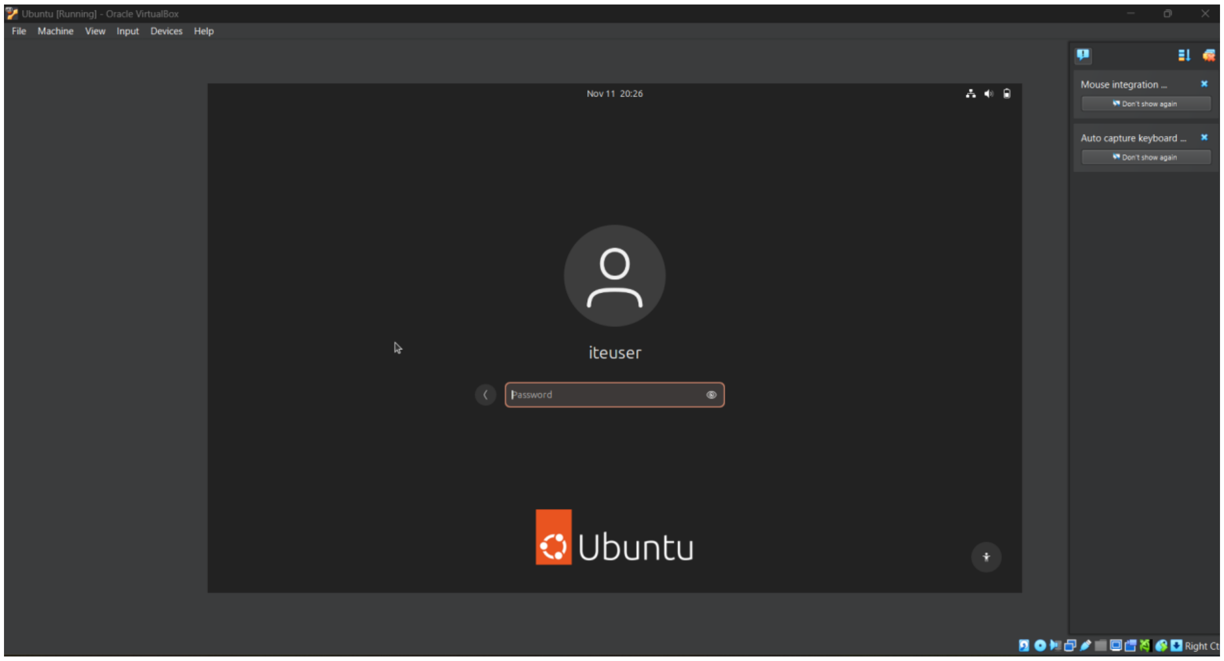
Shared folders

None

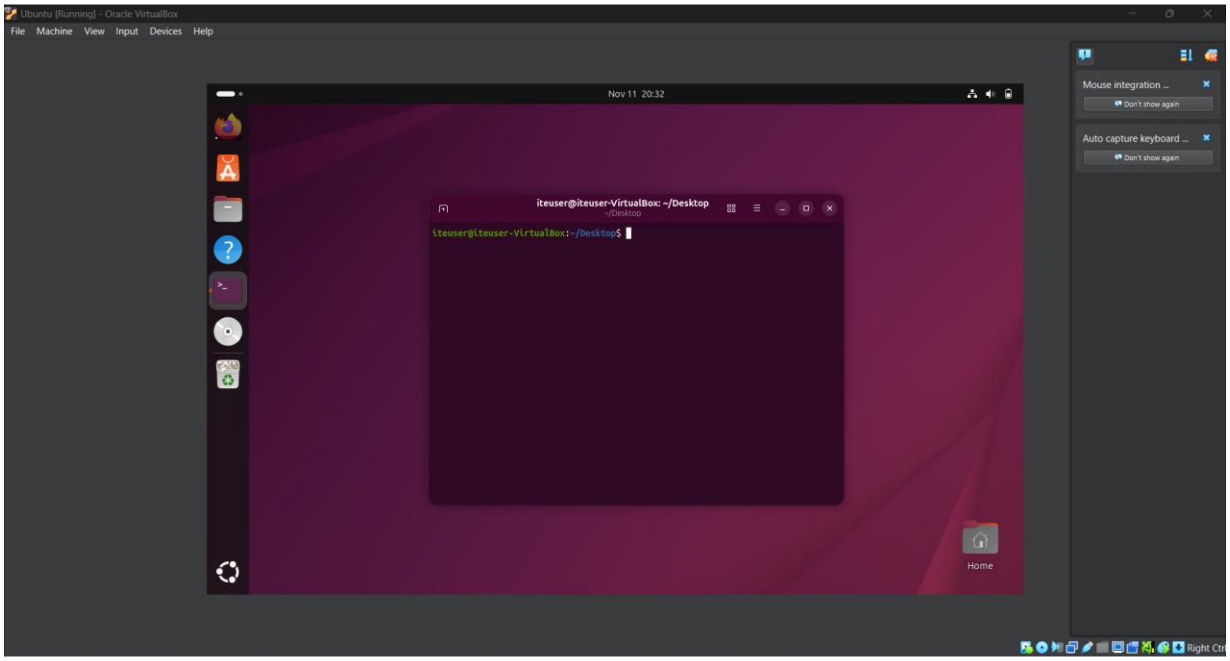
Description

None

2.

