Name: Alexis Neil D. Deneiga	**Date Performed:** August 15th, 2025
Course/Section: CPE 212/CPE31S4	**Date Submitted:** August 15th, 2025
Instructor: Engr. Robin Valenzuela	**Semester and SY:** SY 2025-26 1st Sem.

Activity 2: SSH Key-Based Authentication and Setting up Git

1. **Objectives:**

1.1 Configure remote and local machine to connect via SSH using a KEY instead of using a password

1.2 Create a public key and private key

1.3 Verify connectivity

1.4 Setup Git Repository using local and remote repositories

1.5 Configure and Run ad hoc commands from local machine to remote servers

Part 1: Discussion

It is assumed that you are already done with the last Activity (**Activity 1: Configure Network using Virtual Machines).** *Provide screenshots for each task*.

It is also assumed that you have VMs running that you can SSH but requires 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.

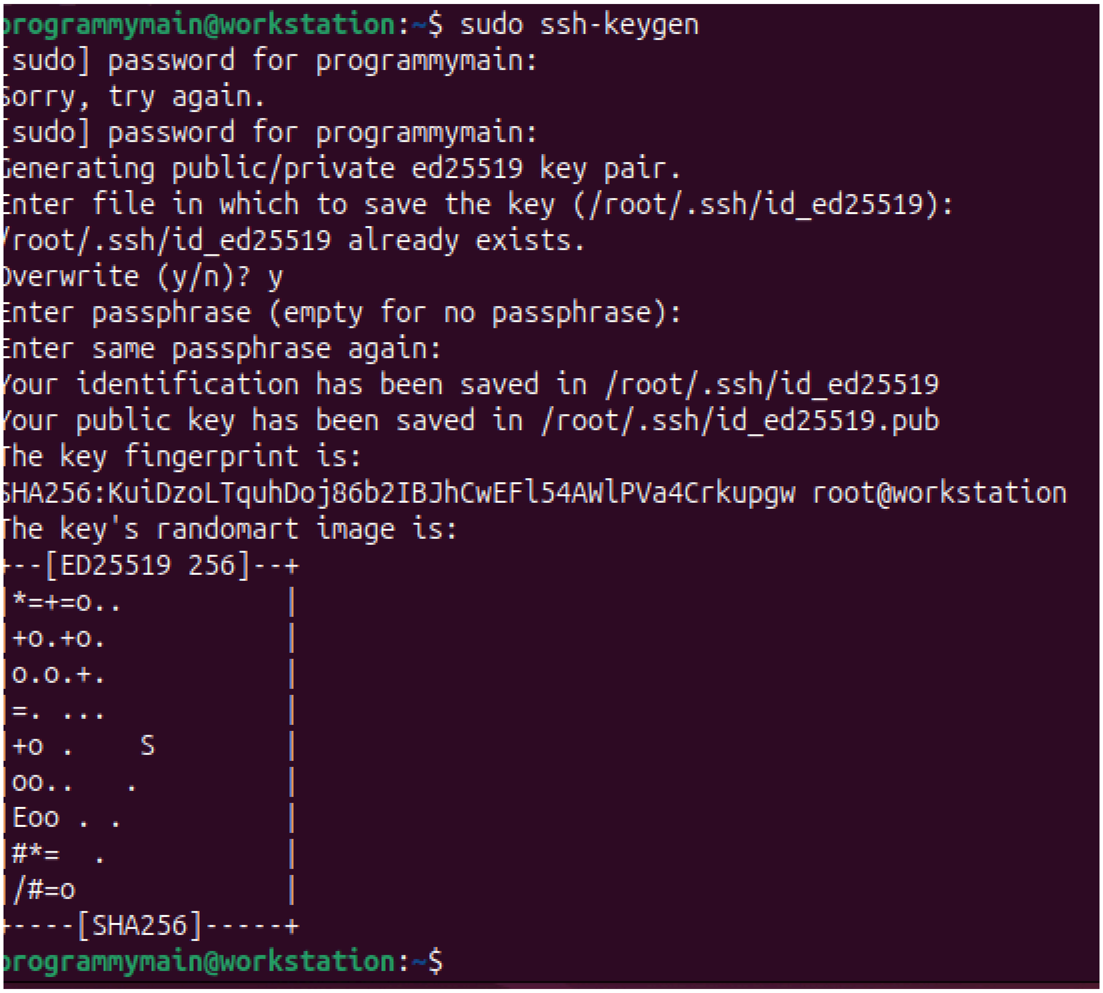
SSH introduced public key authentication as a more secure alternative to the older .rhosts authentication. It improved security by avoiding the need to have password stored in files and eliminated the possibility of a compromised server stealing the user's password.

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.

Task 1: Create an SSH Key Pair for User Authentication

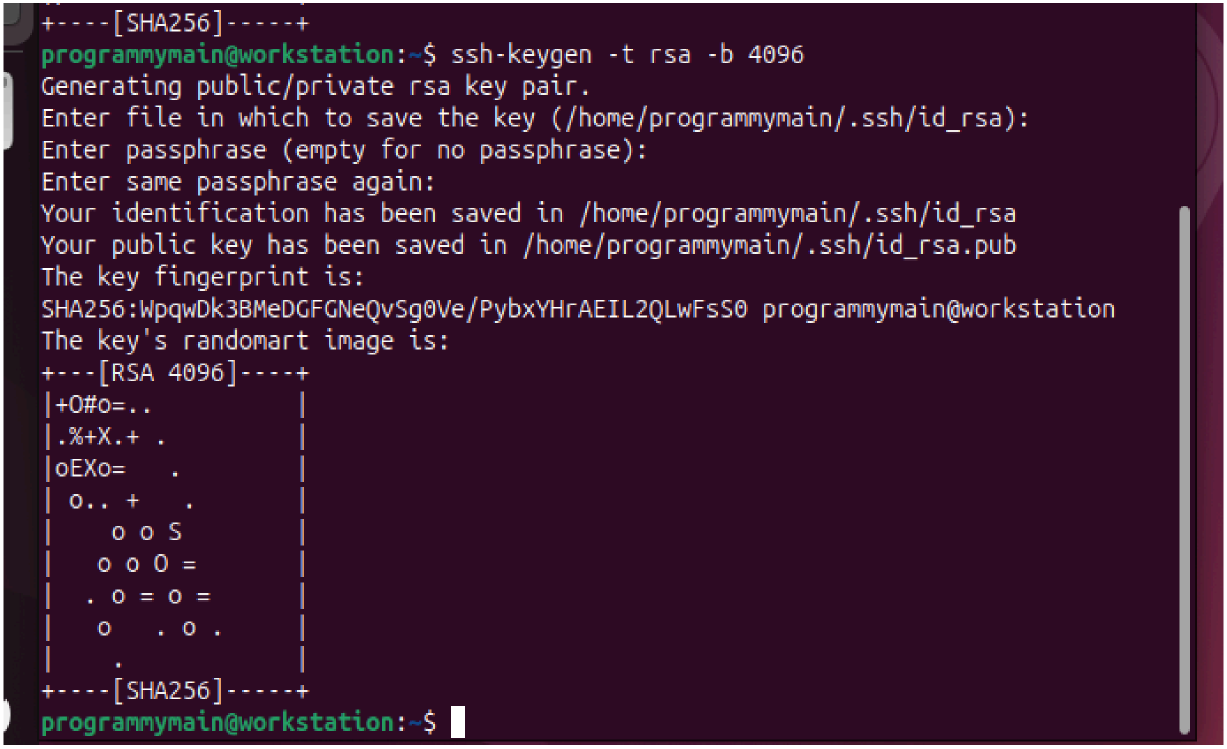
1. The simplest way to generate a key pair is to run *ssh-keygen* without arguments. In this case, it will prompt for the file in which to store keys. First,

the tool asked where to save the file. SSH keys for user authentication are usually stored in the users .ssh directory under the home directory. However, in enterprise environments, the location is often different. The default key file name depends on the algorithm, in this case *id_rsa* when using the default RSA algorithm. It could also be, for example, *id_dsa* or *id_ecdsa*.



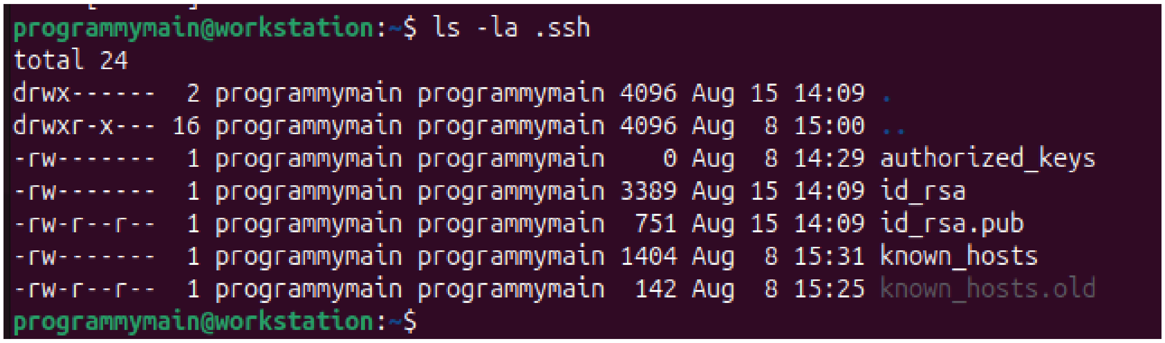
```
programmymain@workstation:~$ sudo ssh-keygen
[sudo] password for programmymain:
Sorry, try again.
[sudo] password for programmymain:
Generating public/private ed25519 key pair.
Enter file in which to save the key (/root/.ssh/id_ed25519):
/root/.ssh/id_ed25519 already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_ed25519
Your public key has been saved in /root/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:KuiDzoLTquhDoj86b2IBJhCwEFl54AWlPVa4Crkupgw root@workstation
The key's randomart image is:
+--[ED25519 256]--+
|*=+=o..          |
|+o.+o.           |
|o.o.+.           |
|=. ...           |
|+o .    S        |
|oo..   .         |
|Eoo . .          |
|#*=  .           |
|/#=o             |
+----[SHA256]-----+
programmymain@workstation:~$
```

