

Activity No. 14	
SSH Key-Based Authentication and GIT Setup	
Name: Lopez, Andrei Dion C.	Date Performed: November 11, 2025
Course Code: CPE 201A	Date Submitted: November 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:	
<p>By the end of this activity, the students should be able to:</p> <ul style="list-style-type: none"> • 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:	
<p>Part 1: Discussion</p> <p>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).</p> <p>Provide screenshots for each task.</p> <p>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.</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 passwords stored in files and eliminated the possibility of a compromised server stealing the user's password.</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.

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

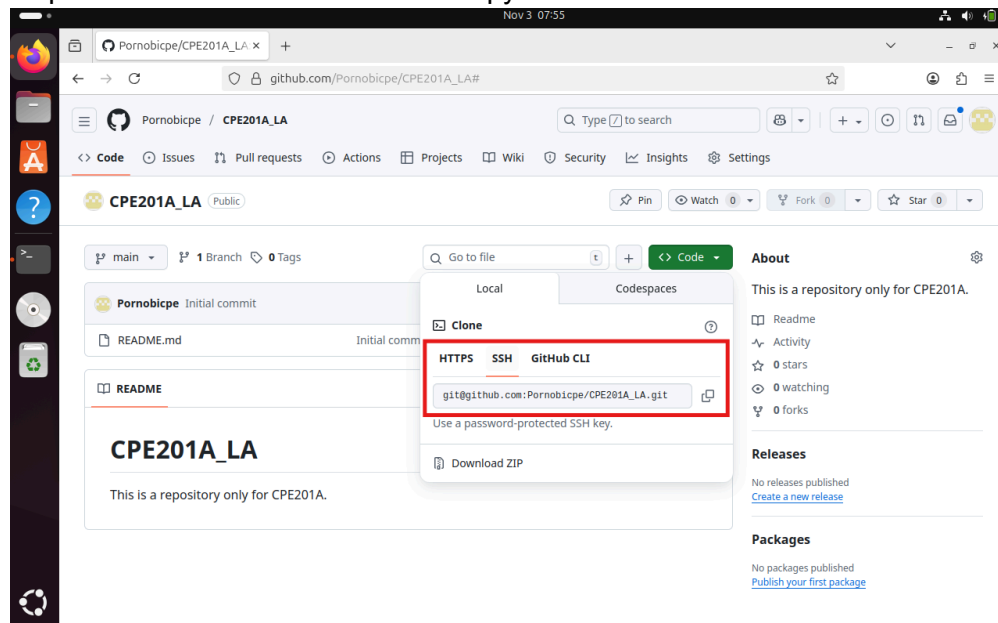
1. Open VirtualBox and start your Ubuntu virtual machine.
2. Log in using your username and password.
3. Open the Terminal.
4. Generate an SSH key pair by typing the following command and pressing Enter:
`ssh-keygen`
5. Navigate to the SSH directory:
`cd ~/.ssh`
6. List the files in the directory:
`ls`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



- 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 command 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:

```
iteuser@iteuser-VirtualBox:~$ ssh-keygen
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/iteuser/.ssh/id_ed25519): /home/iteuser/.ssh/id_ed25519
Enter passphrase (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:yf2A9k6IAASA76zHvXWEiGMZVaqaV0HrhLt2Fv8D+9s iteuser@iteuser-VirtualBox
The key's randomart image is:
+--[ED25519 256]--+
|=. . . . .|
|.. . . .|
|.o . . .|
| .* . o +|
| o* o . S o|
|o+oo o + o o|
|+* .o = o o .|
|+o=o.+ o.o|
|=B+ ..o+oE.|
+----[SHA256]-----+

iteuser@iteuser-VirtualBox:~$ cd ~/.ssh
iteuser@iteuser-VirtualBox:~/.ssh$ ls
authorized_keys  id_ed25519  id_ed25519.pub
iteuser@iteuser-VirtualBox:~/.ssh$ cat id_ed25519.pub
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIPlfFp509qS0/fWxQ2pET+YyN4escFabbNA9+5mgIA/r iteuser@iteuser-VirtualBox
```