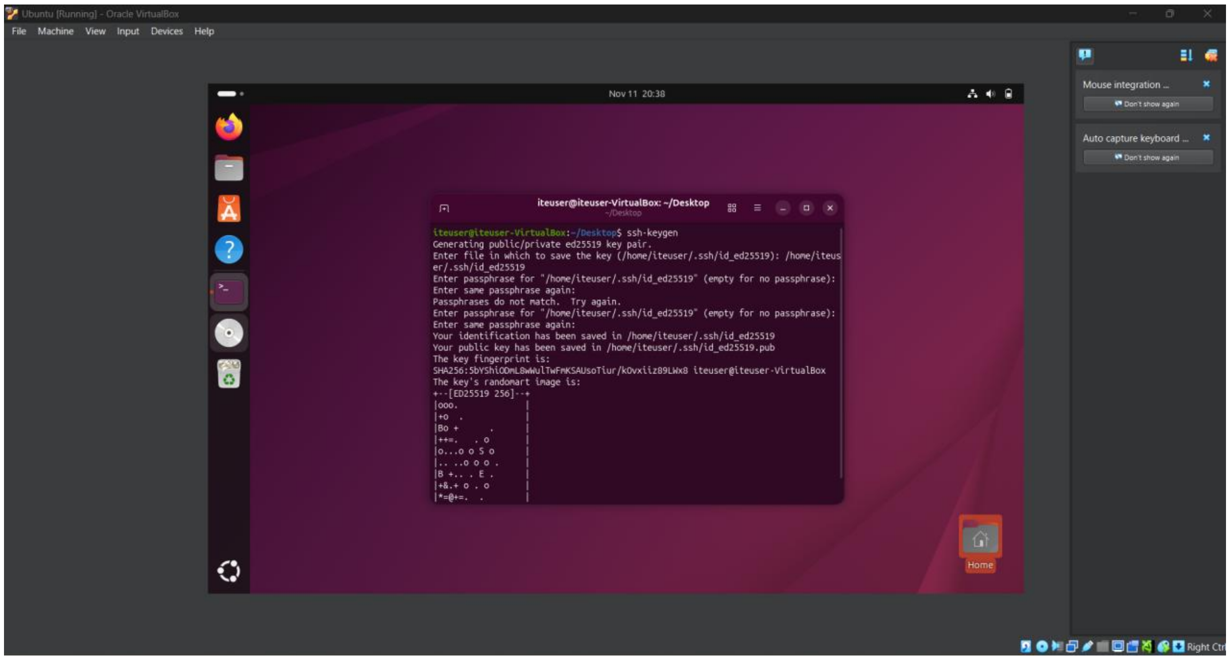

Ubuntu [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Nov 11 20:26
Mouse integration ...
Don't show again
Auto capture keyboard ...
Don't show again
iteuser
Password
Ubuntu
Right Ctrl

3.



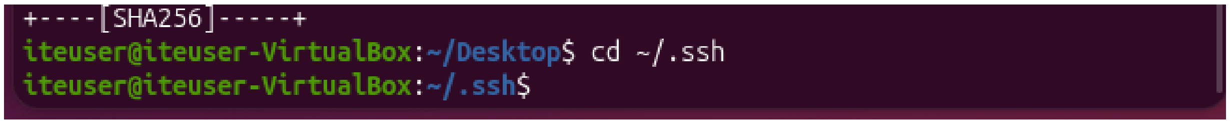
```
iteuser@iteuser-VirtualBox:~/Desktop$
```

4.



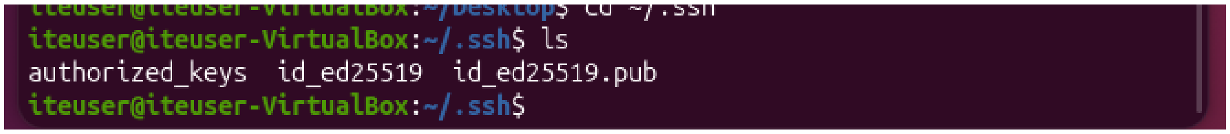
```
iteuser@iteuser-VirtualBox:~/Desktop$ ssh-keygen
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/iteuser/.ssh/id_ed25519): /home/iteus
er/.ssh/id_ed25519
Enter passphrase for "/home/iteuser/.ssh/id_ed25519" (empty for no passphrase):
Enter same passphrase again:
Passphrases do not match.  Try again.
Enter passphrase for "/home/iteuser/.ssh/id_ed25519" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/iteuser/.ssh/id_ed25519
Your public key has been saved in /home/iteuser/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:5bYShiODmL8wWulTwFmKSAUsoTiur/kOvxiiz89LWx8 iteuser@iteuser-VirtualBox
The key's randomart image is:
+--[ED25519 256]--+
|ooo.             |
|+o  .            |
|Bo +      .      |
|++=.    . o      |
|o...o o S o      |
|.. ..o o o .     |
|B +.. . E .      |
|+&.+ o . o       |
|*=@+=.  .        |
```

5.



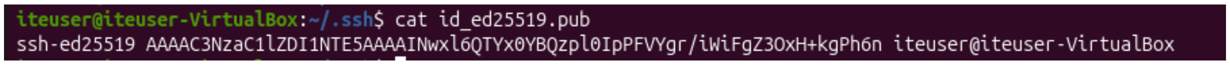
```
+----[SHA256]-----+
iteuser@iteuser-VirtualBox:~/Desktop$ cd ~/.ssh
iteuser@iteuser-VirtualBox:~/.ssh$
```

6.



```
iteuser@iteuser-VirtualBox:~/Desktop$ cd ~/.ssh
iteuser@iteuser-VirtualBox:~/.ssh$ ls
authorized_keys  id_ed25519  id_ed25519.pub
iteuser@iteuser-VirtualBox:~/.ssh$
```

**7.**



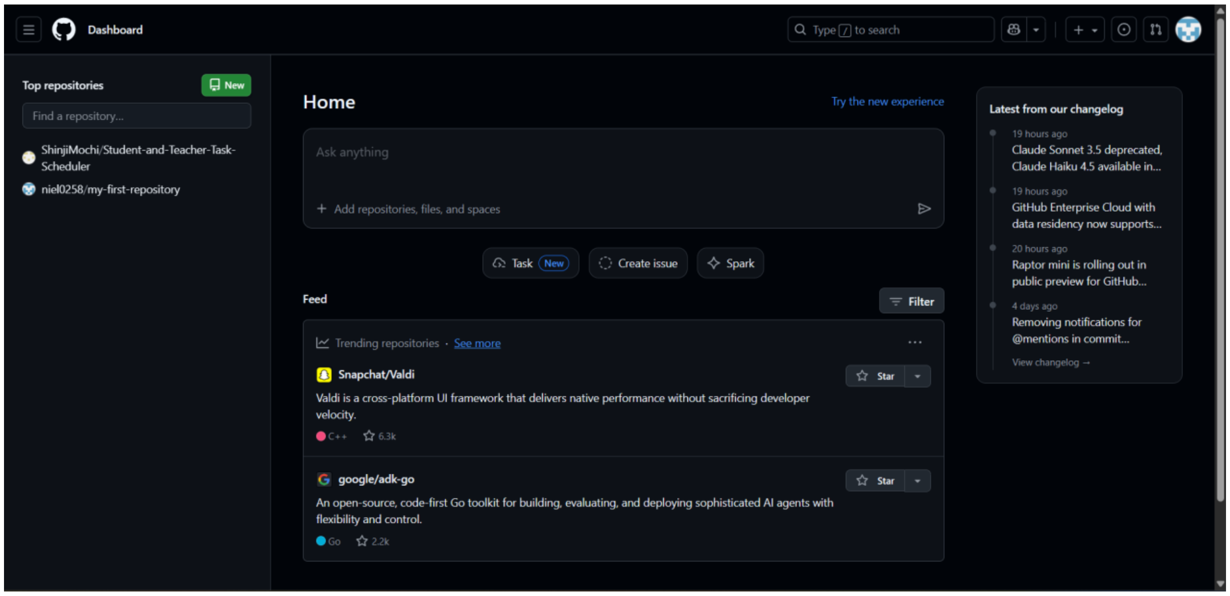
```
iteuser@iteuser-VirtualBox:~/.ssh$ cat id_ed25519.pub
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAINwxl6QTYx0YBQzpl0IpPFVYgr/iWiFgZ3OxH+kgPh6n iteuser@iteuser-VirtualBox
```