2. Issue the command *ssh-keygen -t rsa -b 4096.* The algorithm is selected using the -t option and key size using the -b option.
3. When asked for a passphrase, just press enter. The passphrase is used for encrypting the key, so that it cannot be used even if someone obtains the private key file. The passphrase should be cryptographically strong.



```
+----[SHA256]-----+
programmymain@workstation:~$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/programmymain/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/programmymain/.ssh/id_rsa
Your public key has been saved in /home/programmymain/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:WpqwDk3BMeDGFGNeQvSg0Ve/PybxYHrAEIL2QLwFsS0 programmymain@workstation
The key's randomart image is:
+---[RSA 4096]----+
|+O#o=..          |
|.%+X.+ .         |
|oEXo=   .        |
| o.. +   .       |
|    o o S        |
|   o o O =       |
|  . o = o =      |
|   o   . o .     |
|    .            |
+----[SHA256]-----+
programmymain@workstation:~$ 
```

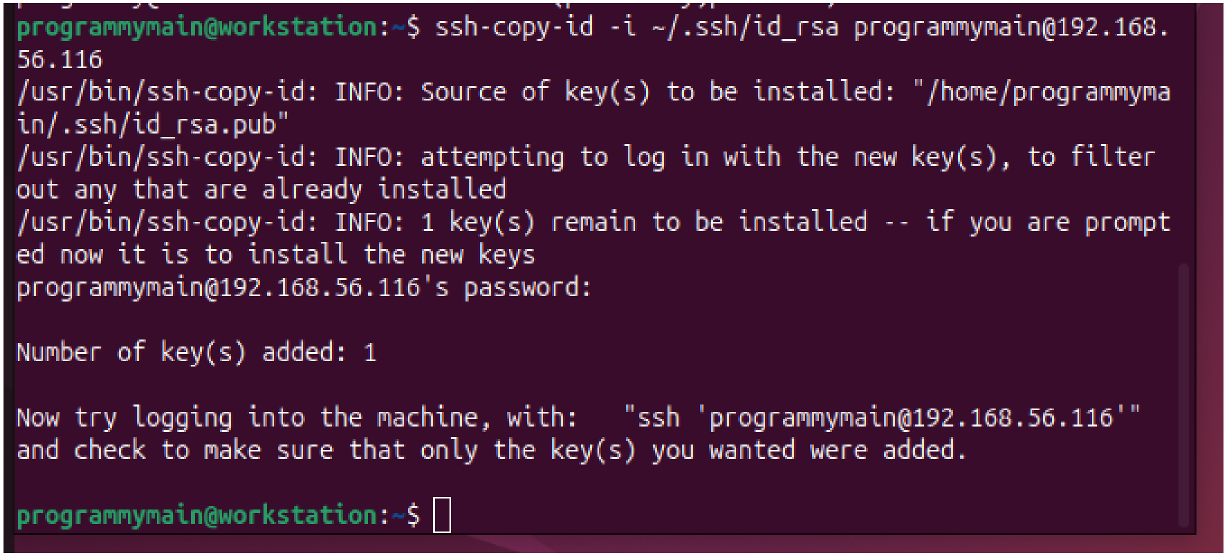
4. Verify that you have created the key by issuing the command *ls -la .ssh.* The command should show the .ssh directory containing a pair of keys. For example, id_rsa.pub and id_rsa.



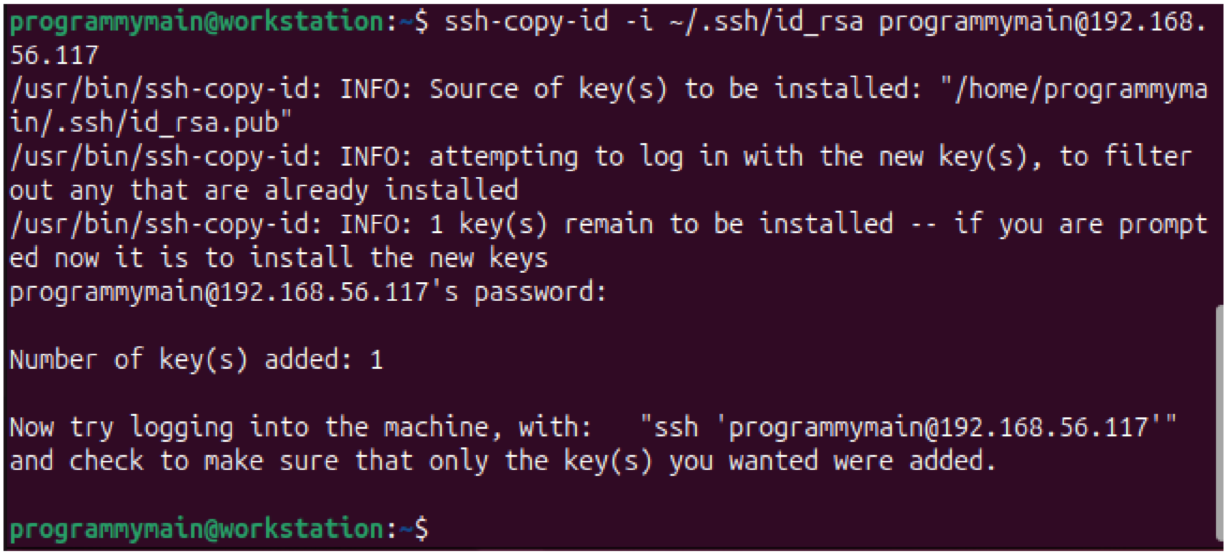
```
programmymain@workstation:~$ ls -la .ssh
total 24
drwx------  2 programmymain programmymain 4096 Aug 15 14:09 .
drwxr-x--- 16 programmymain programmymain 4096 Aug  8 15:00 ..
-rw-------  1 programmymain programmymain    0 Aug  8 14:29 authorized_keys
-rw-------  1 programmymain programmymain 3389 Aug 15 14:09 id_rsa
-rw-r--r--  1 programmymain programmymain  751 Aug 15 14:09 id_rsa.pub
-rw-------  1 programmymain programmymain 1404 Aug  8 15:31 known_hosts
-rw-r--r--  1 programmymain programmymain  142 Aug  8 15:25 known_hosts.old
programmymain@workstation:~$
```

Task 2: Copying the Public Key to the remote servers

1. To use public key authentication, the public key must be copied to a server and installed in an *authorized_keys* file. This can be conveniently done using the *ssh-copy-id* tool.
2. Issue the command similar to this: *ssh-copy-id -i ~/.ssh/id_rsa user@host*



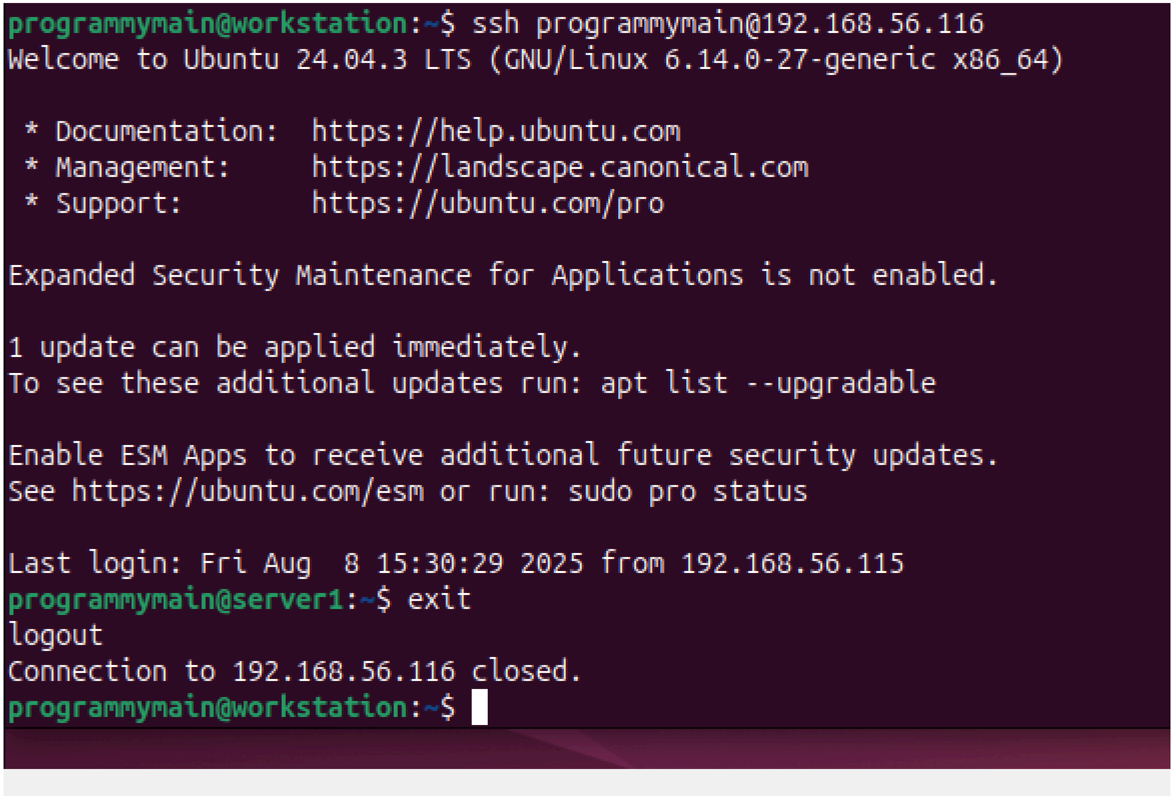
```
programmymain@workstation:~$ ssh-copy-id -i ~/.ssh/id_rsa programmymain@192.168.
56.116
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/programmyma
in/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter
out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompt
ed now it is to install the new keys
programmymain@192.168.56.116's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'programmymain@192.168.56.116'"
and check to make sure that only the key(s) you wanted were added.

programmymain@workstation:~$
```



