

Name: Buduan, Christian Aaron C.	Date Performed: 9/4/2024
Course/Section: CpE31S2	Date Submitted: 9/11/2024
Instructor: Engr. Robin Valenzuela	Semester and SY: 1st sem 2024-2025
Activity 2: SSH Key-Based Authentication and Setting up Git	
1. Objectives: <ul style="list-style-type: none"> 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 <p>It is assumed that you are already done with the last Activity (Activity 1: Configure Network using Virtual Machines). <i>Provide screenshots for each task.</i></p> <p>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.</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 password stored in files and eliminated the possibility of a compromised server stealing the user's password.</p> <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.</p>	
Task 1: Create an SSH Key Pair for User Authentication <ul style="list-style-type: none"> 1. The simplest way to generate a key pair is to run <i>ssh-keygen</i> 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`.

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.
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`.

```
qcacbuduan@Workstation:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/qcacbuduan/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):

qcacbuduan@Workstation:~$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/qcacbuduan/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/qcacbuduan/.ssh/id_rsa
Your public key has been saved in /home/qcacbuduan/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:fgr4sSn5v3vYmIJMpoVq+nlsoxfMJJaPcwlRQ7FmBoyA qcacbuduan@Workstation
The key's randomart image is:
+---[RSA 4096]-----+
|      .               |
|E   o .               |
|.. + . o              |
|  o o = =             |
|   * O XS             |
|  . @.*.o             |
|   =.*oo.o.           |
|  . =.B=.+o           |
|   .*=+B=             |
+----[SHA256]-----+
qcacbuduan@Workstation:~$ ls -la .ssh
total 24
drwx-----  2 qcacbuduan qcacbuduan 4096 Sep  6 19:31 .
drwxr-x--- 18 qcacbuduan qcacbuduan 4096 Apr 26 09:48 ..
-rw-----  1 qcacbuduan qcacbuduan 3389 Sep  6 19:31 id_rsa
-rw-r--r--  1 qcacbuduan qcacbuduan  748 Sep  6 19:31 id_rsa.pub
-rw-----  1 qcacbuduan qcacbuduan 2240 Aug 25 23:28 known_hosts
-rw-----  1 qcacbuduan qcacbuduan 1120 Aug 25 23:23 known_hosts.old
qcacbuduan@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*
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?

```
qcacbuduan@Workstation:~$ ssh-copy-id -i ~/.ssh/id_rsa qcacbuduan@server1
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/qcacbuduan/.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 prompted now it is to install
the new keys
qcacbuduan@server1's password:

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh 'qcacbuduan@server1'"
and check to make sure that only the key(s) you wanted were added.

qcacbuduan@Workstation:~$ ssh-copy-id -i ~/.ssh/id_rsa qcacbuduan@server2
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/qcacbuduan/.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 prompted now it is to install
the new keys
qcacbuduan@server2's password:

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh 'qcacbuduan@server2'"
and check to make sure that only the key(s) you wanted were added.
```

```
qcacbuduan@Workstation:~$ ssh qcacbuduan@server1
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 6.5.0-28-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.

15 updates can be applied immediately.
11 of these updates are standard security updates.
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

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or
proxy settings

147 updates could not be installed automatically. For more details,
see /var/log/unattended-upgrades/unattended-upgrades.log
Last login: Sun Aug 25 23:27:32 2024 from 192.168.56.102
qcacbuduan@server1:~$ exit
logout
Connection to server1 closed.
```

```
qcacbuduan@Workstation:~$ ssh qcacbuduan@server2
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 6.5.0-18-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.

15 updates can be applied immediately.
11 of these updates are standard security updates.
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

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Last login: Sun Aug 25 23:28:36 2024 from 192.168.56.102
qcacbuduan@server2:~$
```

Reflections:

Answer the following:

1. How will you describe the ssh-program? What does it do?
 - The SSH Protocol is a cryptographic network protocol that allows users such as system administrators, to securely access a computer across an unprotected network. Its most prominent applications are remote login and command-line execution.
2. How do you know that you already installed the public key to the remote servers?
 - You can determine if you have the public key in the remote server if it doesn't require you to enter the password when entering.

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*

```
qcacbuduan@Workstation:~$ sudo apt upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following packages have been kept back:
  linux-generic-hwe-22.04 linux-headers-generic-hwe-22.04 linux-image-generic-hwe-22.04
  python3-update-manager shim-signed update-manager update-manager-core
The following packages will be upgraded:
  apparmor libapparmor1 libfprint-2-2 libssl3 openssl ubuntu-advantage-tools ubuntu-pro-client
  ubuntu-pro-client-l10n vim-common vim-tiny xxd
11 upgraded, 0 newly installed, 0 to remove and 7 not upgraded.
7 standard LTS security updates
Need to get 5,105 kB of archives.
After this operation, 39.9 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

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

```
qcacbuduan@Workstation:~$ which git
/usr/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.

```
qcacbuduan@Workstation:~$ which git
/usr/bin/git
```

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 CPE232_yourname. Check Add a README file and click Create repository.


https://github.com/new

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (*).

Owner *

 buduman

Repository name *

CPE212_Buduan

✓ CPE212_Buduan is available.

Great repository names are short and memorable. Need inspiration? How about [cuddly-waffle](#) ?

Description (optional)

☒  **Public**

Anyone on the internet can see this repository. You choose who can commit.

☐  **Private**

You choose who can see and commit to this repository.

Initialize this repository with:

☒ **Add a README file**

This is where you can write a long description for your project. [Learn more about READMEs.](#)


- 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.
- 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.




buduman (buduman)


Your personal account

[Go to your personal profile](#)

 **Public profile**


 Account

 Appearance


 Accessibility

 Notifications

Access


 Billing and plans


 Emails

 Password and authentication

 Sessions

 SSH and GPG keys

 Organizations

 Enterprises

 Moderation

Add new SSH Key

Title

CPE212

Key type

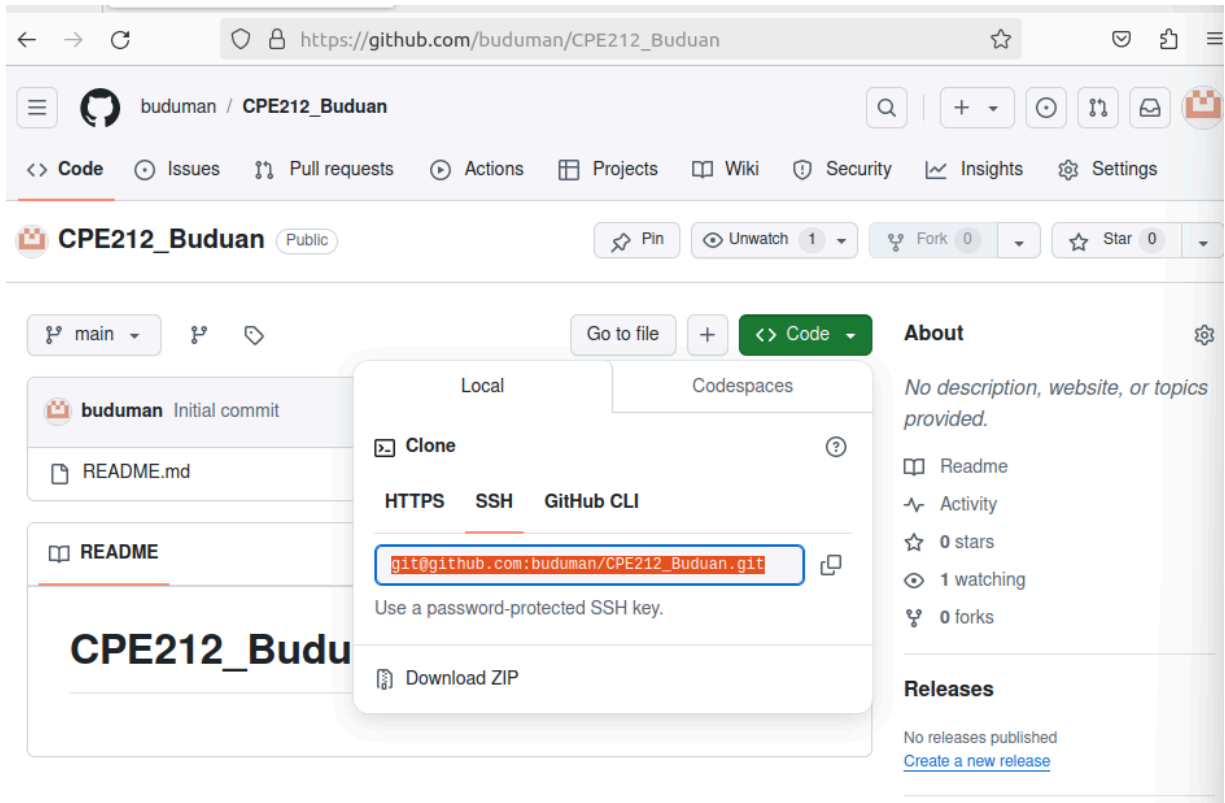
Authentication Key

Key

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQChRUM9imvW9FBac78vXDD1HVQ8wj1I5vEadG
R1JXFo1DLUyE9IP+iFva7Tn4kAxPHJ3x82dh/H6zqPogFohBlcxp/v/
hHve03orpUE+6PQA1YRbSaKox09bSpnEqQVvnu2nOytBYQJl8x9674+EL2rLTBI+A3EdJ872/
0lJShQWmLPW7iU6kJ3uvhFicRNzbKrkbrWmB2QTtwk6OzL80X7BT03Fo2yRhL2JJJAz5cbw
NNse46vV8ZLG7Eq9sIDhJxmawym+BgCeal2KHPu5uqQ2DhihT90xbglzAW/
pLlxzlyFHxbIOi2233IL44kyyQGOuWZKT2N2x6eRtu/
Ewg6M9+e6K9OfEmBBksWuLbpswL5U2GNxZdLsS+BPQKUF3C59Fw2nhYkkgRXh+55PPst
dtAus91S9r9k8crFdX+k7Lpp4OVsNyWW8Vkv+uSQK/
```

Add SSH key

- 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.



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

```
qcacbuduan@Workstation:~$ git clone git@github.com:buduman/CPE212_Buduan.git
Cloning into 'CPE212_Buduan'...
The authenticity of host 'github.com (20.205.243.166)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvV6TuJJhpZisF/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: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
```

- 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`.

```

qcacbuduan@Workstation:~$ ls
CPE212_Buduan  Documents  ipv6config.sh  Music      Public  Templates
Desktop        Downloads  IPv6config.sh  Pictures   snap    Videos
qcacbuduan@Workstation:~$ cd CPE212_Buduan
qcacbuduan@Workstation:~/CPE212_Buduan$ ls
README.md

```

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`

```

qcacbuduan@Workstation:~/CPE212_Buduan$ cat ~/.gitconfig
cat: /home/qcacbuduan/.gitconfig: No such file or directory
qcacbuduan@Workstation:~/CPE212_Buduan$ git config --global user.name "Buduan"
qcacbuduan@Workstation:~/CPE212_Buduan$ git config --global user.email qcacbuduan@tip.edu.ph
qcacbuduan@Workstation:~/CPE212_Buduan$ cat ~/.gitconfig
[user]
    name = Buduan
    email = qcacbuduan@tip.edu.ph

```

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.

```

GNU nano 6.2                                README.md *
# CPE212_Buduan
CPE212 key

```

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?

```

qcacbuduan@Workstation:~/CPE212_Buduan$ 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")

```

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`.
- 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.

```
qcacbuduan@Workstation:~/CPE212_Buduan$ git add README.md
qcacbuduan@Workstation:~/CPE212_Buduan$ git commit -m "authorized user only"
[main a9e35b5] authorized user only
 1 file changed, 2 insertions(+), 1 deletion(-)
qcacbuduan@Workstation:~/CPE212_Buduan$ git push origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Writing objects: 100% (3/3), 268 bytes | 268.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:buduman/CPE212_Buduan.git
 e192025..a9e35b5  main -> main
```

  main **CPE212_Buduan / README.md** 

 **buduman** authorized user only

2 lines (2 loc) · 27 Bytes

Preview Code Blame

CPE212_Buduan

CPE212 key

Reflections:

Answer the following:

3. What sort of things have we so far done to the remote servers using ansible commands?
4. How important is the inventory file?

Conclusions/Learnings:

During the course of following the directions and finishing the tasks, I was able to learn and demonstrate how to construct an SSH key pair for user authentication, how to copy the public key to servers, install git, and most crucially, establish a GitHub repository and clone it to my local PC to obtain a local repository that will serve as a link to the main GitHub repository. In the end, I am able to upload or transfer the changes and commits I made in the local repository to the GitHub repository.