**8.**

```
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAINwxl6QTYx0YBQzpl0IpPFVYgr/iWiFgZ3OxH+kgPh6n iteuser@iteuser-VirtualBox
```

## Task 2:

**1.**



Dashboard

Top repositories — New

Find a repository...

- ShinjiMochi/Student-and-Teacher-Task-Scheduler
- niel0258/my-first-repository

Home — Try the new experience

Ask anything

+ Add repositories, files, and spaces

Task New · Create issue · Spark

Feed — Filter

Trending repositories · See more

Snapchat/Valdi — Star

Valdi is a cross-platform UI framework that delivers native performance without sacrificing developer velocity.

C++ · 6.3k

google/adk-go — Star

An open-source, code-first Go toolkit for building, evaluating, and deploying sophisticated AI agents with flexibility and control.

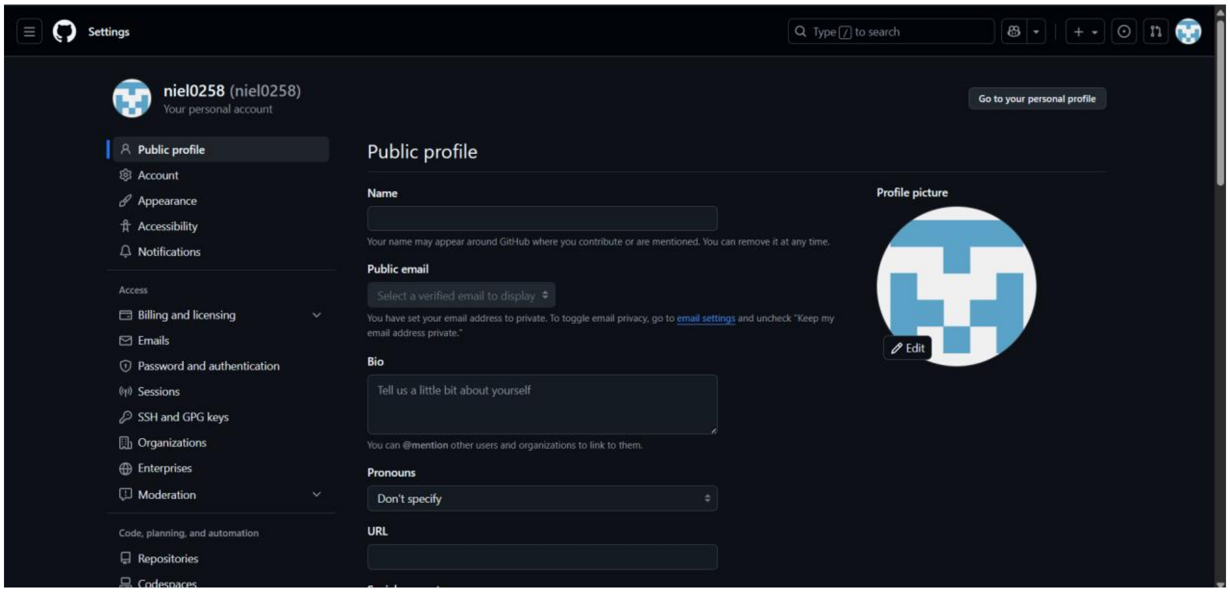
Go · 2.2k

Latest from our changelog

- 19 hours ago — Claude Sonnet 3.5 deprecated, Claude Haiku 4.5 available in...
- 19 hours ago — GitHub Enterprise Cloud with data residency now supports...
- 20 hours ago — Raptor mini is rolling out in public preview for GitHub...
- 4 days ago — Removing notifications for @mentions in commit...

View changelog →

**2.**



Settings

niel0258 (niel0258)
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

Code, planning, and automation: Repositories · Codespaces

Public profile

Name

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 have set your email address to private. To toggle email privacy, go to email settings and uncheck "Keep my email address private."

Bio: Tell us a little bit about yourself

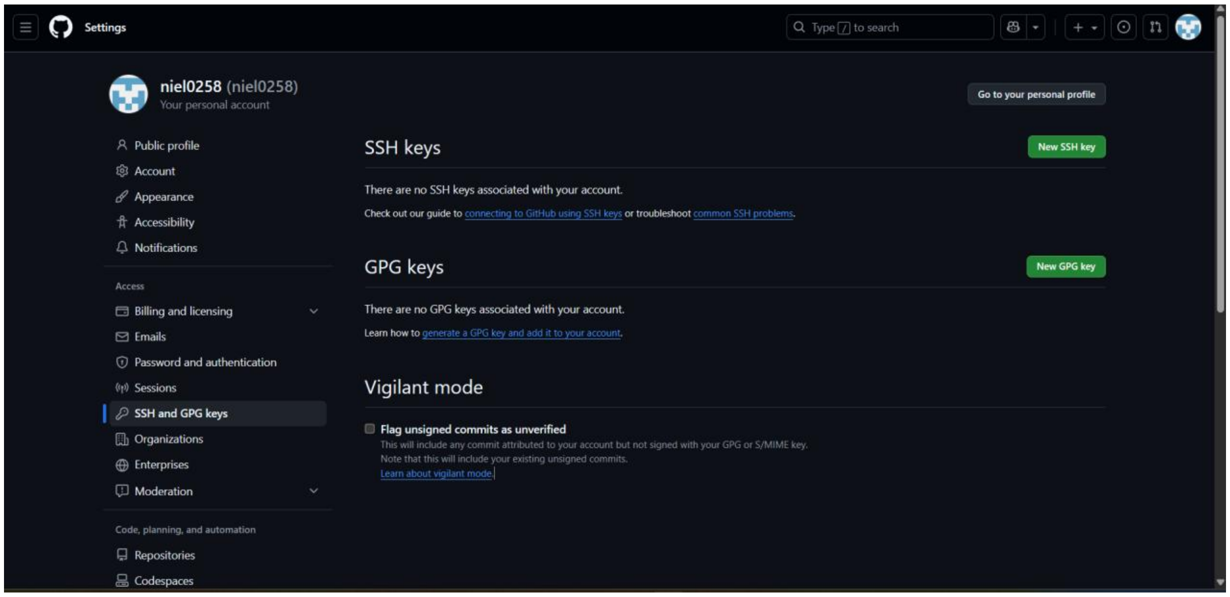
You can @mention other users and organizations to link to them.

Pronouns: Don't specify

URL

Profile picture — Edit

3.



SSH keys

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

There are no GPG keys associated with your account.

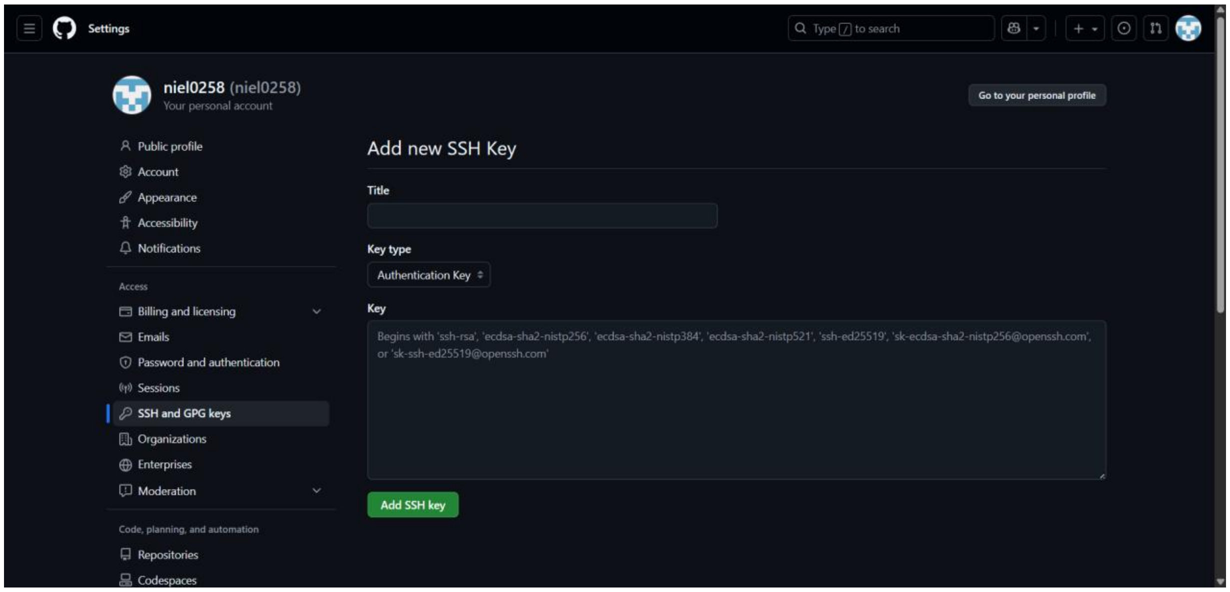
Learn how to generate a GPG key and add it to your account.

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.

5.



Add new SSH Key

Title

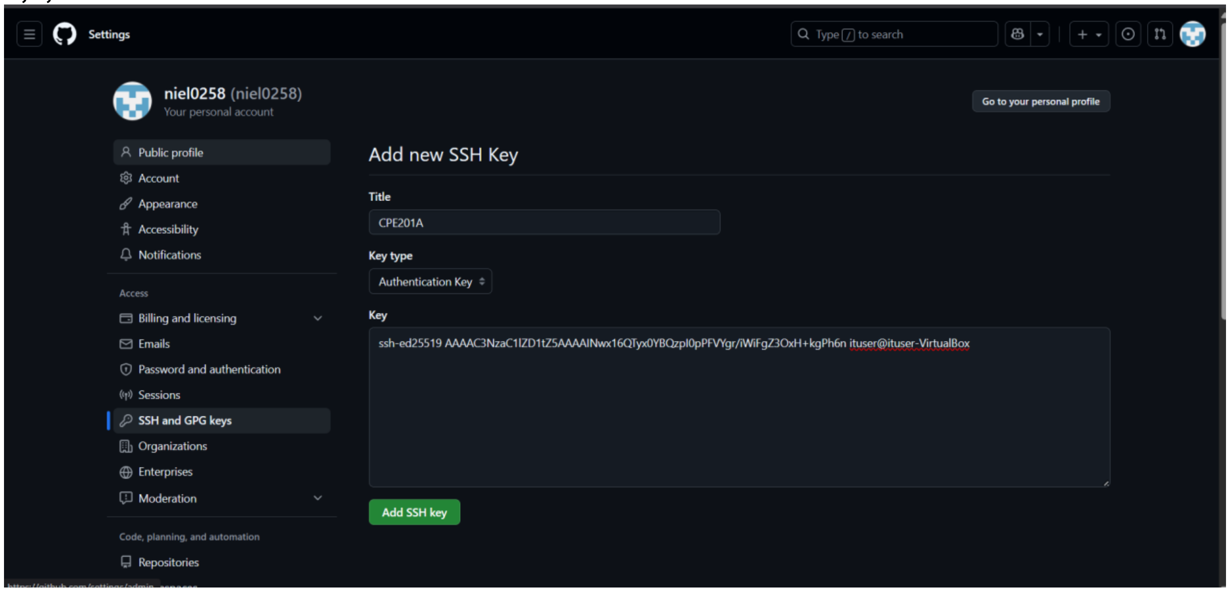
Key type

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'

6,7,8.



Add new SSH Key

Title

CPE201A

Key type

Authentication Key

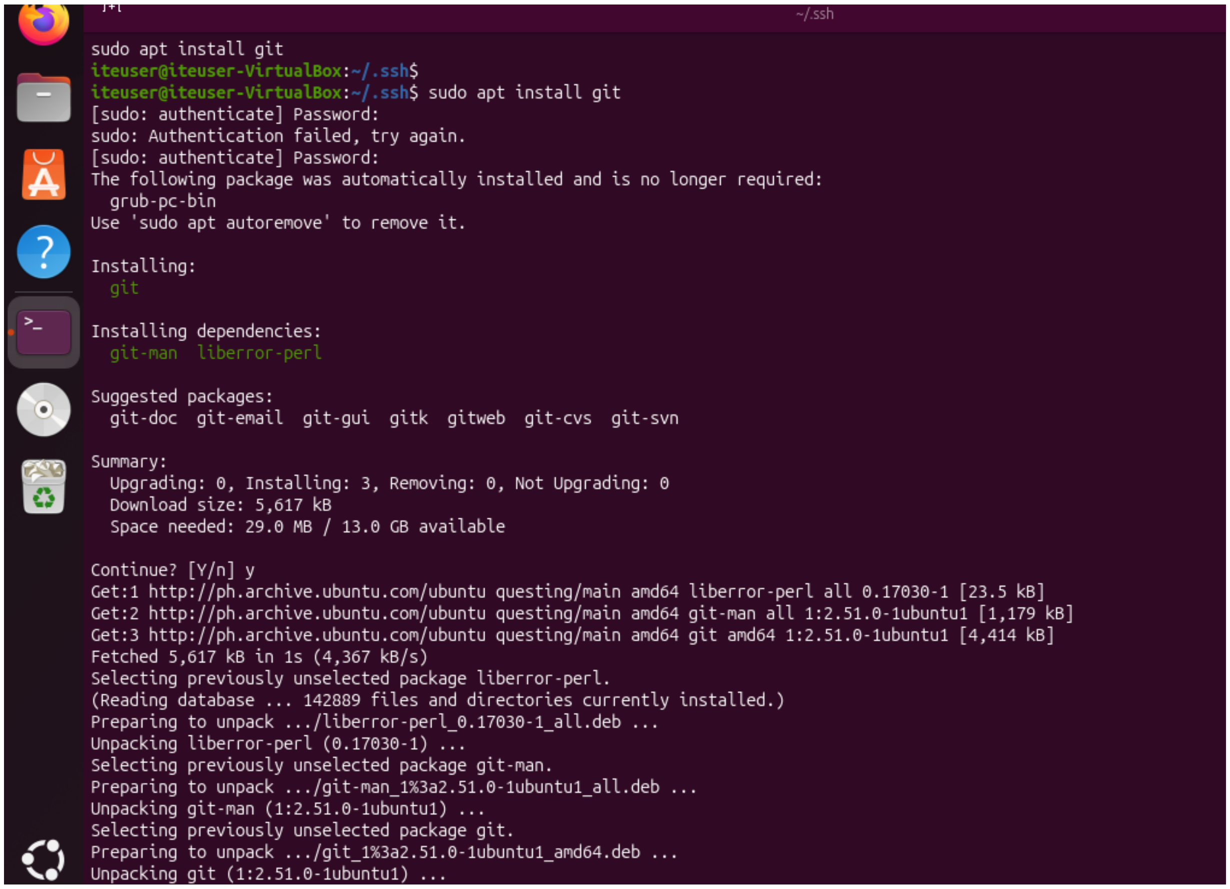
Key

```
ssh-ed25519 AAAAC3NzaC1lZD1tZ5AAAAINwx16QTyx0YBQzpl0pPFVYgr/iWiFgZ3OxH+kgPh6n ituser@ituser-VirtualBox
```

Add SSH key

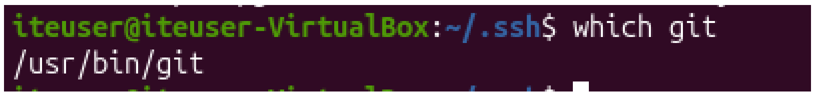
Task 3

1.



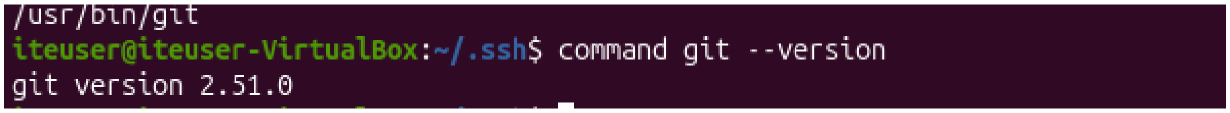
```
sudo apt install git
iteuser@iteuser-VirtualBox:~/.ssh$
iteuser@iteuser-VirtualBox:~/.ssh$ sudo apt install git
[sudo: authenticate] Password:
sudo: Authentication failed, try again.
[sudo: authenticate] Password:
The following package was automatically installed and is no longer required:
  grub-pc-bin
Use 'sudo apt autoremove' to remove it.

Installing:
  git

Installing dependencies:
  git-man  liberror-perl

Suggested packages:
  git-doc  git-email  git-gui  gitk  gitweb  git-cvs  git-svn

Summary:
  Upgrading: 0, Installing: 3, Removing: 0, Not Upgrading: 0
  Download size: 5,617 kB
  Space needed: 29.0 MB / 13.0 GB available

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 1s (4,367 kB/s)
Selecting previously unselected package liberror-perl.
(Reading database ... 142889 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) ...
```

2.