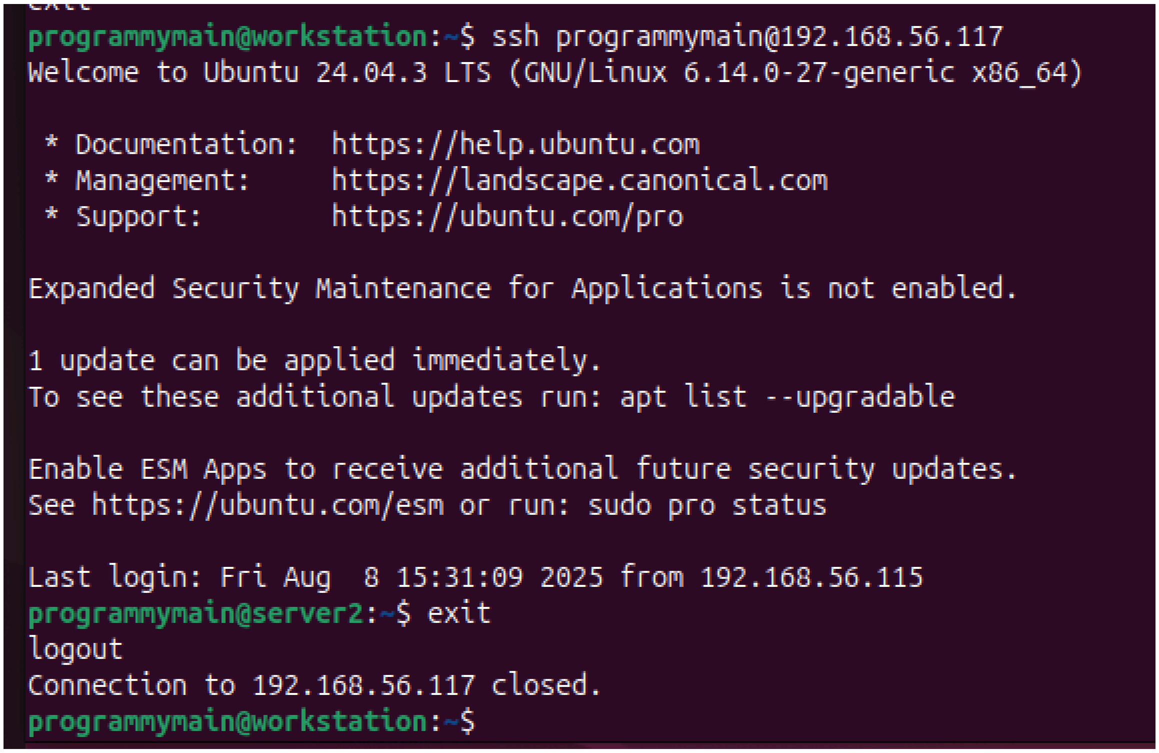
```
programmymain@workstation:~$ ssh-copy-id -i ~/.ssh/id_rsa programmymain@192.168.
56.117
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/programmyma
in/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter
out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompt
ed now it is to install the new keys
programmymain@192.168.56.117's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'programmymain@192.168.56.117'"
and check to make sure that only the key(s) you wanted were added.

programmymain@workstation:~$
```

3. Once the public key has been configured on the server, the server will allow any connecting user that has the private key to log in. During the login process, the client proves possession of the private key by digitally signing the key exchange
4. On the local machine, verify that you can SSH with Server 1 and Server 2. What did you notice? Did the connection ask for a password? If not, why?

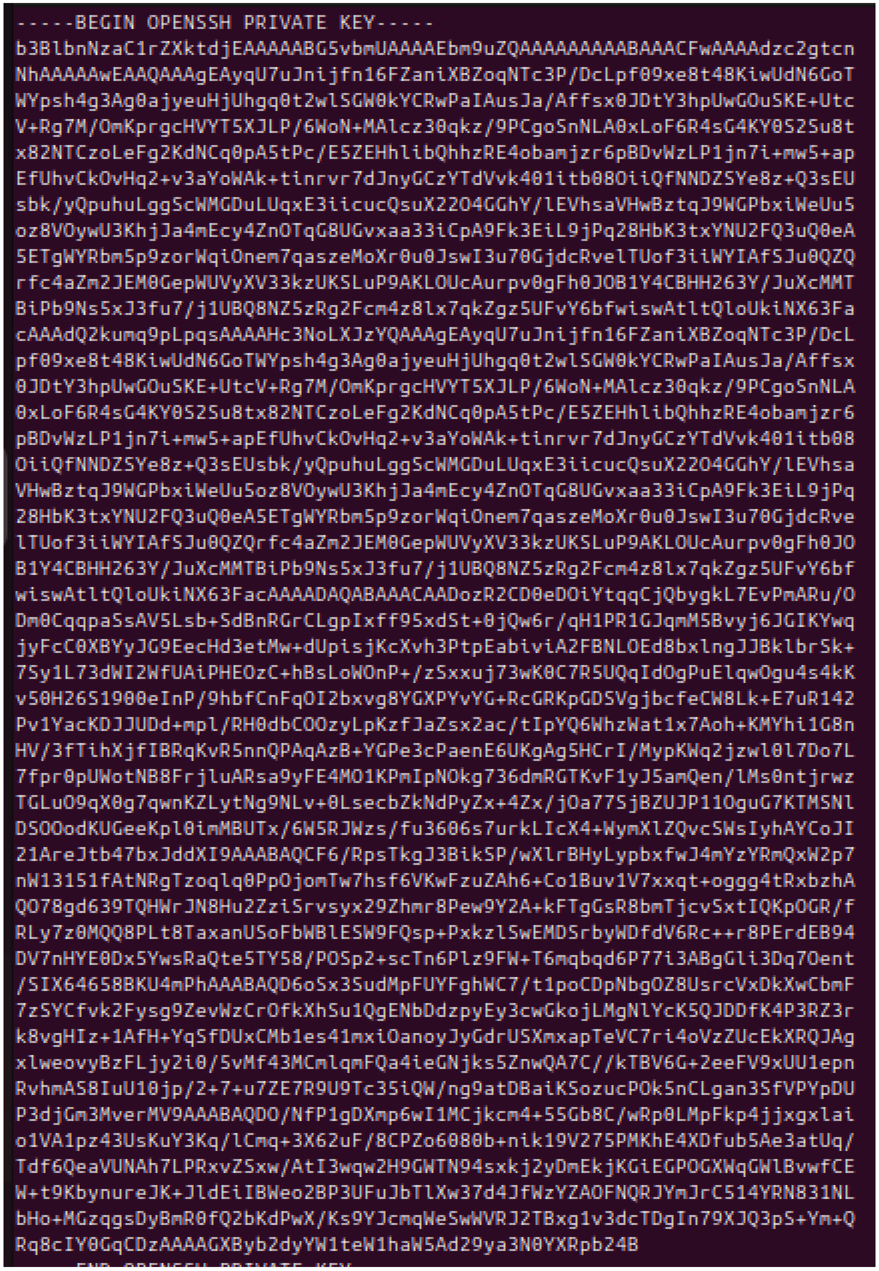
I noticed that it did not ask for a password, it immediately went to the station itself. This is probably because the public-private key combo is already a secure way to transmit identity with one another, making it easier to login to remote servers.



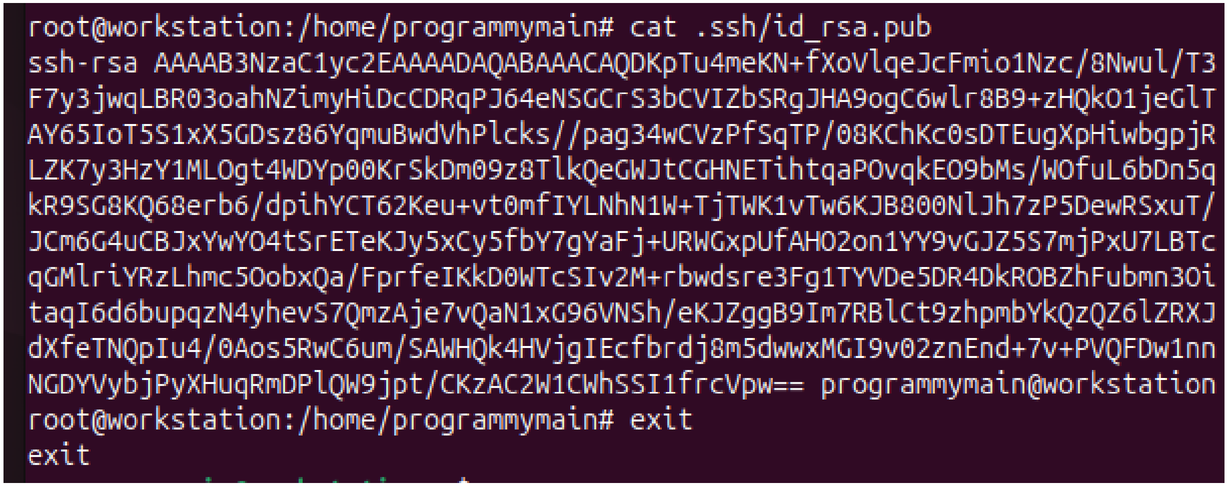
```
programmymain@workstation:~$ ssh programmymain@192.168.56.116
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-27-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

1 update can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Fri Aug  8 15:30:29 2025 from 192.168.56.115
programmymain@server1:~$ exit
logout
Connection to 192.168.56.116 closed.
programmymain@workstation:~$ 
```



```
exit
programmymain@workstation:~$ ssh programmymain@192.168.56.117
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-27-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

1 update can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Fri Aug  8 15:31:09 2025 from 192.168.56.115
programmymain@server2:~$ exit
logout
Connection to 192.168.56.117 closed.
programmymain@workstation:~$
```



```
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAACFwAAAAdzc2gtcn
NhAAAAAwEAAQAAAgEAyqU7uJnijfn16FZaniXBZoqNTc3P/DcLpf09xe8t48KiwUdN6GoT
WYpsh4g3Ag0ajyeuHjUhgq0t2wlSGW0kYCRwPaIAusJa/Affsx0JDtY3hpUwGOuSKE+Utc
V+Rg7M/OmKprgcHVYT5XJLP/6WoN+MAlcz30qkz/9PCgoSnNLA0xLoF6R4sG4KY0S2Su8t
x82NTCzoLeFg2KdNCq0pA5tPc/E5ZEHhlibQhhzRE4obamjzr6pBDvWzLP1jn7i+mw5+ap
EfUhvCkOvHq2+v3aYoWAk+tinrvr7dJnyGCzYTdVvk401itb08OiiQfNNDZSYe8z+Q3sEU
sbk/yQpuhuLggScWMGDuLUqxE3iicucQsuX22O4GGhY/lEVhsaVHwBztqJ9WGPbxiWeUu5
oz8VOywU3KhjJa4mEcy4ZnOTqG8UGvxaa33iCpA9Fk3EiL9jPq28HbK3txYNU2FQ3uQ0eA
5ETgWYRbm5p9zorWqiOnem7qaszeMoXr0u0JswI3u70GjdcRvelTUof3iiWYIAfSJu0QZQ
rfc4aZm2JEM0GepWUVyXV33kzUKSLuP9AKLOUcAurpv0gFh0JOB1Y4CBHH263Y/JuXcMMT
BiPb9Ns5xJ3fu7/j1UBQ8NZ5zRg2Fcm4z8lx7qkZgz5UFvY6bfwiswAtltQloUkiNX63Fa
cAAAdQ2kumq9pLpqsAAAAHc3NoLXJzYQAAAgEAyqU7uJnijfn16FZaniXBZoqNTc3P/DcL
pf09xe8t48KiwUdN6GoTWYpsh4g3Ag0ajyeuHjUhgq0t2wlSGW0kYCRwPaIAusJa/Affsx
0JDtY3hpUwGOuSKE+UtcV+Rg7M/OmKprgcHVYT5XJLP/6WoN+MAlcz30qkz/9PCgoSnNLA
0xLoF6R4sG4KY0S2Su8tx82NTCzoLeFg2KdNCq0pA5tPc/E5ZEHhlibQhhzRE4obamjzr6
pBDvWzLP1jn7i+mw5+apEfUhvCkOvHq2+v3aYoWAk+tinrvr7dJnyGCzYTdVvk401itb08
OiiQfNNDZSYe8z+Q3sEUsbk/yQpuhuLggScWMGDuLUqxE3iicucQsuX22O4GGhY/lEVhsa
VHwBztqJ9WGPbxiWeUu5oz8VOywU3KhjJa4mEcy4ZnOTqG8UGvxaa33iCpA9Fk3EiL9jPq
28HbK3txYNU2FQ3uQ0eA5ETgWYRbm5p9zorWqiOnem7qaszeMoXr0u0JswI3u70GjdcRve
lTUof3iiWYIAfSJu0QZQrfc4aZm2JEM0GepWUVyXV33kzUKSLuP9AKLOUcAurpv0gFh0JO
B1Y4CBHH263Y/JuXcMMTBiPb9Ns5xJ3fu7/j1UBQ8NZ5zRg2Fcm4z8lx7qkZgz5UFvY6bf
wiswAtltQloUkiNX63FacAAAADAQABAAACAADozR2CD0eDOiYtqqCjQbygkL7EvPmARu/O
Dm0CqqpaSsAV5Lsb+SdBnRGrCLgpIxff95xd5t+0jQw6r/qH1PR1GJqmM5Bvyj6JGIKYwq
jyFcC0XBYyJG9EecHd3etMw+dUpisjKcXvh3PtpEabiviA2FBNLOEd8bxlngJJBklbrSk+
7Sy1L73dWI2WfUAiPHEOzC+hBsLoWOnP+/z5xxuj73wK0C7R5UQqIdOgPuElqwOgu4s4kK
v50H26S1900eInP/9hbfCnFqOI2bxvg8YGXPYvYG+RcGRKpGDSVgjbcfeCW8Lk+E7uR142
Pv1YacKDJJUDd+mpl/RH0dbCOOzyLpKzfJaZsx2ac/tIpYQ6WhzWat1x7Aoh+KMYhi1G8n
HV/3fTihXjfIBRqKvR5nnQPAqAzB+YGPe3cPaenE6UKgAg5HCrI/MypKWq2jzwl0l7Do7L
7fpr0pUWotNB8FrjluARsa9yFE4MO1KPmIpNOkg736dmRGTKvF1yJ5amQen/lMs0ntjrwz
TGLuO9qX0g7qwnKZLytNg9NLv+0LsecbZkNdPyZx+4Zx/jOa77SjBZUJP11OguG7KTMSNl
DSOOodKUGeeKpl0imMBUTx/6W5RJWzs/fu3606s7urkLIcX4+WymXlZQvcSWsIyhAYCoJI
21AreJtb47bxJddXI9AAABAQCF6/RpsTkgJ3BikSP/wXlrBHyLypbxfwJ4mYzYRmQxW2p7
nW13151fAtNRgTzoqlq0PpOjomTw7hsf6VKwFzuZAh6+Co1Buv1V7xxqt+oggg4tRxbzhA
QO78gd639TQHWrJN8Hu2ZziSrvsyx29Zhmr8Pew9Y2A+kFTgGsR8bmTjcvSxtIQKpOGR/f
RLy7z0MQQ8PLt8TaxanUSoFbWBlESW9FQsp+PxkzlSwEMDSrbyWDfdV6Rc++r8PErdEB94
DV7nHYE0Dx5YwsRaQte5TY58/POSp2+scTn6Plz9FW+T6mqbqd6P77i3ABgGli3Dq7Oent
/SIX64658BKU4mPhAAABAQD6oSx3SudMpFUYFghWC7/t1poCDpNbgOZ8UsrcVxDkXwCbmF
7zSYCfvk2Fysg9ZevWzCrOfkXhSu1QgENbDdzpyEy3cwGkojLMgNlYcK5QJDDfK4P3RZ3r
k8vgHIz+1AfH+YqSfDUxCMb1es41mxiOanoyJyGdrUSXmxapTeVC7ri4oVzZUcEkXRQJAg
xlweovyBzFLjy2i0/5vMf43MCmlqmFQa4ieGNjks5ZnwQA7C//kTBV6G+2eeFV9xUU1epn
RvhmAS8IuU10jp/2+7+u7ZE7R9U9Tc35iQW/ng9atDBaiKSozucPOk5nCLgan3SfVPYpDU
P3djGm3MverMV9AAABAQDO/NfP1gDXmp6wI1MCjkcm4+55Gb8C/wRp0LMpFkp4jjxgxlai
o1VA1pz43UsKuY3Kq/lCmq+3X62uF/8CPZo6080b+nik19V275PMKhE4XDfub5Ae3atUq/
Tdf6QeaVUNAh7LPRxvZSxw/AtI3wqw2H9GWTN94sxkj2yDmEkjKGiEGPOGXWqGWlBvwfCE
W+t9KbynureJK+JldEiIBWeo2BP3UFuJbTlXw37d4JfWzYZAOFNQRJYmJrC514YRN831NL
bHo+MGzqgsDyBmR0fQ2bKdPwX/Ks9YJcmqWeSwWVRJ2TBxg1v3dcTDgIn79XJQ3pS+Ym+Q
Rq8cIY0GqCDzAAAAGXByb2dyYW1teW1haW5Ad29ya3N0YXRpb24B
-----END OPENSSH PRIVATE KEY-----
```



```
root@workstation:/home/programmymain# cat .ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAACAQDKpTu4meKN+fXoVlqeJcFmio1Nzc/8Nwul/T3
F7y3jwqLBR03oahNZimyHiDcCDRqPJ64eNSGCrS3bCVIZbSRgJHA9ogC6wlr8B9+zHQkO1jeGlT
AY65IoT5S1xX5GDsz86YqmuBwdVhPlcks//pag34wCVzPfSqTP/08KChKc0sDTEugXpHiwbgpjR
LZK7y3HzY1MLOgt4WDYp00KrSkDm09z8TlkQeGWJtCGHNETihtqaPOvqkEO9bMs/WOfuL6bDn5q
kR9SG8KQ68erb6/dpihYCT62Keu+vt0mfIYLNhN1W+TjTWK1vTw6KJB800NlJh7zP5DewRSxuT/
JCm6G4uCBJxYwYO4tSrETeKJy5xCy5fbY7gYaFj+URWGxpUfAHO2on1YY9vGJZ5S7mjPxU7LBTc
qGMlriYRzLhmc5OobxQa/FprfeIKkD0WTcSIv2M+rbwdsre3Fg1TYVDe5DR4DkROBZhFubmn3Oi
taqI6d6bupqzN4yhevS7QmzAje7vQaN1xG96VNSh/eKJZggB9Im7RBlCt9zhpmbYkQzQZ6lZRXJ
dXfeTNQpIu4/0Aos5RwC6um/SAWHQk4HVjgIEcfbrdj8m5dwwxMGI9v02znEnd+7v+PVQFDw1nn
NGDYVybjPyXHuqRmDPlQW9jpt/CKzAC2W1CWhSSI1frcVpw== programmymain@workstation
root@workstation:/home/programmymain# exit
exit
```

Reflections:

Answer the following:

1. How will you describe the ssh-program? What does it do?

SSH is a secure way to login to remote servers. It uses a public-private key, where the public and private key is mangled to create a unique string that only the host and remote server can only access. This is usually safer than just a simple password as it has hashes that take longer than human life to bruteforce.

