

<b>Name: Magboo, Matt Clemence C.</b>	<b>Date Performed: 08/25/2025</b>
<b>Course/Section: CPE212 - CPE32S2</b>	<b>Date Submitted: 08/25/2025</b>
<b>Instructor: Engr. Robin Valenzuela</b>	<b>Semester and SY: 2025-2026</b>
<b>Activity 4: Running Elevated Ad hoc Commands</b>	
<b>1. Objectives:</b> 1.1 Use commands that makes changes to remote machines 1.2 Use playbook in automating ansible commands	
<b>2. Discussion:</b>  <i>Provide screenshots for each task.</i>  <b>Elevated Ad hoc commands</b> So far, we have not performed ansible commands that makes changes to the remote servers. We manage to gather facts and connect to the remote machines, but we still did not make changes on those machines. In this activity, we will learn to use commands that would install, update, and upgrade packages in the remote machines. We will also create a playbook that will be used for automations.  <b>Playbooks</b> record and execute <b>Ansible's</b> configuration, deployment, and orchestration functions. They can describe a policy you want your remote systems to enforce, or a set of steps in a general IT process. If Ansible modules are the tools in your workshop, playbooks are your instruction manuals, and your inventory of hosts are your raw material. At a basic level, playbooks can be used to manage configurations of and deployments to remote machines. At a more advanced level, they can sequence multi-tier rollouts involving rolling updates, and can delegate actions to other hosts, interacting with monitoring servers and load balancers along the way. You can check this documentation if you want to learn more about playbooks. <a href="#">Working with playbooks — Ansible Documentation</a>	
<b>Task 1: Run elevated ad hoc commands</b>  1. Locally, we use the command <i>sudo apt update</i> when we want to download package information from all configured resources. The sources often defined in <i>/etc/apt/sources.list</i> file and other files located in <i>/etc/apt/sources.list.d/</i> directory. So, when you run update command, it downloads the package information from the Internet. It is useful to get info on an updated version of packages or their dependencies. We can only run	

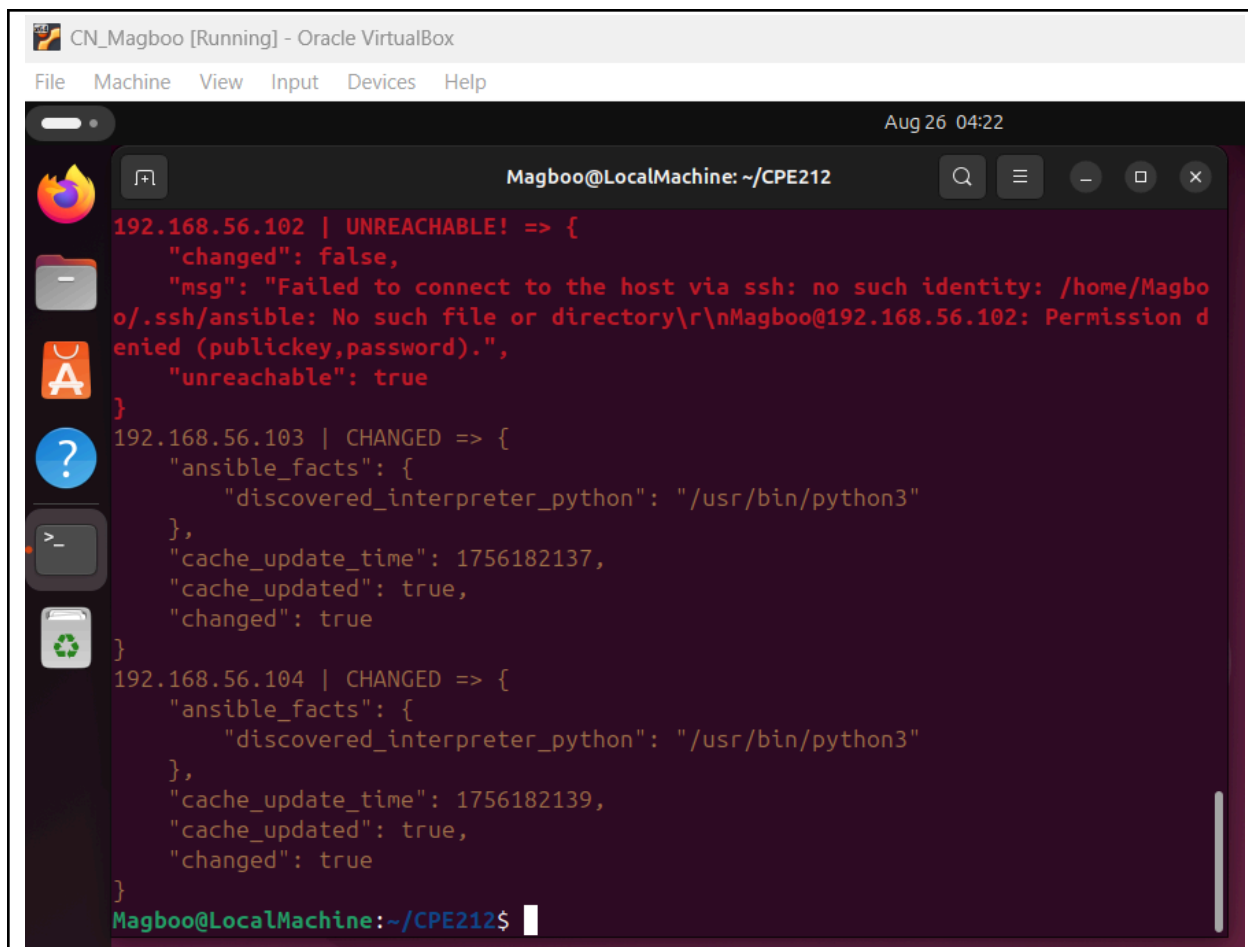
an apt update command in a remote machine. Issue the following command:

*ansible all -m apt -a update\_cache=true*

What is the result of the command? Is it successful? **No**

```
CN_Magboo [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Aug 26 04:21
Magboo@LocalMachine: ~/CPE212
Magboo@LocalMachine:~/CPE212$ ansible all -m apt -a update_cache=true
192.168.56.102 | UNREACHABLE! => {
  "changed": false,
  "msg": "Failed to connect to the host via ssh: no such identity: /home/Magboo/.ssh/ansible: No such file or directory\r\nMagboo@192.168.56.102: Permission denied (publickey,password).",
  "unreachable": true
}
192.168.56.103 | UNREACHABLE! => {
  "changed": false,
  "msg": "Failed to connect to the host via ssh: Connection timed out during banner exchange\r\nConnection to 192.168.56.103 port 22 timed out",
  "unreachable": true
}
192.168.56.104 | FAILED! => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "msg": "Failed to lock apt for exclusive operation: Failed to lock directory /var/lib/apt/lists/: E:Could not open lock file /var/lib/apt/lists/lock - open (13: Permission denied)"
}
Magboo@LocalMachine:~/CPE212$
```

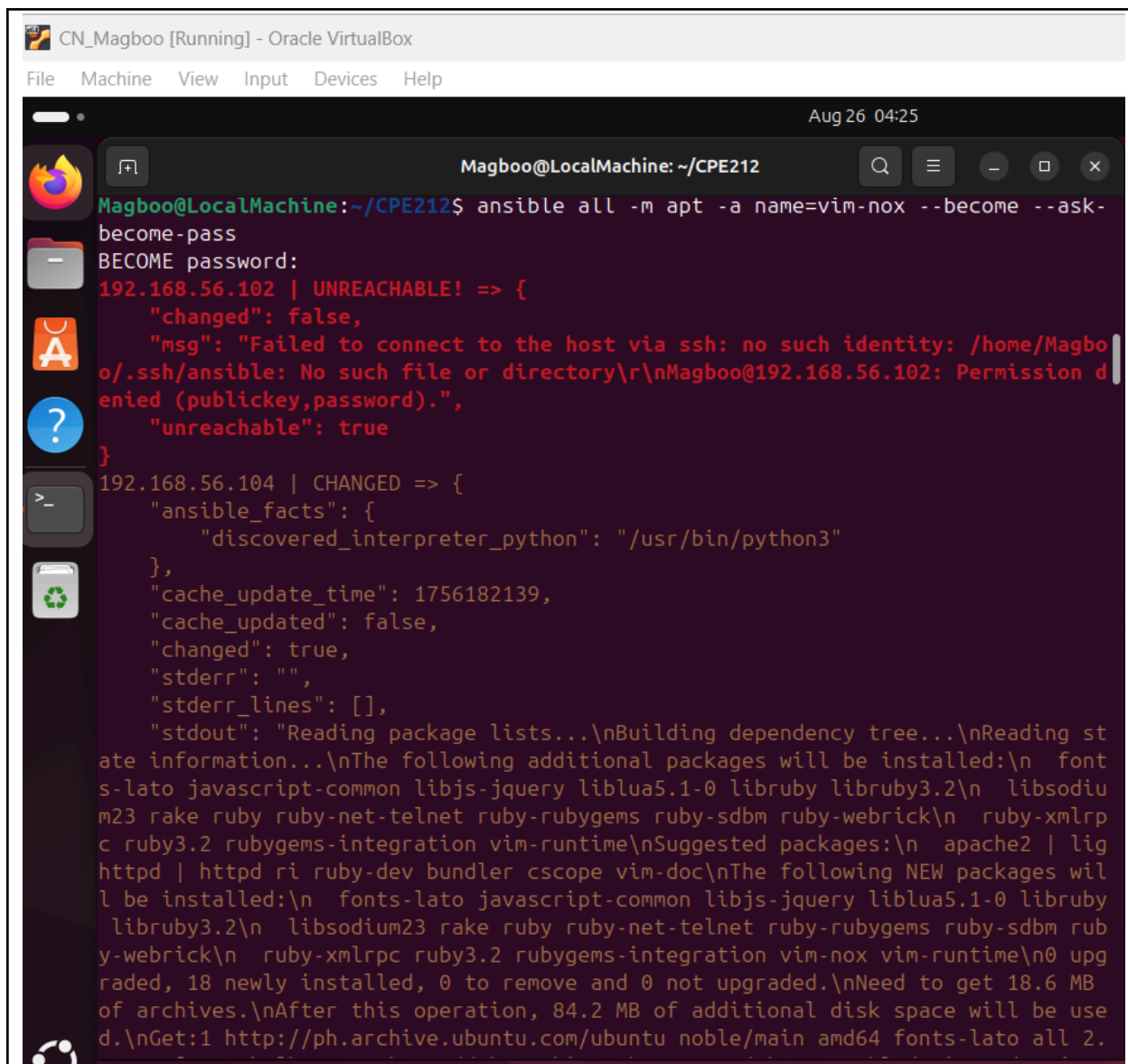
Try editing the command and add something that would elevate the privilege. Issue the command *ansible all -m apt -a update\_cache=true --become --ask-become-pass*. Enter the sudo password when prompted. You will notice now that the output of this command is a success. The *update\_cache=true* is the same thing as running *sudo apt update*. The *--become* command elevate the privileges and the *--ask-become-pass* asks for the password. For now, even if we only have changed the packaged index, we were able to change something on the remote server.



```
192.168.56.102 | UNREACHABLE! => {
  "changed": false,
  "msg": "Failed to connect to the host via ssh: no such identity: /home/Magboo/.ssh/ansible: No such file or directory\r\nMagboo@192.168.56.102: Permission denied (publickey,password).",
  "unreachable": true
}
192.168.56.103 | CHANGED => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "cache_update_time": 1756182137,
  "cache_updated": true,
  "changed": true
}
192.168.56.104 | CHANGED => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "cache_update_time": 1756182139,
  "cache_updated": true,
  "changed": true
}
Magboo@LocalMachine: ~/CPE212$
```