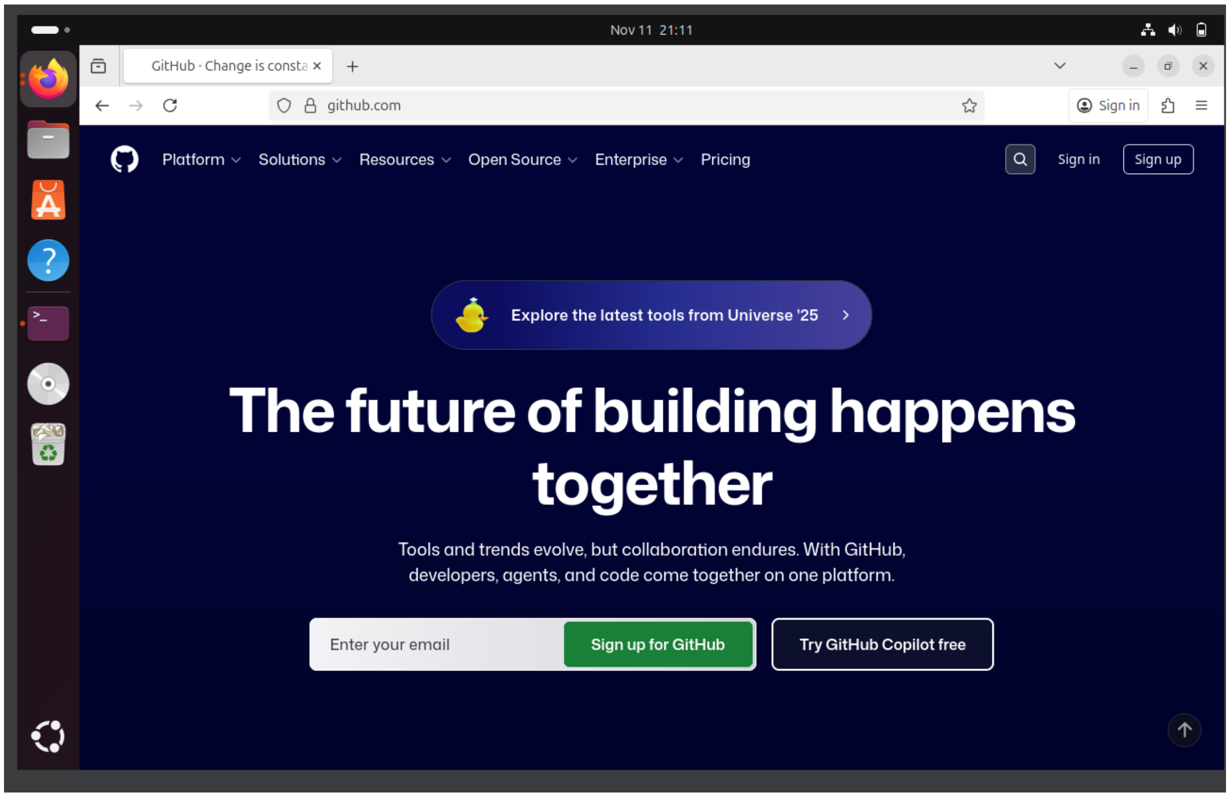
```
iteuser@iteuser-VirtualBox:~/.ssh$ which git
/usr/bin/git
```

**3.**



```
/usr/bin/git
iteuser@iteuser-VirtualBox:~/.ssh$ command git --version
git version 2.51.0
```

**4.**



Nov 11 21:11

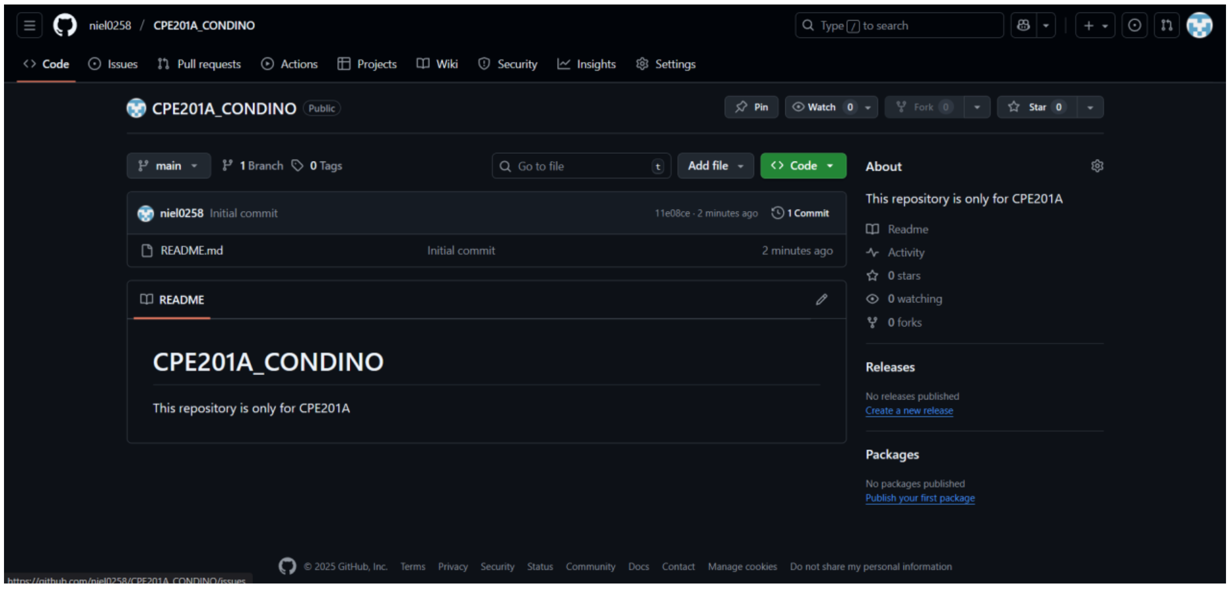
GitHub · Change is consta

github.com

The future of building happens together

Tools and trends evolve, but collaboration endures. With GitHub, developers, agents, and code come together on one platform.

**5.**



niel0258 / CPE201A_CONDINO

CPE201A_CONDINO Public

niel0258 Initial commit 11e08ce · 2 minutes ago 1 Commit

README.md Initial commit 2 minutes ago

README

CPE201A_CONDINO

This repository is only for CPE201A

About

This repository is only for CPE201A

Readme
Activity
0 stars
0 watching
0 forks

Releases

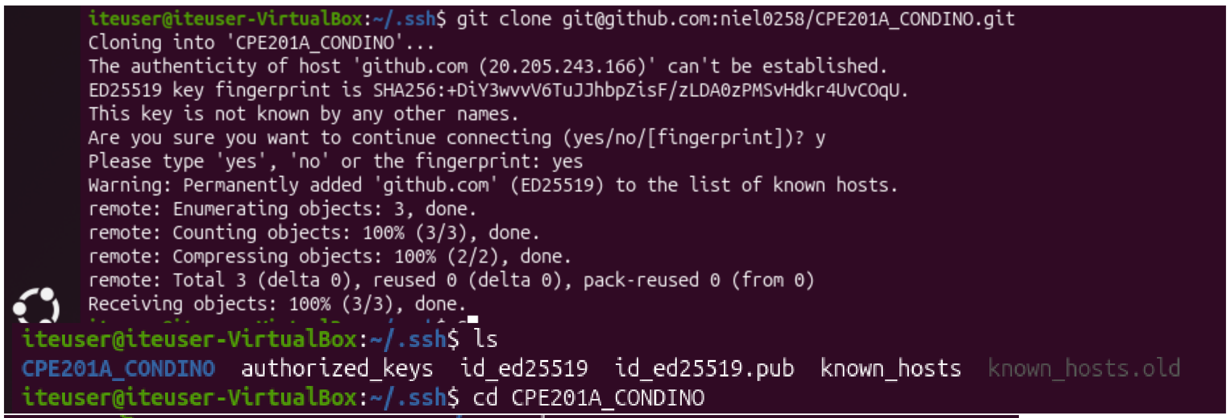
No releases published
Create a new release