2. How do you know that you already installed the public key to the remote servers?

After installing the public key, I would SSH the server. If there is a password request, then there's something wrong. Otherwise, it is a success.

Part 2: Discussion

Provide screenshots for each task.

It is assumed that you are done with the last activity (**Activity 2: SSH Key-Based Authentication**).

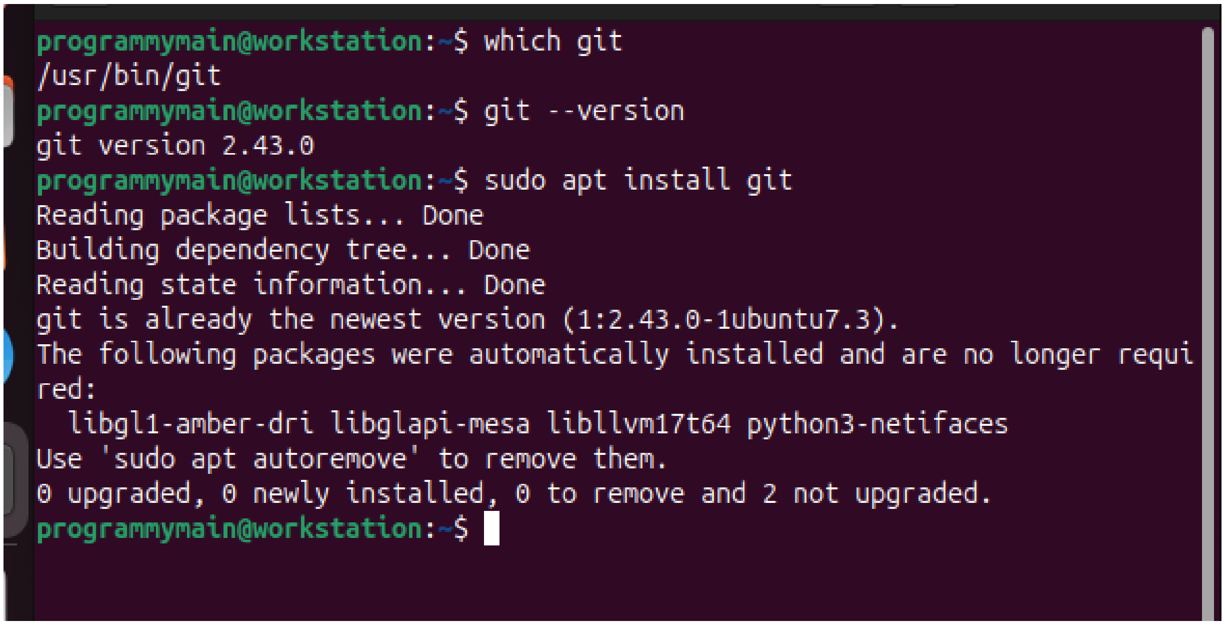
Set up Git

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

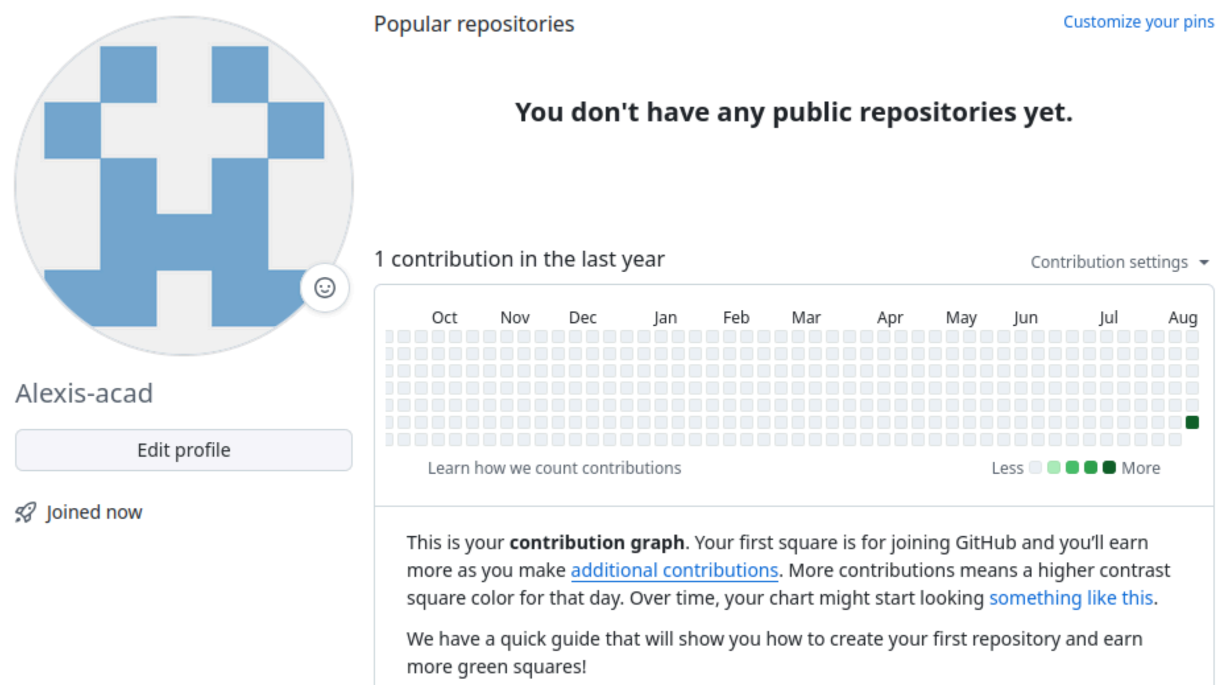
Task 3: Set up the Git Repository

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.



```
programmymain@workstation:~$ which git
/usr/bin/git
programmymain@workstation:~$ git --version
git version 2.43.0
programmymain@workstation:~$ sudo apt install git
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
git is already the newest version (1:2.43.0-1ubuntu7.3).
The following packages were automatically installed and are no longer requi
red:
  libgl1-amber-dri libglapi-mesa libllvm17t64 python3-netifaces
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.
programmymain@workstation:~$
```

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.



Popular repositories

Customize your pins

You don't have any public repositories yet.

Alexis-acad

Edit profile

Joined now

1 contribution in the last year

Contribution settings

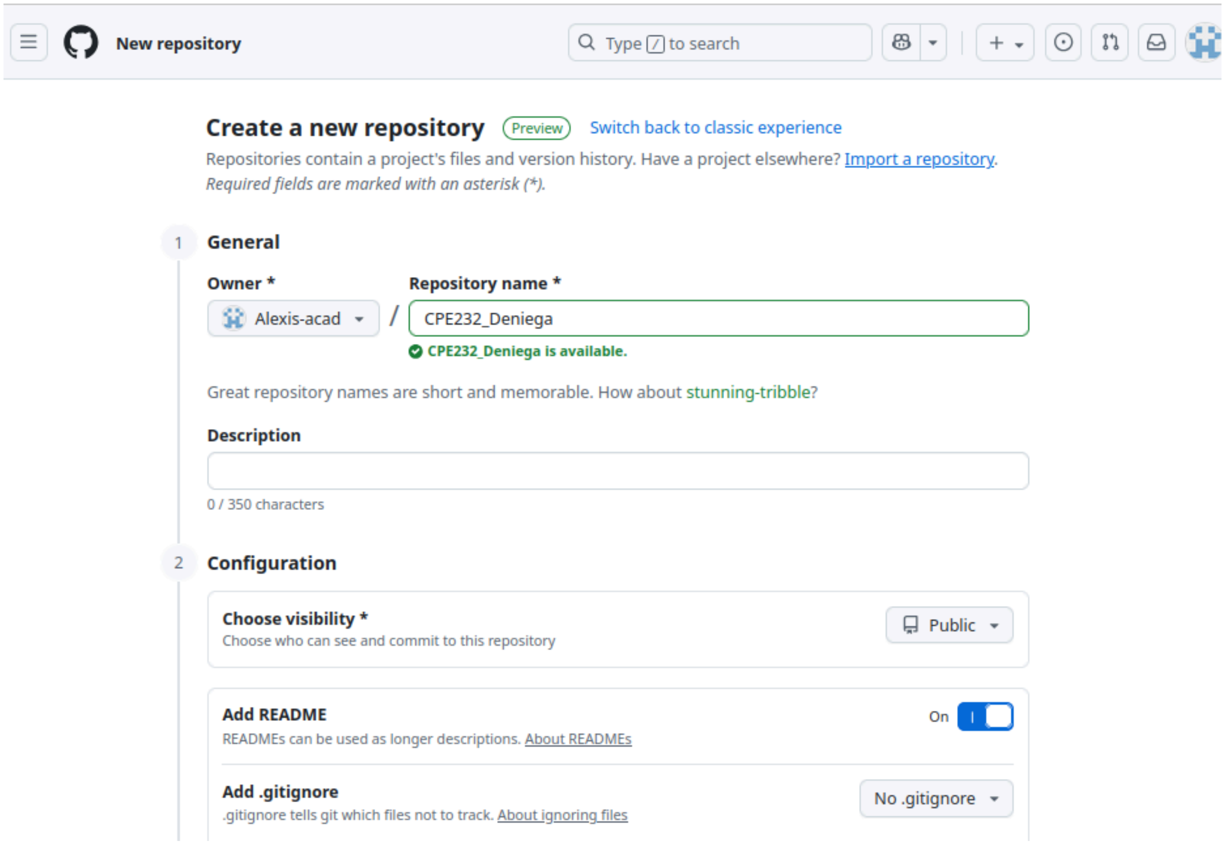
Learn how we count contributions

Less 🟩🟩🟩🟩 More

This is your **contribution graph**. Your first square is for joining GitHub and you'll earn more as you make additional contributions. More contributions means a higher contrast square color for that day. Over time, your chart might start looking something like this.

We have a quick guide that will show you how to create your first repository and earn more green squares!

a. Create a new repository and name it as CPE232_yourname. Check Add a README file and click Create repository.



Create a new repository Preview Switch back to classic experience

Repositories contain a project's files and version history. Have a project elsewhere? Import a repository.

Required fields are marked with an asterisk (*).

1 **General**

Owner * Alexis-acad / **Repository name *** CPE232_Deniega

CPE232_Deniega is available.

Great repository names are short and memorable. How about stunning-tribble?

Description

0 / 350 characters

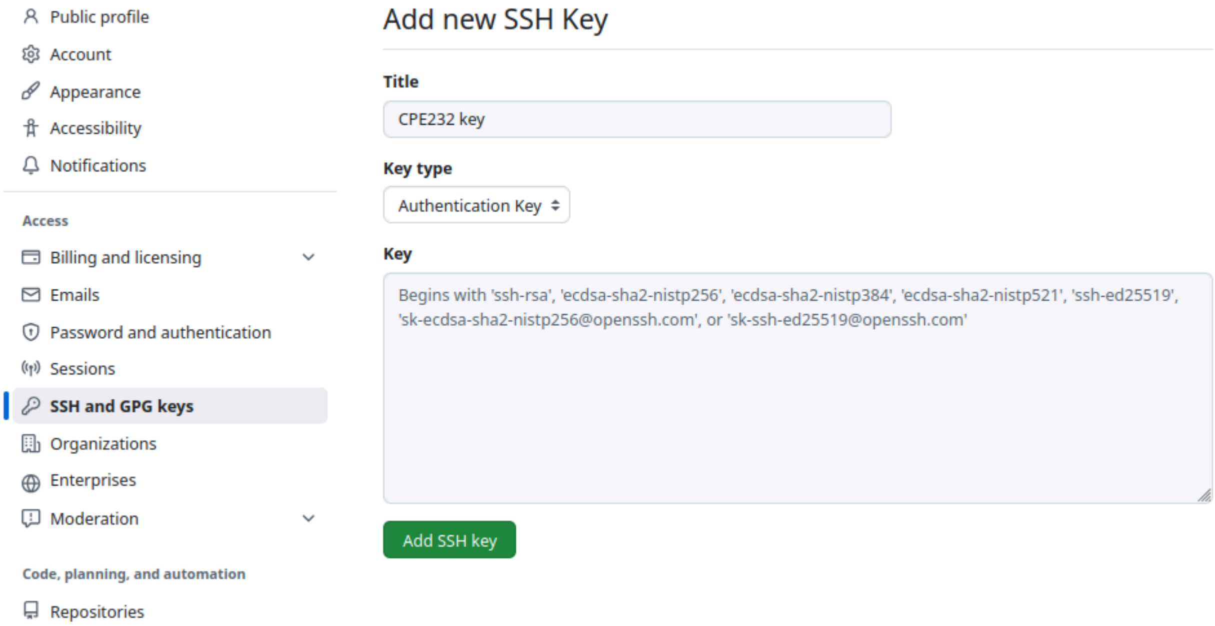
2 **Configuration**

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

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

Add .gitignore — No .gitignore
.gitignore tells git which files not to track. About ignoring files

b. Create a new SSH key on GitHub. Go your profile's setting and click SSH and GPG keys. If there is an existing key, make sure to delete it. To create a new SSH keys, click New SSH Key. Write CPE232 key as the title of the key.



Add new SSH Key

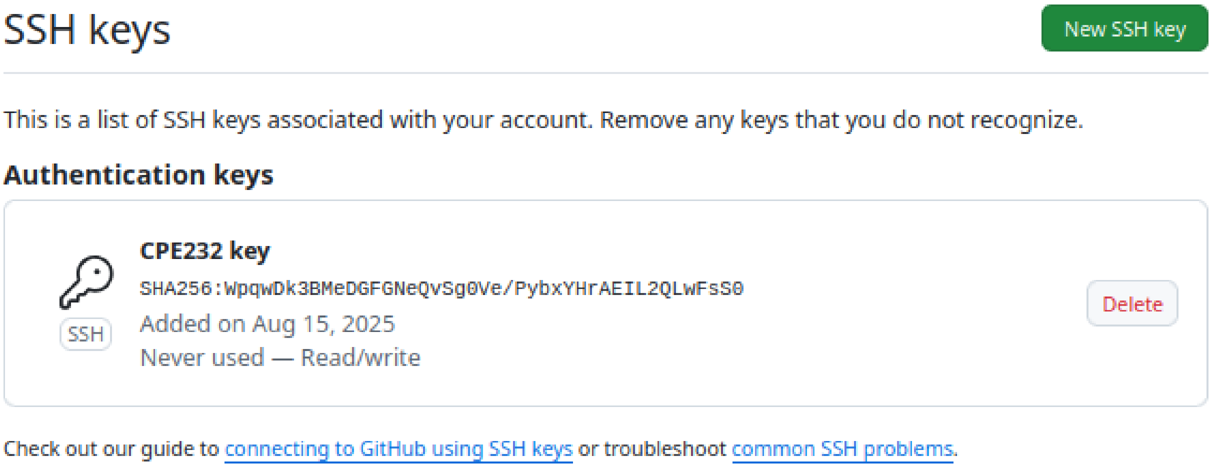
Title: CPE232 key

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'

c. On the local machine's terminal, issue the command cat .ssh/id_rsa.pub and copy the public key. Paste it on the GitHub key and press Add SSH key.



SSH keys

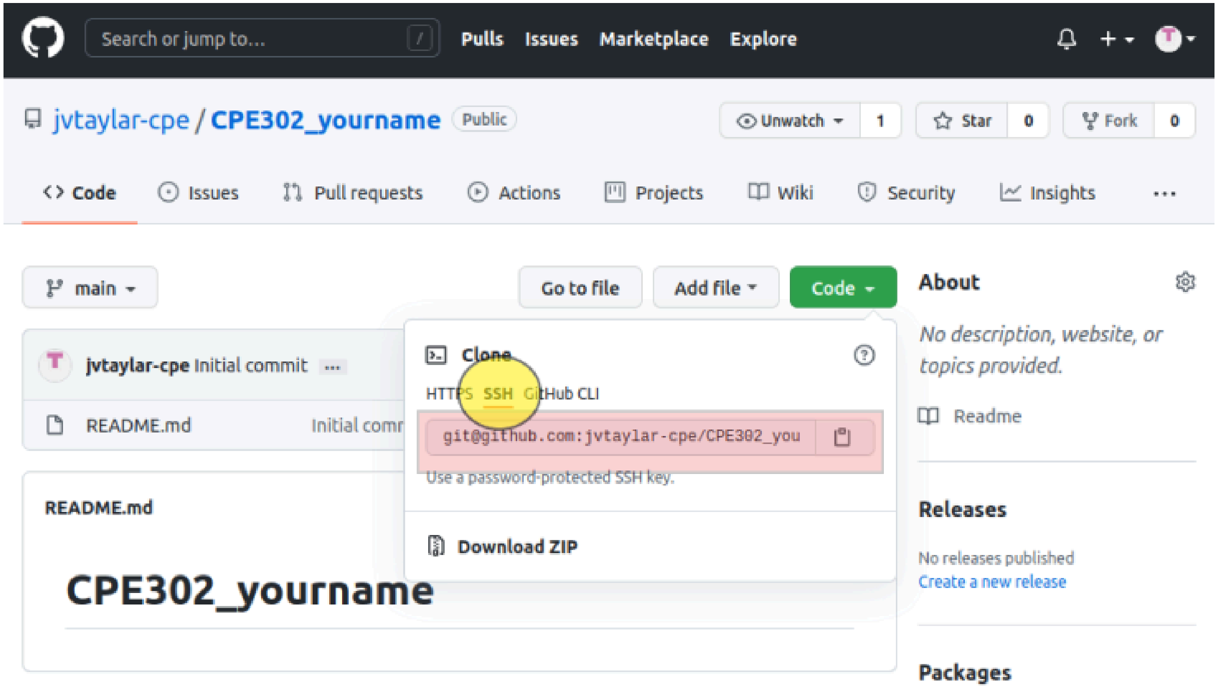
This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

Authentication keys

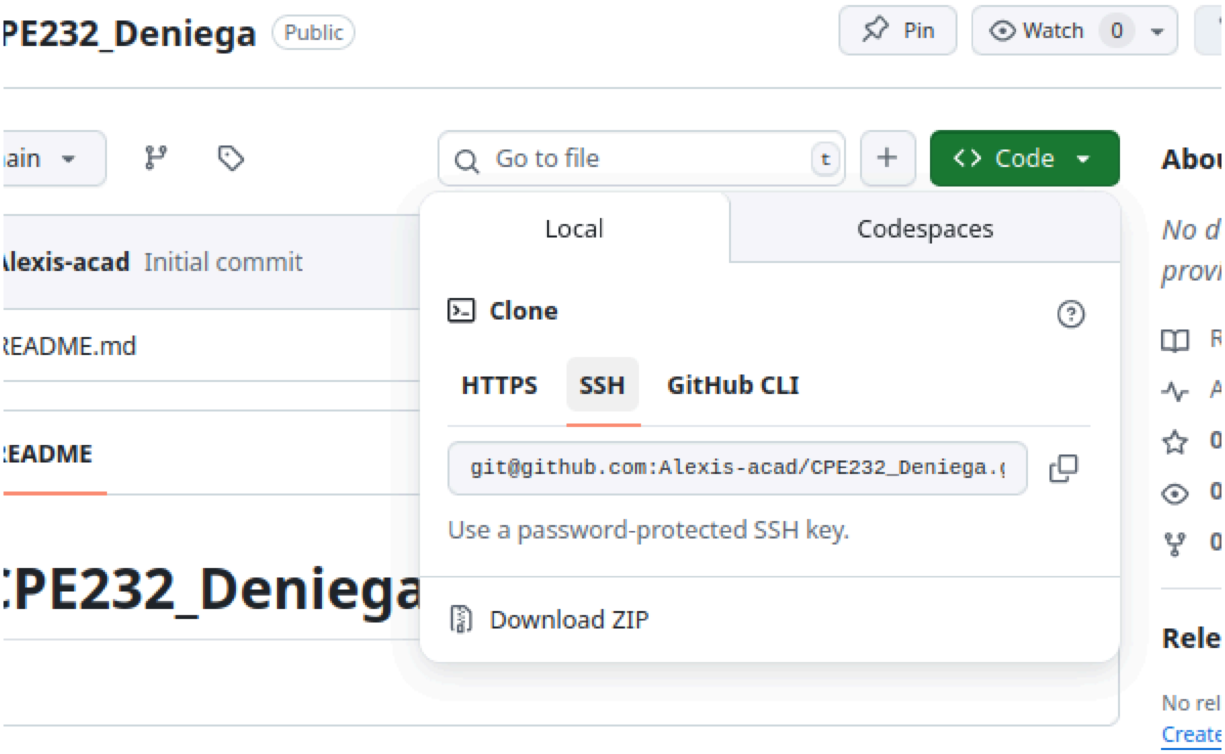
CPE232 key
SHA256:WpqwDk3BMeDGFGNeQvSg0Ve/PybxYHrAEIL2QLwFsS0
Added on Aug 15, 2025
Never used — Read/write

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

d. 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.

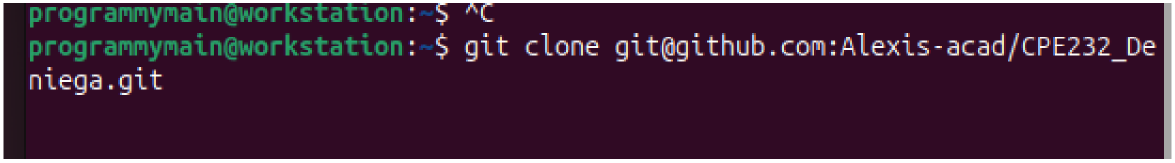


```
git@github.com:jvtaylar-cpe/CPE302_you
```

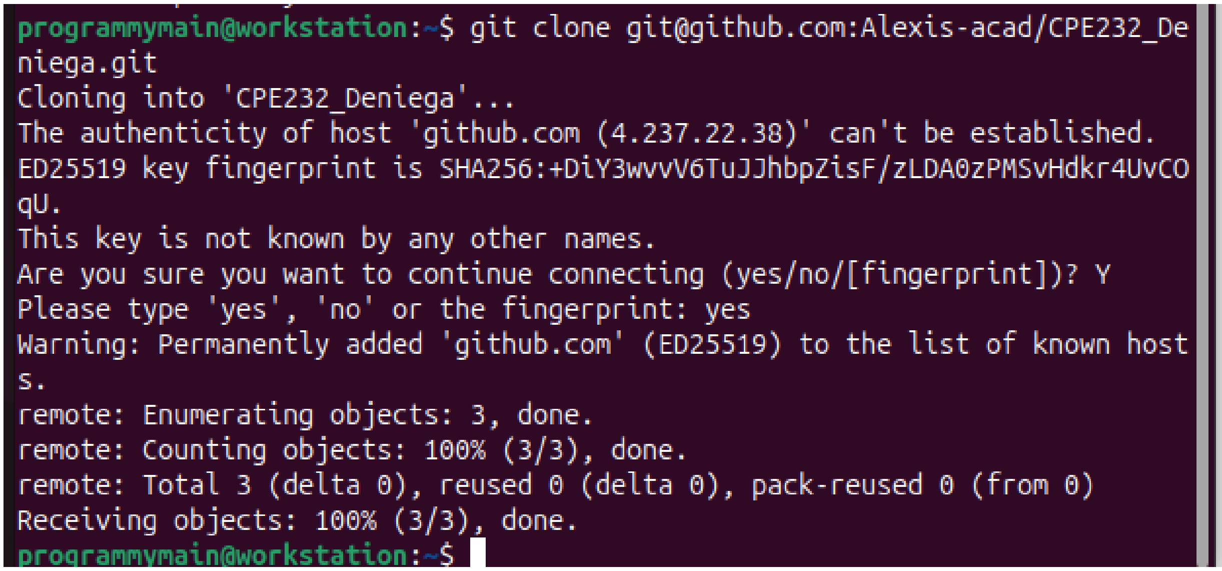


```
git@github.com:Alexis-acad/CPE232_Deniega.(
```

e. Issue the command git clone followed by the copied link. For example, *git clone git@github.com:jvtaylar-cpe/CPE232_yourname.git*. When prompted to continue connecting, type yes and press enter.

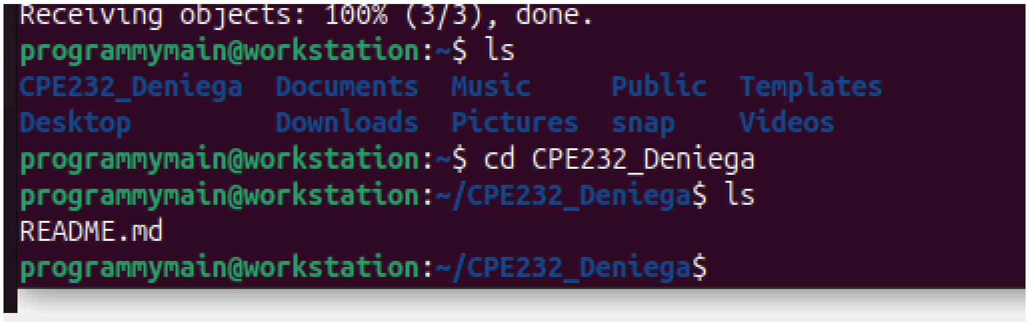


```
programmymain@workstation:~$ ^C
programmymain@workstation:~$ git clone git@github.com:Alexis-acad/CPE232_De
niega.git
```



```
programmymain@workstation:~$ git clone git@github.com:Alexis-acad/CPE232_De
niega.git
Cloning into 'CPE232_Deniega'...
The authenticity of host 'github.com (4.237.22.38)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvCO
qU.
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 host
s.
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
programmymain@workstation:~$
```

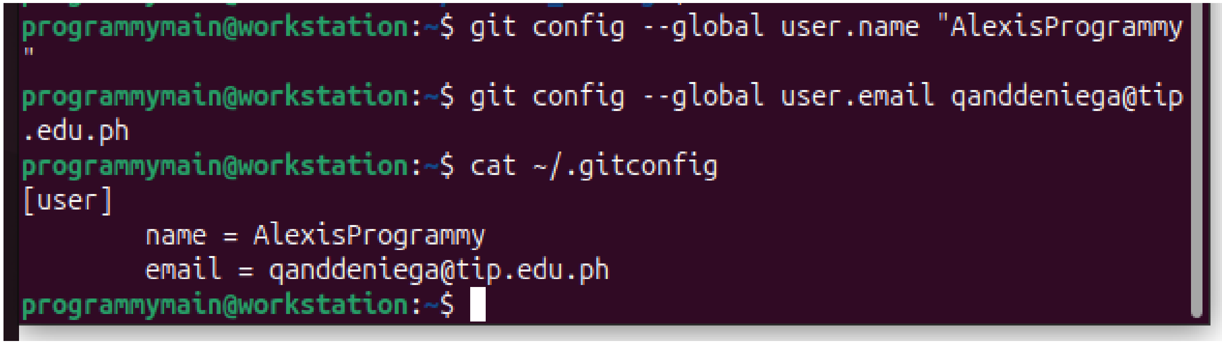
f. To verify that you have cloned the GitHub repository, issue the command *ls*. Observe that you have the CPE232_yourname in the list of your directories. Use CD command to go to that directory and LS command to see the file README.md.



```
Receiving objects: 100% (3/3), done.
programmymain@workstation:~$ ls
CPE232_Deniega  Documents  Music     Public  Templates
Desktop         Downloads  Pictures  snap    Videos
programmymain@workstation:~$ cd CPE232_Deniega
programmymain@workstation:~/CPE232_Deniega$ ls
README.md
programmymain@workstation:~/CPE232_Deniega$
```

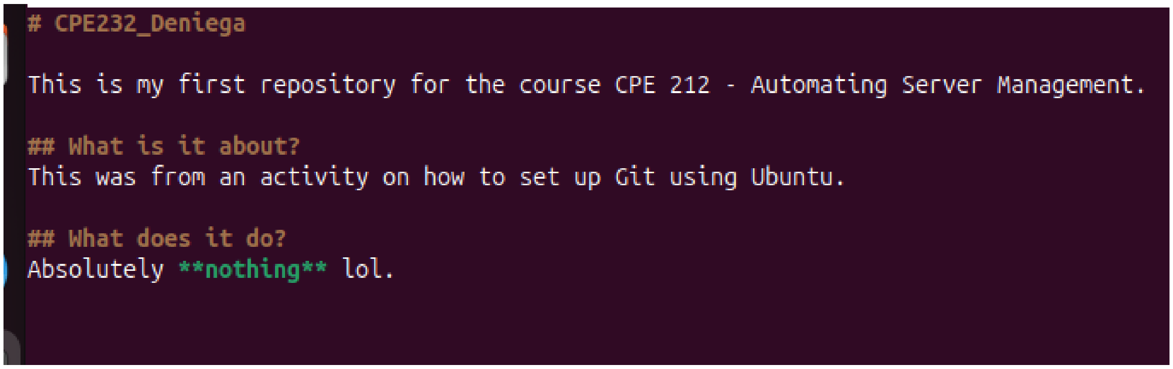
g. Use the following commands to personalize your git.