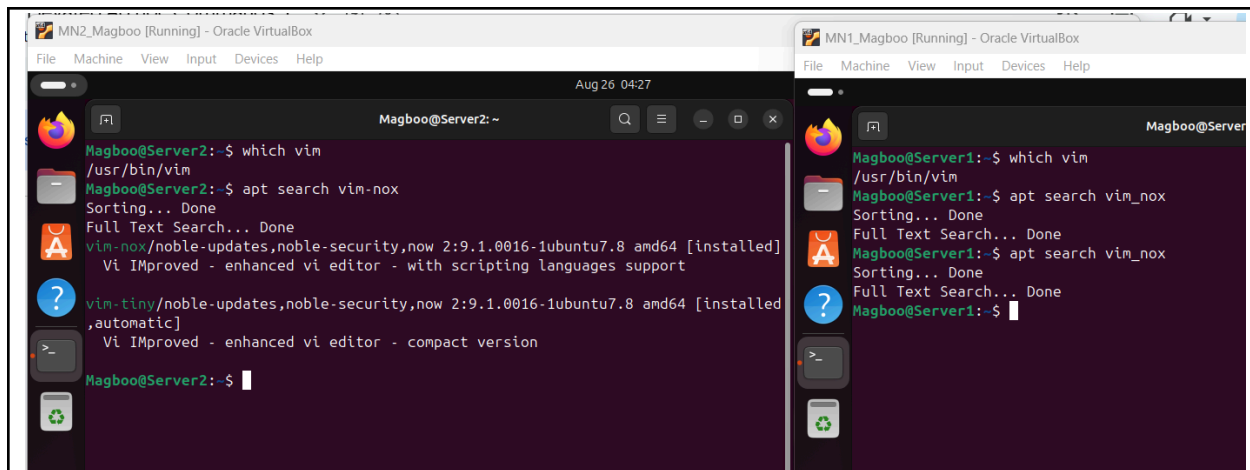
You may notice after the second command was executed, the status is CHANGED compared to the first command, which is FAILED.

2. Let's try to install VIM, which is an almost compatible version of the UNIX editor Vi. To do this, we will just changed the module part in 1.1 instruction. Here is the command: `ansible all -m apt -a name=vim-nox --become --ask-become-pass`. The command would take some time after typing the password because the local machine instructed the remote servers to actually install the package.

