

Activity No. 14	
SSH Key-Based Authentication and GIT Setup	
Name: Andee Ramon V. Salgado	Date Performed: 11/11/2025
Course Code: CPE 201A	Date Submitted: 11/11/2025
Course Title: Computer System Administration and Troubleshooting	Instructor: Lloyd Aldrin Pornobi
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: <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 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.</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? 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 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

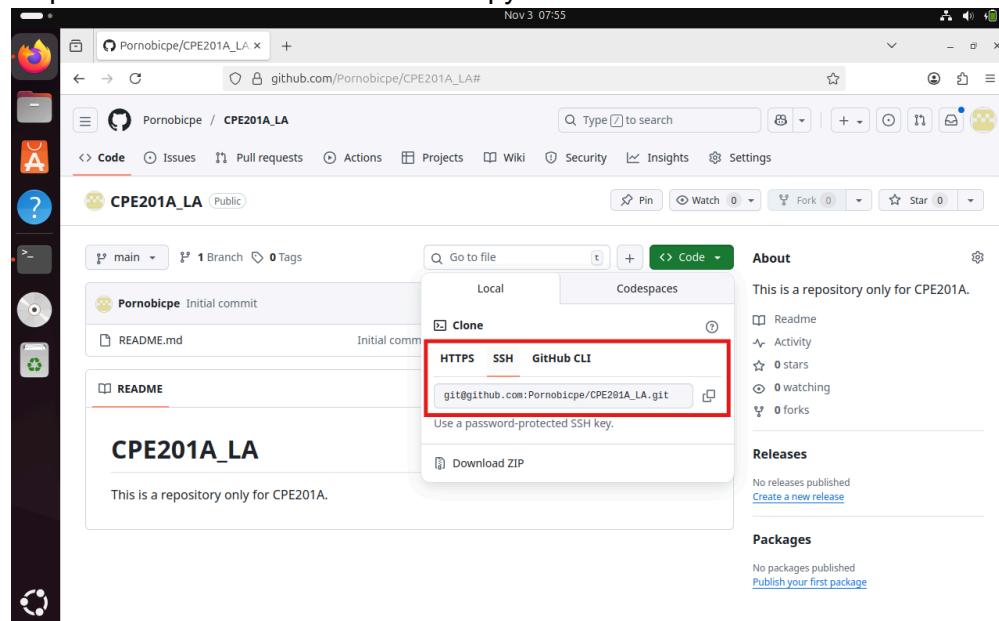
1. Open VirtualBox and start your Ubuntu virtual machine.
2. Log in using your username and password.
3. Open the Terminal.
4. Generate an SSH key pair by typing the following command and pressing Enter:
`ssh-keygen`
5. Navigate to the SSH directory:
`cd ~/.ssh`
6. List the files in the directory:
`ls`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

- 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
- After the installation, issue the command which git again. The directory of git is usually installed in this location: user/bin/git.
- The version of git installed in your device is the latest. Try issuing the command git --version to know the version installed.
- Using the browser in the local machine, go to www.github.com.
- Sign up in case you don't have an account yet. Otherwise, login to your GitHub account.
 - 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.
 - 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.



- 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.
- 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.
- 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
- 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:

```
qmaiko@salgado:~$ ssh-keygen
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/qmaiko/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/qmaiko/.ssh/id_ed25519
Your public key has been saved in /home/qmaiko/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:L8zaeF+aDwMfWewbng0+iDEH+e00FeBdgd083N19+lg qmaiko@salgado
The key's randomart image is:
+--[ED25519 256]--+
|      . o++=|
|      o +...o0|
|      o . = .o|
|      o + .. E|
|      S B = + |
|      o @ O *.. .|
|      = O B .   |
|      +... * .   |
|      o....+...  |
+----[ SHA256 ]-----+
```

```
qmaiko@salgado:~$ cd ~/.ssh  
qmaiko@salgado:~/ssh$ ls  
authorized_keys  id_ed25519  id_ed25519.pub  
qmaiko@salgado:~/ssh$ cat id_ed25519.pub  
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIFFU/UrlaNtH5sdZ+xB0WRHhHVaRuAlBgRgQhroHScJ3 q  
maiko@salgado
```

```
qmaiko@salgado:~$ which git  
/usr/bin/git  
qmaiko@salgado:~$ git --version  
git version 2.43.0
```

The screenshot shows a GitHub repository page for 'CPE201A_Salgado'. The repository is public and has 1 branch and 0 tags. The README file contains the text 'This repository is for CPE201A only'. The repository has 1 commit from 'Salgadocpe' with a timestamp of 2 minutes ago. The repository has 0 stars, 0 forks, and 0 watching.

```
qmaiko@salgado:~$ git clone git@github.com:Salgadocpe/CPE201A_Salgado.git  
Cloning into 'CPE201A_Salgado'...  
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])? 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.  
qmaiko@salgado:~$
```

```
qmaiko@salgado:~/CPE201A_Salgado$ cat ~/.gitconfig  
[user]  
    name = Andee  
    email = qarsalgado@tip.edu.ph
```

```
qmaiko@salgado:~/CPE201A_Salgado$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

qmaiko@salgado:~/CPE201A_Salgado$ nano README.md
qmaiko@salgado:~/CPE201A_Salgado$ 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")

qmaiko@salgado:~/CPE201A_Salgado$ git add README.md
qmaiko@salgado:~/CPE201A_Salgado$ git commit -m "Hello World!"
[main f710950] Hello World!
 1 file changed, 3 insertions(+)
qmaiko@salgado:~/CPE201A_Salgado$ git push origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 308 bytes | 308.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:Salgadocpe/CPE201A_Salgado.git
  bf7da42..f710950  main -> main
qmaiko@salgado:~/CPE201A_Salgado$
```

The screenshot shows a GitHub repository page for a user named 'Salgadocope'. The repository is titled 'Hello World!' and has one branch ('main') and two commits. The most recent commit, 'f10950 · now', was made by 'Salgadocope' and contains the message 'Hello World!'. A file named 'README.md' is present in the repository. The main content area displays the 'README' file, which contains the text 'CPE201A_Salgado', 'This repository is for CPE201A only', and 'Commit test - 1'. Below the repository details, there are two sections: '6. Conclusions/Learnings/Analysis:' and '7. Assessment Rubric:', each with a large empty text area.

main 1 Branch 0 Tags

Go to file t + Code

Salgadocope Hello World! f10950 · now 2 Commits

README.md Hello World! now

README

CPE201A_Salgado

This repository is for CPE201A only

Commit test - 1

6. Conclusions/Learnings/Analysis:

I've used GitHub before, and it's really cool to see how people make the repositories that I get files from on GitHub.

7. Assessment Rubric: