Guided Practice: Install, Configure, and Test Ansible

Outcome

In this Guided Practice, you will install and configure the Ansible on your CentOS computer and use the CentOS computer to run an Ansible test.

Resources Needed

- VCASTLE Pod configured for the class. For this Guided Practice, we use the CentOS 8 and Ubuntu 20.04 LTS machines.
- Your user needs to be able to elevate their privileges with sudo.

Level of Difficulty

Moderate

Deliverables

Deliverables are marked in red font or with a red picture border around the screenshot. Additionally, there are questions at the end. **Your username or studentid should be visible in all screenshots that you submit.**

General Considerations

You should be familiar with configuring Linux networking. ssh should already be configured on your Ubuntu computer from a prior Guided Practice.

Install Ansible on CentOS

You'll need to download and install Ansible. Since the CentOS computer will be our "Ansible workstation," the consideration is how to install Ansible. One way is to use pip.

1. Before beginning, you need to add a user, and add your new user to the **wheel** group in **CentOS.**

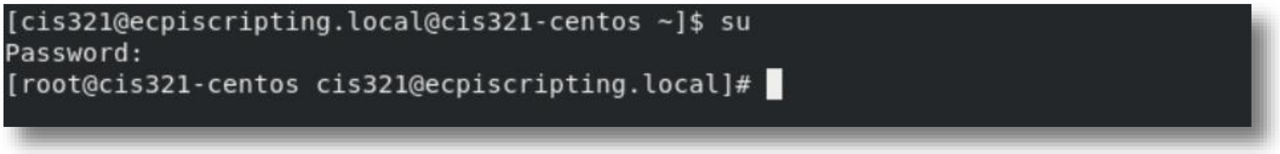
   a. Log onto the CentOS computer using the CIS321 login.
   b. Open a terminal. You're going to make changes as root.
   c. Type

   ```
   su
   ```

   d. When prompted for a password, type the following:

   ```
   Password1
   ```

   Your screen should resemble this:



```
[cis321@ecpiscripting.local@cis321-centos ~]$ su
Password:
[root@cis321-centos cis321@ecpiscripting.local]# █
```

   e. Type the following (substituting your studentID for ranbel1234).

   ```
   sudo useradd ranbel1234
   ```

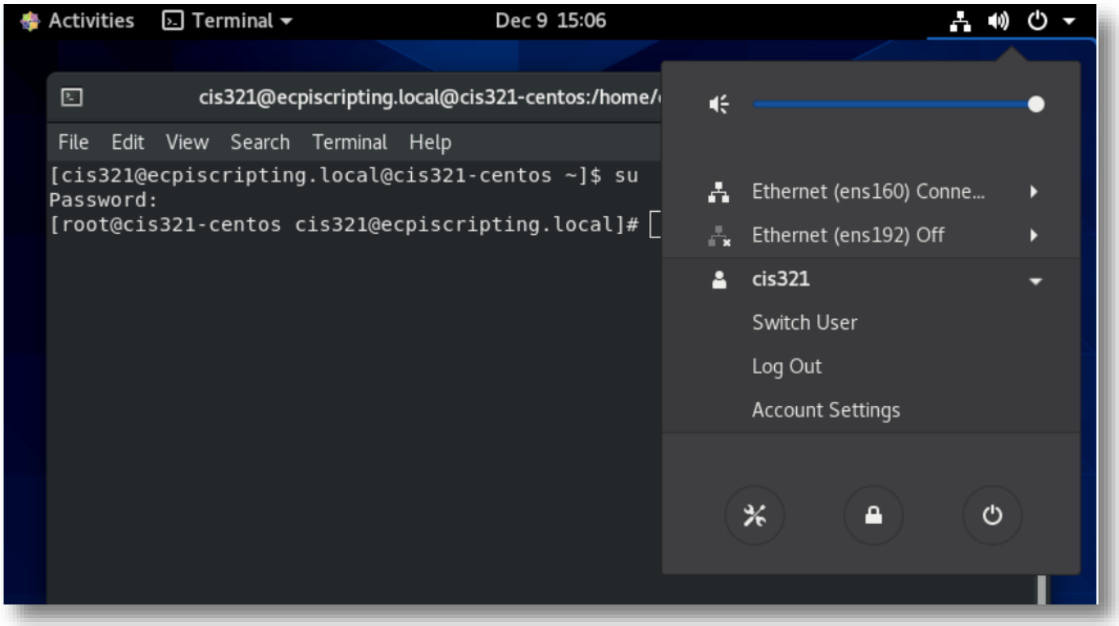
   f. Then, give your user a password. Type the following (substituting your studentID for ranbel1234).

   ```
   sudo passwd ranbel1234
   ```

   g. Now, add your user to the **wheel** group. Type the following (substituting your user name):

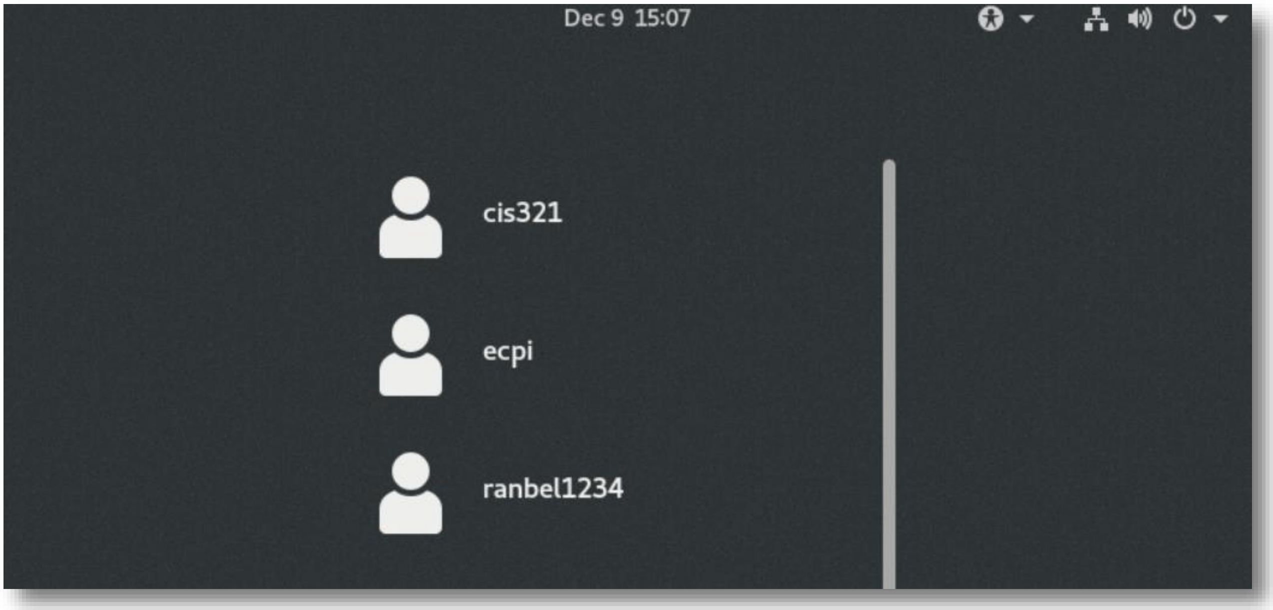
   ```
   sudo usermod –aG wheel ranbel1234
   ```

   h. In the upper right of your screen, you'll see a down arrow. Click it, and select **Switch User**.


Activities
Terminal
Dec 9 15:06
cis321@ecpiscripting.local@cis321-centos:/home/
File Edit View Search Terminal Help
Ethernet (ens160) Conne...
Ethernet (ens192) Off
cis321
Switch User
Log Out
Account Settings

```
[cis321@ecpiscripting.local@cis321-centos ~]$ su
Password:
[root@cis321-centos cis321@ecpiscripting.local]#
```

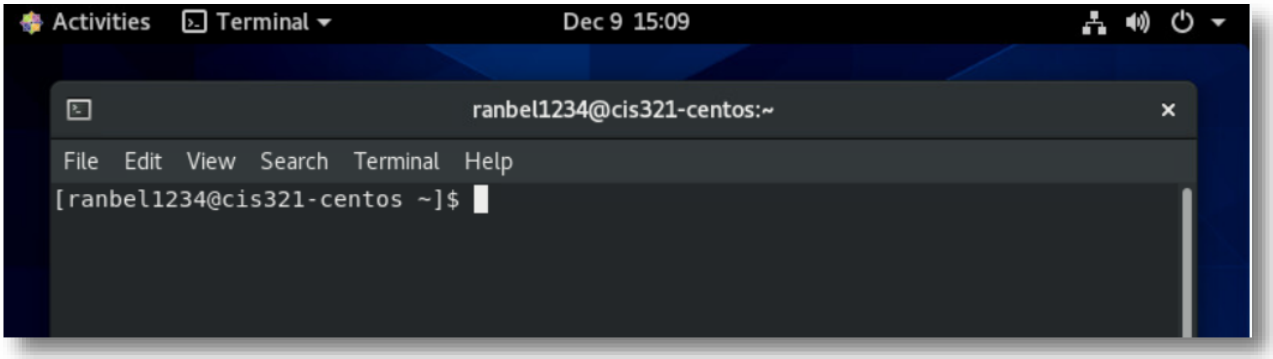
i. Then, select the user you just added. In this case, that's ranbel1234.



j. You'll receive a password prompt.

Now, you should be logged on as the user you created at the beginning of this Guided Practice (your studentID). You may be prompted with start-up introduction for operating system. Just hit **next** and **skip** to remove or click away to continue.

k. Open a terminal, and it should resemble this:

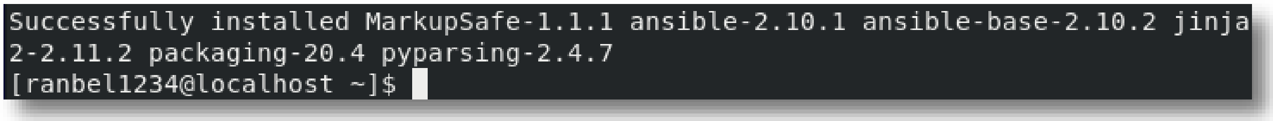


```
[ranbel1234@cis321-centos ~]$
```

2. Type:

```
sudo pip3 install ansible
```

After installation, your output should resemble this:

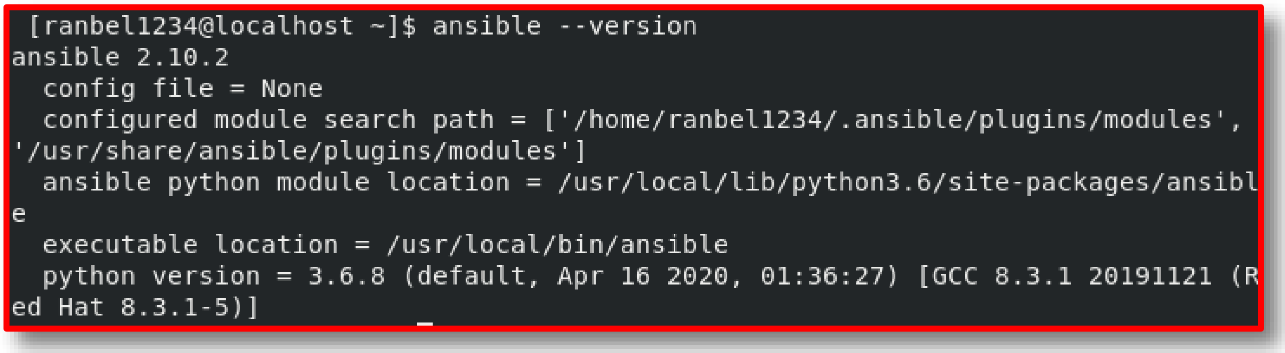


```
Successfully installed MarkupSafe-1.1.1 ansible-2.10.1 ansible-base-2.10.2 jinja
2-2.11.2 packaging-20.4 pyparsing-2.4.7
[ranbel1234@localhost ~]$
```

3. Let’s check the version of Ansible that you’ve installed. Type what you see below, and note that there are two dashes before version:

```
ansible --version
```

In this case, we have version 2.10.2.



```
 [ranbel1234@localhost ~]$ ansible --version
ansible 2.10.2
  config file = None
  configured module search path = ['/home/ranbel1234/.ansible/plugins/modules',
'/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/local/lib/python3.6/site-packages/ansibl
e
  executable location = /usr/local/bin/ansible
  python version = 3.6.8 (default, Apr 16 2020, 01:36:27) [GCC 8.3.1 20191121 (R
ed Hat 8.3.1-5)]
```

Take a screenshot like the one above, and paste it into your Lab Report.

4. Now, we need to configure a file to tell Ansible the ip addresses of the remote computers we want to contact. This is done by editing the hosts file. **Note**: This is not the /etc/hosts file, which has another purpose. This file is /etc/ansible/hosts. You need to create the directory, create the hosts file, and enter data into it.

   a. Create the ansible directory by typing:

```
sudo mkdir /etc/ansible
```

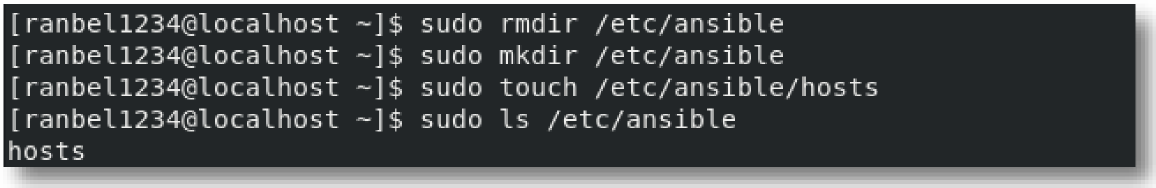
b. Then, create the hosts file by typing:

```
sudo touch /etc/ansible/hosts
```

c. Then, check the directory by typing:

```
sudo ls /etc/ansible
```

Your output should resemble this:



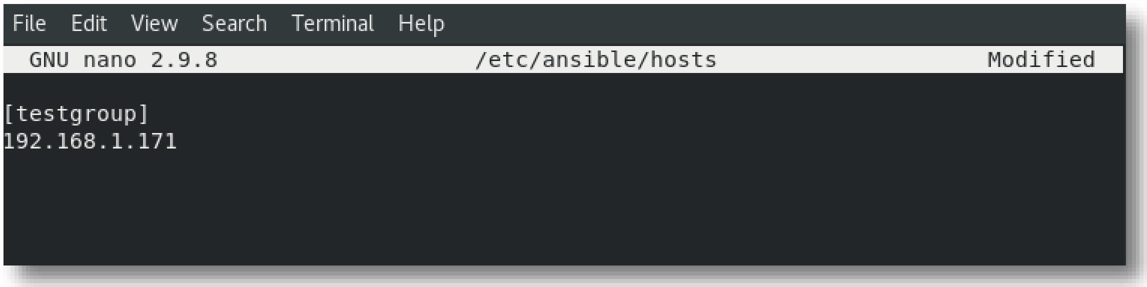
```
[ranbel1234@localhost ~]$ sudo rmdir /etc/ansible
[ranbel1234@localhost ~]$ sudo mkdir /etc/ansible
[ranbel1234@localhost ~]$ sudo touch /etc/ansible/hosts
[ranbel1234@localhost ~]$ sudo ls /etc/ansible
hosts
```

d. Now, we can open the hosts file, and make our entry. Type:

```
sudo nano /etc/ansible/hosts
```

e. We create a group of one and enter the ip address of the Ubuntu computer. We'll call our group **testgroup**. Your file should look like the one you see in the image below. To save the file and exit nano, you'll need to type **Ctrl+O** to save the file (and confirm), and **Ctrl+X** to exit nano.

NOTE: You should confirm that the IP address you enter is the IP address of your Ubuntu server, which **may not be** the one listed below. In some VCASTLE pods, it may be 192.168.1.3.