Packages

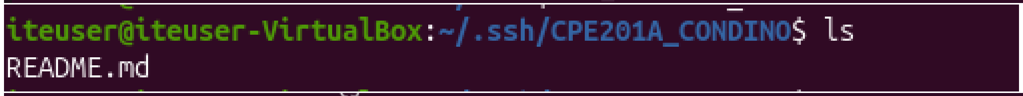
No packages published
Publish your first package

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

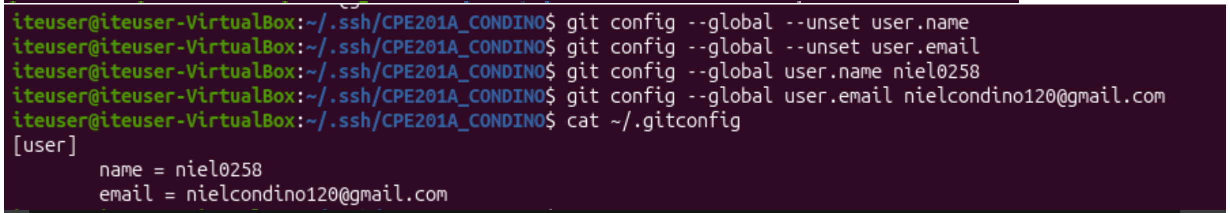
https://github.com/niel0258/CPE201A_CONDINO/issues



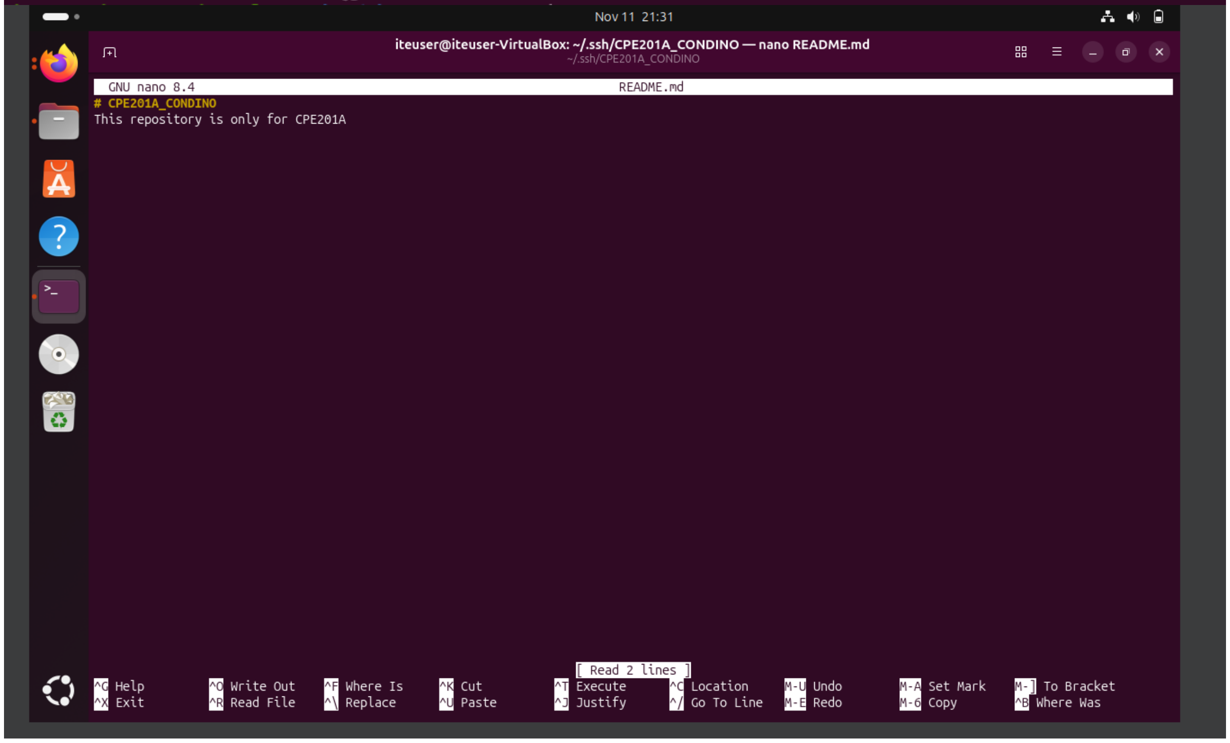
```
iteuser@iteuser-VirtualBox:~/.ssh$ git clone git@github.com:niel0258/CPE201A_CONDINO.git
Cloning into 'CPE201A_CONDINO'...
The authenticity of host 'github.com (20.205.243.166)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvCOqU.
This key is not known 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.
iteuser@iteuser-VirtualBox:~/.ssh$ ls
CPE201A_CONDINO  authorized_keys  id_ed25519  id_ed25519.pub  known_hosts  known_hosts.old
iteuser@iteuser-VirtualBox:~/.ssh$ cd CPE201A_CONDINO
```



```
iteuser@iteuser-VirtualBox:~/.ssh/CPE201A_CONDINO$ ls
README.md
```

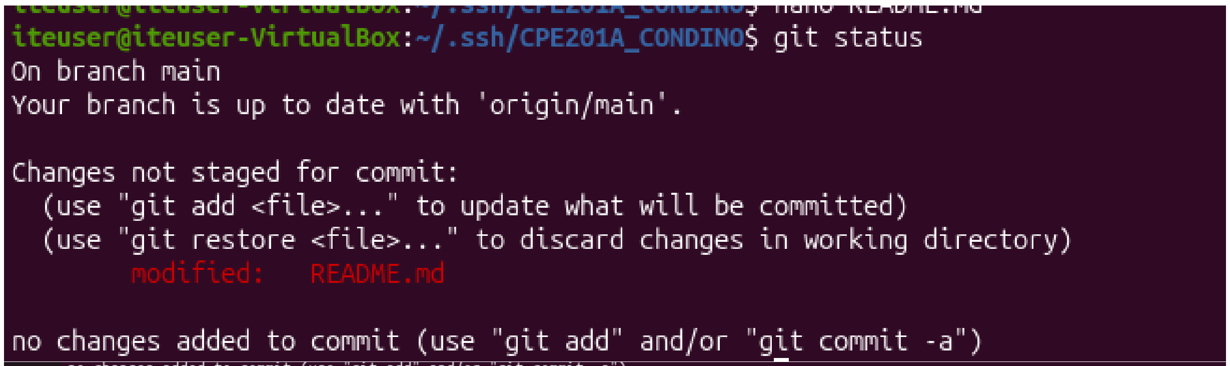


```
iteuser@iteuser-VirtualBox:~/.ssh/CPE201A_CONDINO$ git config --global --unset user.name
iteuser@iteuser-VirtualBox:~/.ssh/CPE201A_CONDINO$ git config --global --unset user.email
iteuser@iteuser-VirtualBox:~/.ssh/CPE201A_CONDINO$ git config --global user.name niel0258
iteuser@iteuser-VirtualBox:~/.ssh/CPE201A_CONDINO$ git config --global user.email nielcondino120@gmail.com
iteuser@iteuser-VirtualBox:~/.ssh/CPE201A_CONDINO$ cat ~/.gitconfig
[user]
        name = niel0258
        email = nielcondino120@gmail.com
```