- *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*



```
programmymain@workstation:~$ git config --global user.name "AlexisProgrammy
"
programmymain@workstation:~$ git config --global user.email qanddeniega@tip
.edu.ph
programmymain@workstation:~$ cat ~/.gitconfig
[user]
        name = AlexisProgrammy
        email = qanddeniega@tip.edu.ph
programmymain@workstation:~$ 
```

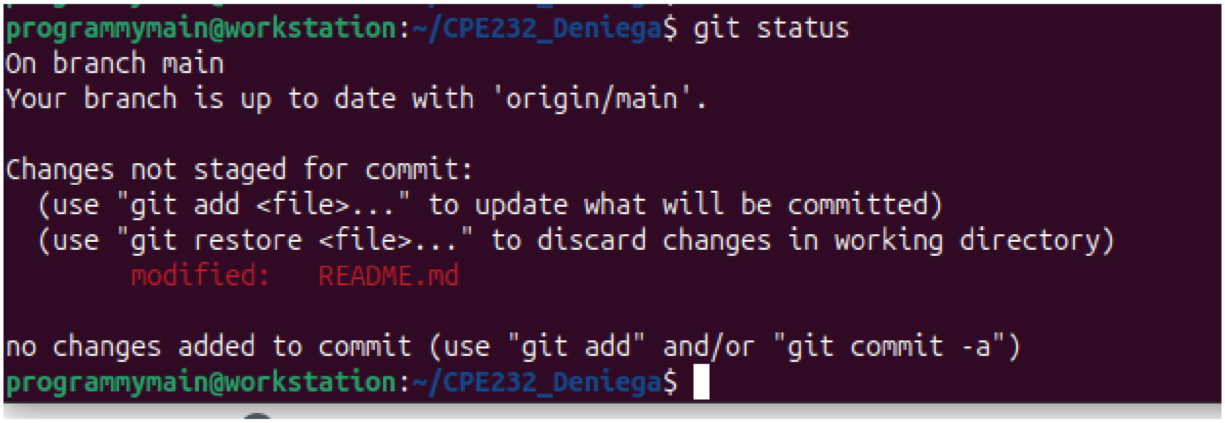
h. Edit the README.md file using nano command. Provide any information on the markdown file pertaining to the repository you created. Make sure to write out or save the file and exit.



```
# CPE232_Deniega

This is my first repository for the course CPE 212 - Automating Server Management.

## What is it about?
This was from an activity on how to set up Git using Ubuntu.

## What does it do?
Absolutely **nothing** lol.
```

I know markdown formatting because I use Obsidian (markdown editor app) to do notes lol.

i. 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? ***A notice appears: it says that the README file is modified.***

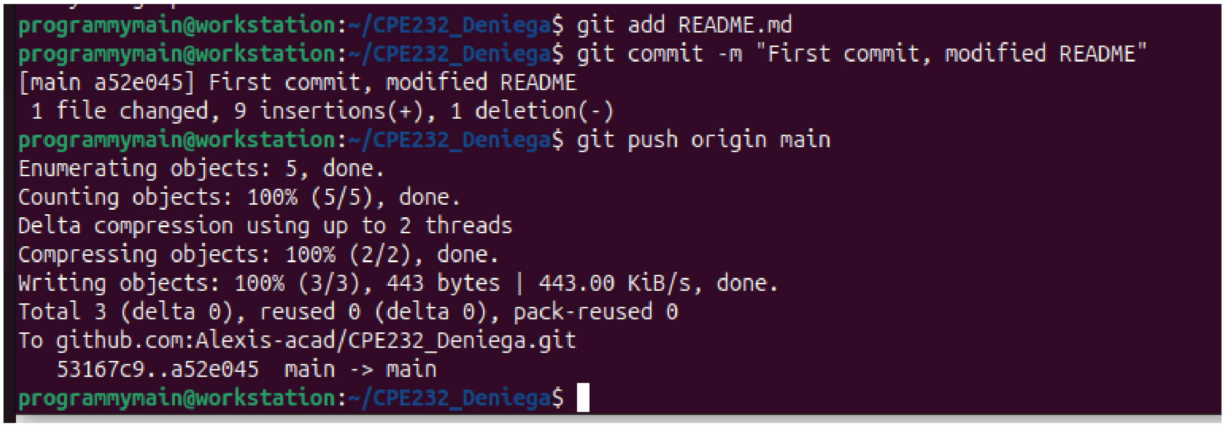


```
programmymain@workstation:~/CPE232_Deniega$ git status
On branch main
Your branch is up to date with 'origin/main'.

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")
programmymain@workstation:~/CPE232_Deniega$ 
```

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

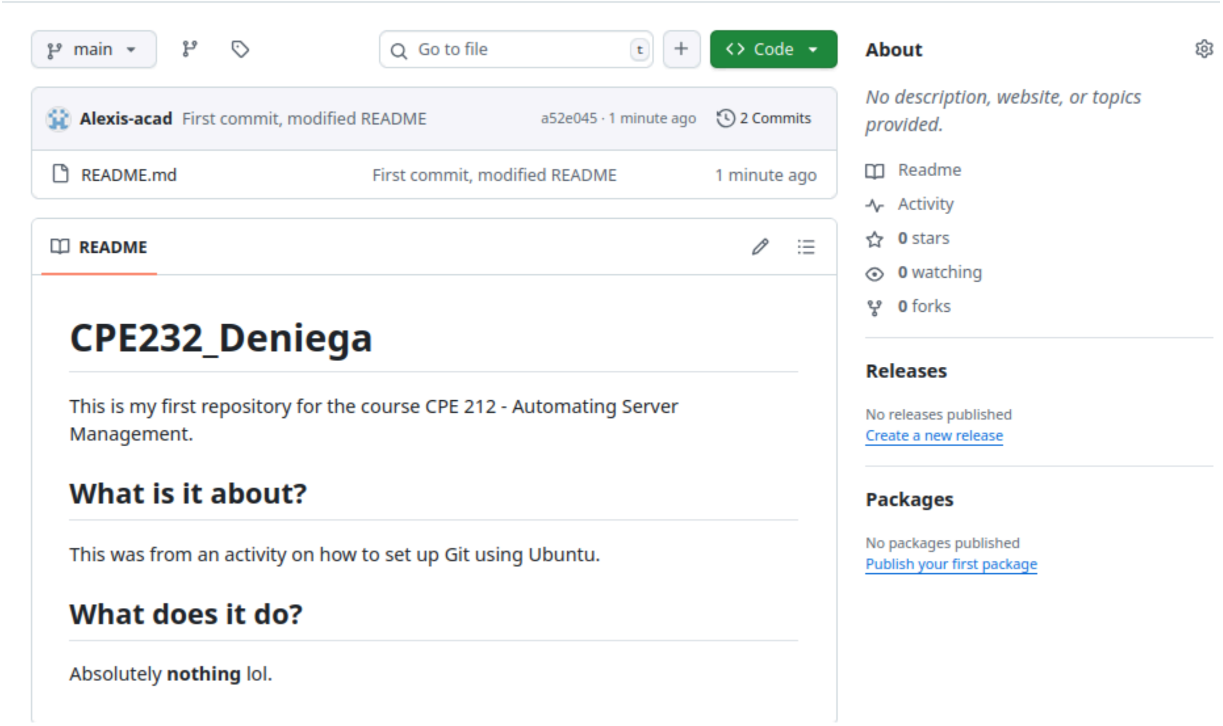
k. Use the *git commit -m “your message”* to create a snapshot of the staged changes along the timeline of the Git projects history. The use of this command is required to select the changes that will be staged for the next commit.

l. 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.*



```
programmymain@workstation:~/CPE232_Deniega$ git add README.md
programmymain@workstation:~/CPE232_Deniega$ git commit -m "First commit, modified README"
[main a52e045] First commit, modified README
 1 file changed, 9 insertions(+), 1 deletion(-)
programmymain@workstation:~/CPE232_Deniega$ git push origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 2 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 443 bytes | 443.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:Alexis-acad/CPE232_Deniega.git
   53167c9..a52e045  main -> main
programmymain@workstation:~/CPE232_Deniega$ 
```

m. 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 the 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.



Alexis-acad First commit, modified README a52e045 · 1 minute ago 2 Commits

README.md First commit, modified README 1 minute ago

No description, website, or topics provided.

No releases published

No packages published

CPE232_Deniega

This is my first repository for the course CPE 212 - Automating Server Management.

What is it about?

This was from an activity on how to set up Git using Ubuntu.

What does it do?

Absolutely **nothing** lol.

Reflections:

Answer the following:

3. What sort of things have we so far done to the remote servers using ansible commands?

I have done the following: added the public key file of the workstation to the two remote servers, and remembered their IP-hostname, made changes to the newly-made repository (README markdown), then used commit and made a message. After that pushed it, which got put to the GitHub files.

4. How important is the inventory file?

The inventory file is basically a storage of important information that is used. It can be used in anything that has essential information, but in the context of git, it can be used for version control and documentation.

Conclusions/Learnings:

In this activity, I learned about how to generate and add SSH keys and copy them to remote servers. This allowed for both machines to communicate and login without a password. I also learned about the basics of git and GitHub, adding my first repository, and to commit, and push a file from the local repository to the GitHub repo.

This will make it more accessible to the public, Github and git allows for a seamless experience transferring files between teams or strangers.