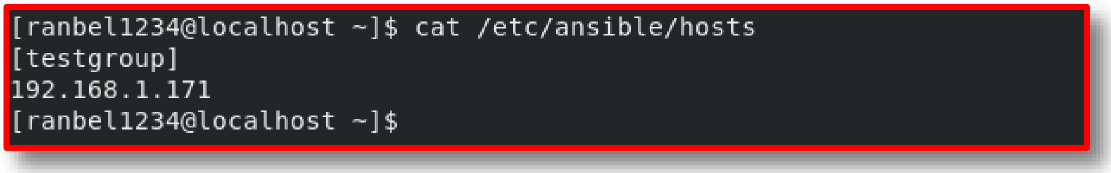


```
  GNU nano 2.9.8                    /etc/ansible/hosts                    Modified

[testgroup]
192.168.1.171
```

f. Now let's check the file's contents. Type:

```
cat /etc/ansible/hosts
```

Your output should look like this:



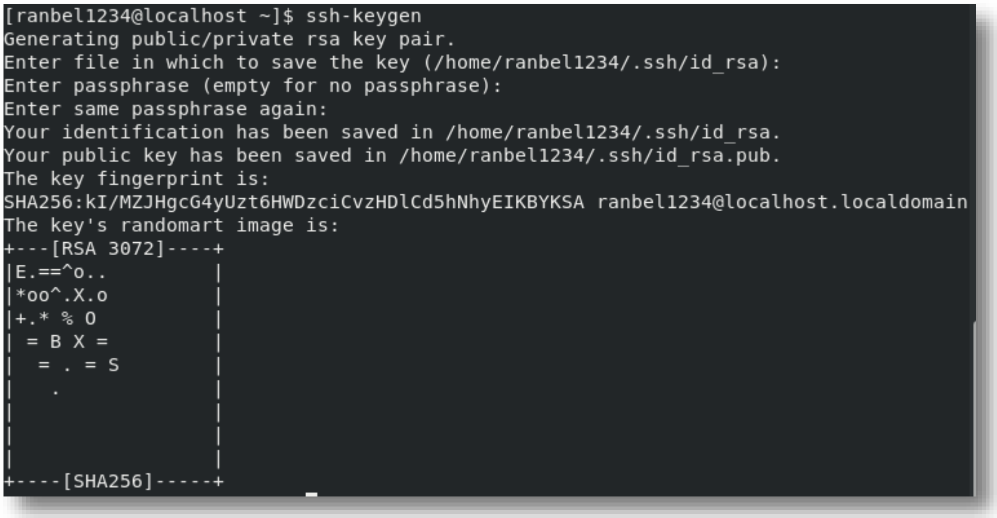
```
[ranbel1234@localhost ~]$ cat /etc/ansible/hosts
[testgroup]
192.168.1.171
[ranbel1234@localhost ~]$
```

Take a screenshot like the one above, and paste it into your Lab Report.

5. Configure SSH Key-Based Authentication on the CentOS Ansible Workstation.
   a. First, run

```
ssh-keygen
```

   b. To accept all the defaults, simply press **Enter**.
   c. When prompted for a keyphrase, simply press **Enter**.



```
[ranbel1234@localhost ~]$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ranbel1234/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ranbel1234/.ssh/id_rsa.
Your public key has been saved in /home/ranbel1234/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:kI/MZJHgcG4yUzt6HWDzciCvzHDlCd5hNhyEIKBYKSA ranbel1234@localhost.localdomain
The key's randomart image is:
+---[RSA 3072]----+
|E.==^o..         |
|*oo^.X.o         |
|+.* % O          |
| = B X =         |
|  = . = S        |
|    .            |
|                 |
|                 |
|                 |
+----[SHA256]-----+
```