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

**CPE201A**

SHA256:AzK+MoUT1ixbC1N90//peZ7iN7TLmeCmFV+NcImH1Jg

Added on Nov 12, 2025

Never used — Read/write

[Delete](#)

```
iteuser@iteuser-VirtualBox:~/.ssh$ sudo apt install git
[sudo] password for iteuser:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  git-man liberror-perl
Suggested packages:
  git-daemon-run | git-daemon-sysvinit git-doc git-email git-gui gitk 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 80 not upgraded.
Need to get 4,806 kB of archives.
After this operation, 24.5 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://ph.archive.ubuntu.com/ubuntu noble/main amd64 liberror-perl all 0.17029-2 [25.6 kB]
Get:2 http://ph.archive.ubuntu.com/ubuntu noble-updates/main amd64 git-man all 1:2.43.0-1ubuntu7.3 [1,100 kB]
Get:3 http://ph.archive.ubuntu.com/ubuntu noble-updates/main amd64 git amd64 1:2.43.0-1ubuntu7.3 [3,680 kB]
86% [3 git 3,254 kB/3,680 kB 88%] 75.2 kB/s 5s
88% [3 git 3,369 kB/3,680 kB 92%] 26.6 kB/s 11s
Fetched 4,806 kB in 1min 26s (56.2 kB/s)
Selecting previously unselected package liberror-perl.
(Reading database ... 189651 files and directories currently installed.)
Preparing to unpack .../liberror-perl_0.17029-2_all.deb ...
Unpacking liberror-perl (0.17029-2) ...
Selecting previously unselected package git-man.
Preparing to unpack .../git-man_1%3a2.43.0-1ubuntu7.3_all.deb ...
Unpacking git-man (1:2.43.0-1ubuntu7.3) ...
Selecting previously unselected package git.
Preparing to unpack .../git_1%3a2.43.0-1ubuntu7.3_amd64.deb ...
Unpacking git (1:2.43.0-1ubuntu7.3) ...
Setting up liberror-perl (0.17029-2) ...
```

```
iteuser@iteuser-VirtualBox:~/.ssh$ git clone git@github.com:ShinjiMochi/CPE201A_DionLopez.git
Cloning into 'CPE201A_DionLopez'...
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$ cat ~/.gitconfig
[user]
    name = ShinjiMochi
    email = dionlopez2306@gmail.com
```

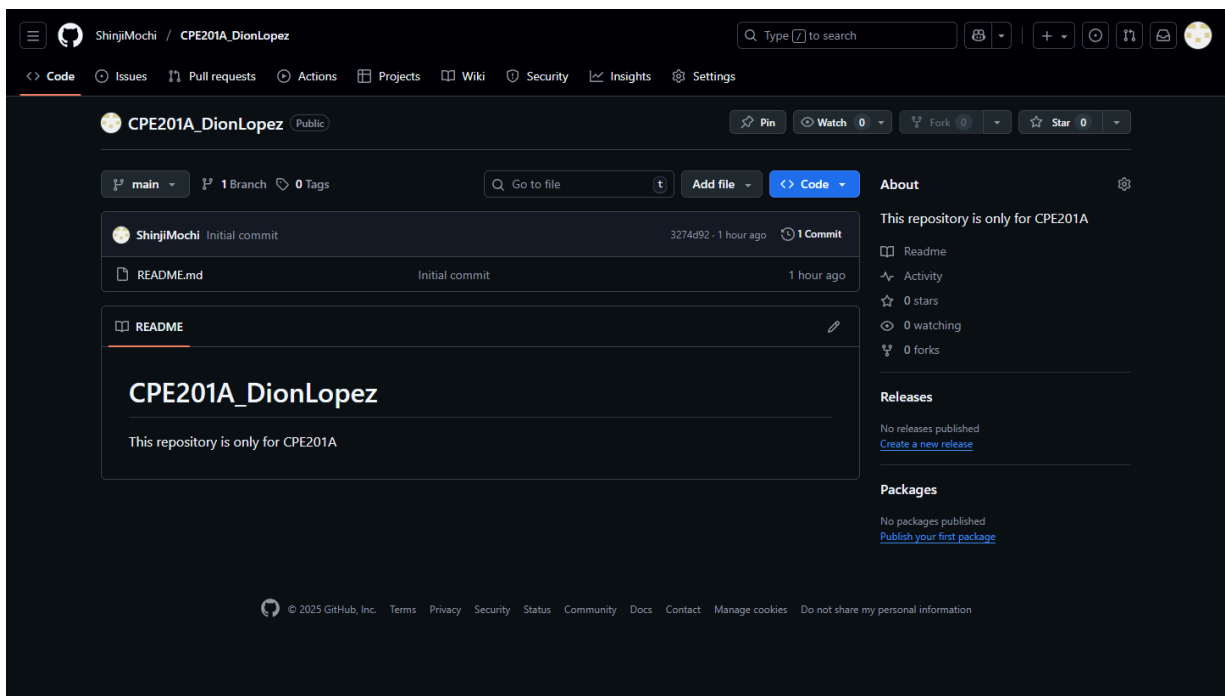
```

iteuser@iteuser-VirtualBox:~/.ssh$ cd ./CPE201A_DionLopez
iteuser@iteuser-VirtualBox:~/.ssh/CPE201A_DionLopez$ git add README.md
iteuser@iteuser-VirtualBox:~/.ssh/CPE201A_DionLopez$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
iteuser@iteuser-VirtualBox:~/.ssh/CPE201A_DionLopez$ git commit -m "Update"
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
iteuser@iteuser-VirtualBox:~/.ssh/CPE201A_DionLopez$ git push origin main
Everything up-to-date

```



6. Conclusions/Learnings/Analysis:

This activity has given me a difficult time due to the multiple roadblocks that I have encountered, but even then I was able to finish and learn from those problems with newfound knowledge on how to solve them, and I understood how to use SSH to link my CLI terminal to my github account to access the repositories that I have created in the github website. This activity has also taught me how to transfer files from my local machine to the repository that is located in my github profile.

7. Assessment Rubric:

--