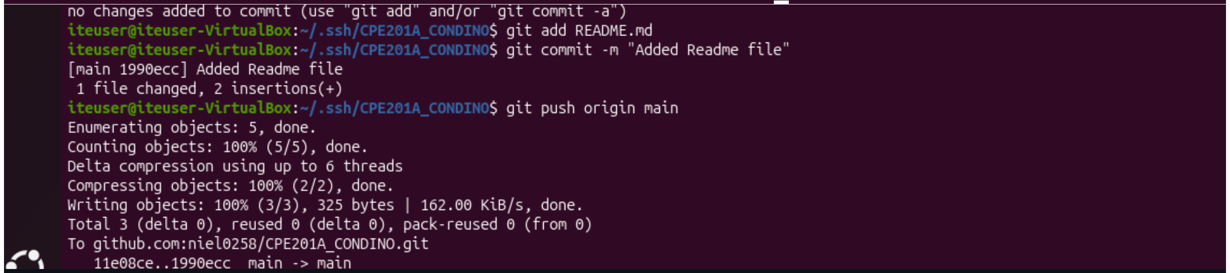


Nov 11  21:31
iteuser@iteuser-VirtualBox: ~/.ssh/CPE201A_CONDINO — nano README.md
~/.ssh/CPE201A_CONDINO

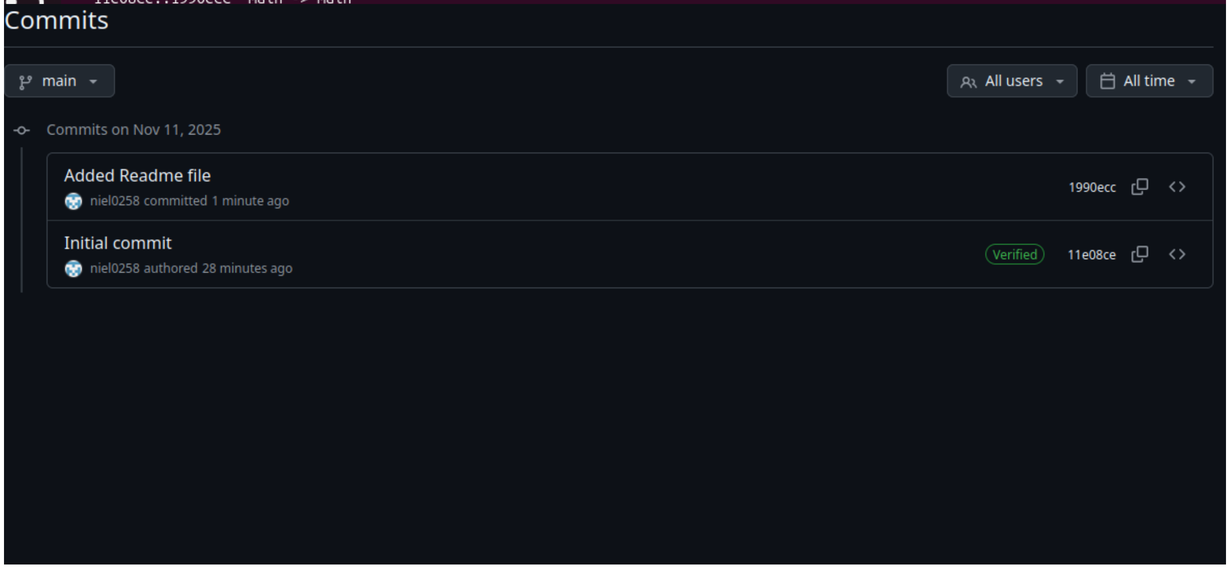
```
  GNU nano 8.4                          README.md
# CPE201A_CONDINO
This repository is only for CPE201A




                                  [ Read 2 lines ]
^G Help      ^O Write Out  ^F Where Is   ^K Cut       ^T Execute   ^C Location   M-U Undo    M-A Set Mark   M-] To Bracket
^X Exit      ^R Read File  ^\ Replace    ^U Paste     ^J Justify   ^/ Go To Line M-E Redo    M-6 Copy       ^B Where Was
```



```
iteuser@iteuser-VirtualBox:~/.ssh/CPE201A_CONDINO$ nano README.md
iteuser@iteuser-VirtualBox:~/.ssh/CPE201A_CONDINO$ 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")
```



```
no changes added to commit (use "git add" and/or "git commit -a")
iteuser@iteuser-VirtualBox:~/.ssh/CPE201A_CONDINO$ git add README.md
iteuser@iteuser-VirtualBox:~/.ssh/CPE201A_CONDINO$ git commit -m "Added Readme file"
[main 1990ecc] Added Readme file
 1 file changed, 2 insertions(+)
iteuser@iteuser-VirtualBox:~/.ssh/CPE201A_CONDINO$ git push origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 6 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 325 bytes | 162.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:niel0258/CPE201A_CONDINO.git
   11e08ce..1990ecc  main -> main
```



Commits on Nov 11, 2025

Added Readme file
niel0258 committed 1 minute ago — 1990ecc

Initial commit
niel0258 authored 28 minutes ago — Verified — 11e08ce

**6. Conclusions/Learnings/Analysis:**

In this activity, I learned more about setting up a github repository. By following all the steps provided in the activity, I was able to complete the task and, while doing so, learned how to clone a repository using a SSH key. I also became more familiar with how the github cli process works. Based on what I have seen, the git commands are similar in both Windows and Ubuntu. From what I learned, the process generally follows this order: clone – modify – add – commit. Overall, I can say that I was able to perform the task successfully and, in the process, learned how to use git commands in Linux.

| 7. Assessment Rubric: |
|---|
| |