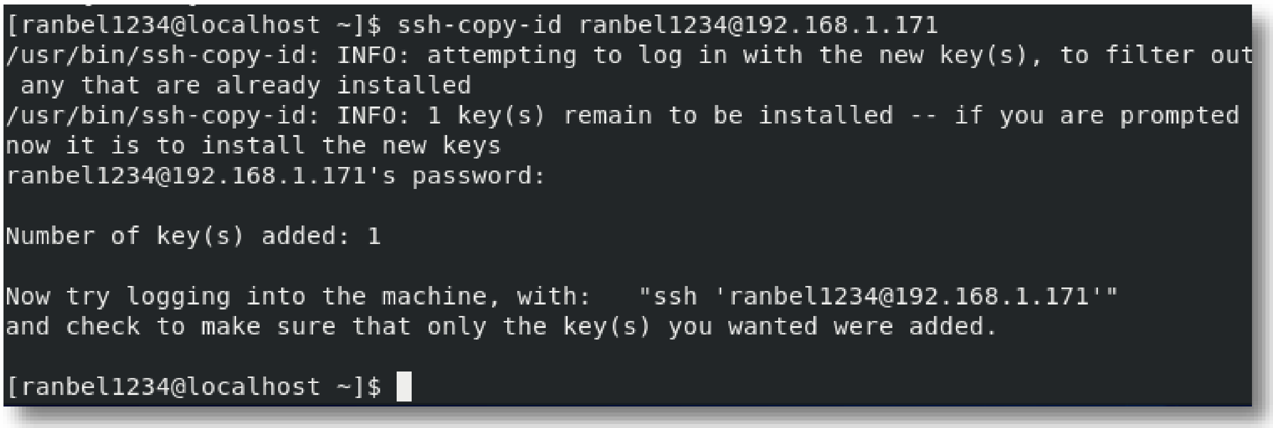
6. Copy the public key to your Ubuntu computer using ssh, which you have already configured on Ubuntu. If you have forgotten the IP Address of Ubuntu, run **ip addr show** on the Ubuntu computer. (**NOTE**: If, throughout this step, you are asked to confirm, respond **Yes**.)

```
ssh-copy-id username@IPAddressOfUbuntu
```

In this case we use:

```
ssh-copy-id ranbel1234@192.168.1.171
```

Remember, you need to enter the password of the user of the remote computer (the Ubuntu computer). Your output should resemble this:

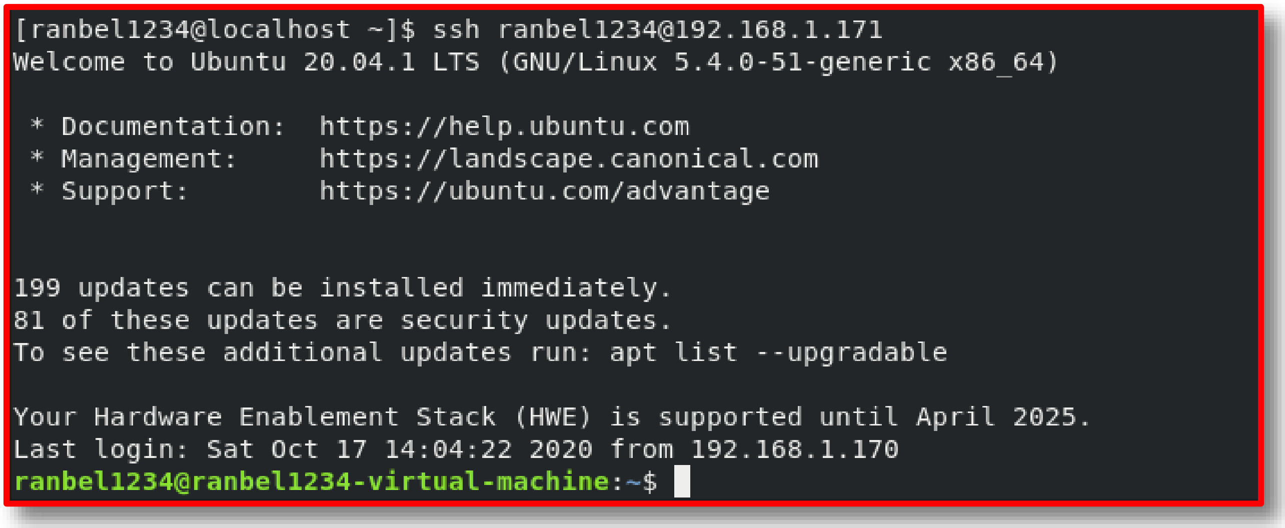


```
[ranbel1234@localhost ~]$ ssh-copy-id ranbel1234@192.168.1.171
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out
 any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted
now it is to install the new keys
ranbel1234@192.168.1.171's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'ranbel1234@192.168.1.171'"
and check to make sure that only the key(s) you wanted were added.

[ranbel1234@localhost ~]$ 
```

You can test whether you can ssh into the remote computer without a password by typing

```
ssh username@IPAddressOfUbuntu
```

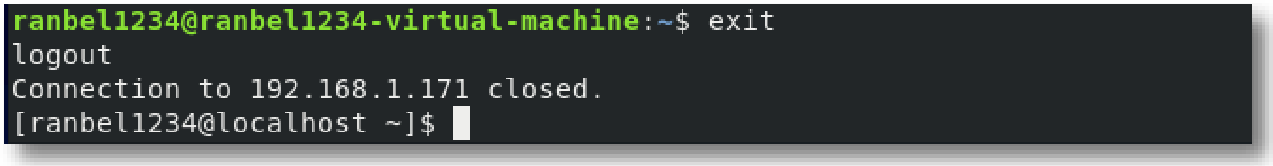
If successful, your output should resemble this:



```
[ranbel1234@localhost ~]$ ssh ranbel1234@192.168.1.171
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-51-generic x86_64)

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


199 updates can be installed immediately.
81 of these updates are security updates.
To see these additional updates run: apt list --upgradable

Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Sat Oct 17 14:04:22 2020 from 192.168.1.170
ranbel1234@ranbel1234-virtual-machine:~$ 
```

Take a screenshot like the one above, and paste it into your Lab Report.

Type the following to close the ssh connection:
exit



```
ranbel1234@ranbel1234-virtual-machine:~$ exit
logout
Connection to 192.168.1.171 closed.
[ranbel1234@localhost ~]$ 
```

Test Ansible

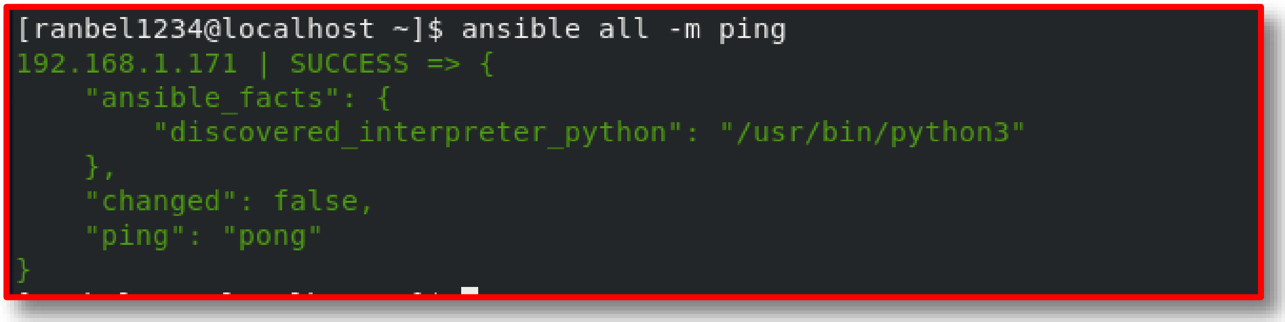
We've installed ssh on the remote machine, installed and configured Ansible on CentOS, generated a Public Key on CentOS, and copied the Public Key to Ubuntu. Now, we can test Ansible. We'll use a simple

Ansible command (not a playbook yet), which will read the **/etc/ansible/hosts** file, and ping the remote computer.

7. Type the following:

```
ansible all -m ping
```

Your output should resemble this:



```
[ranbel1234@localhost ~]$ ansible all -m ping
192.168.1.171 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
```

Take a screenshot like the one above, and paste it into your Lab Report.

What's going on with the command? We're calling Ansible, telling it to run against all computers in the **/etc/ansible/hosts** file and execute a module (-m) called ping. **Note**: This is NOT ICMP ping; this is just a trivial test module that requires Python on the remote-node.

Guided Practice Questions

In your **Guided Practice Lab Report**, answer the following questions about this learning activity. Some may require research.

1. Explain the content of the **/etc/ansible/hosts** file.
2. What was the purpose of generating a Public Key and copying it onto the remote computer?
3. What is an Ansible module?