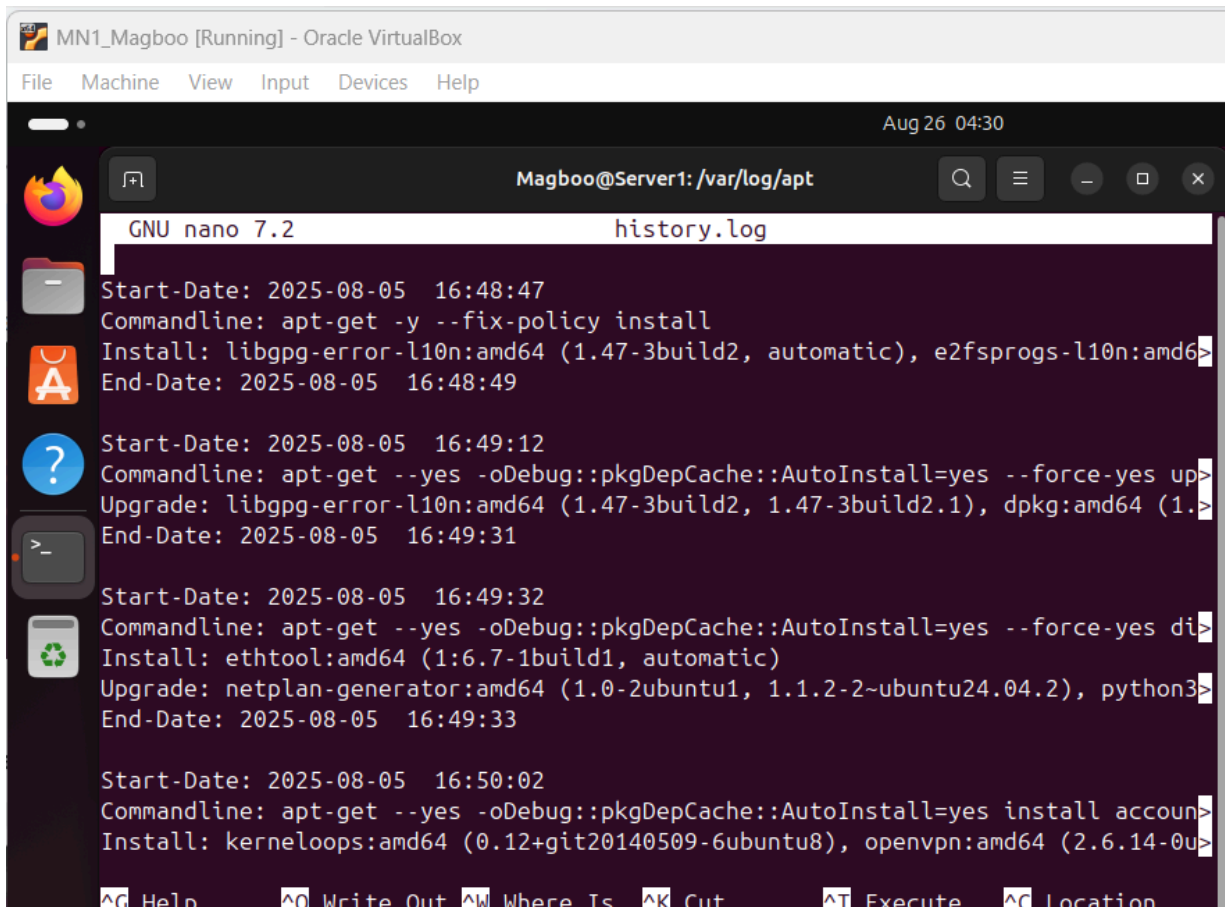


```
Magboo@LocalMachine: ~/CPE212
Magboo@LocalMachine:~/CPE212$ ansible all -m apt -a name=vim-nox --become --ask-
become-pass
BECOME password:
192.168.56.102 | UNREACHABLE! => {
  "changed": false,
  "msg": "Failed to connect to the host via ssh: no such identity: /home/Magboo/.ssh/ansible: No such file or directory\r\nMagboo@192.168.56.102: Permission denied (publickey,password).",
  "unreachable": true
}
192.168.56.104 | CHANGED => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "cache_update_time": 1756182139,
  "cache_updated": false,
  "changed": true,
  "stderr": "",
  "stderr_lines": [],
  "stdout": "Reading package lists...\nBuilding dependency tree...\nReading state information...\nThe following additional packages will be installed:\n fonts-lato javascript-common libjs-jquery liblua5.1-0 libruby libruby3.2\n libsodium23 rake ruby ruby-net-telnet ruby-rubygems ruby-sdbm ruby-webrick\n ruby-xmlrpc ruby3.2 rubygems-integration vim-runtime\nSuggested packages:\n apache2 | lighttpd | httpd ri ruby-dev bundler cscope vim-doc\nThe following NEW packages will be installed:\n fonts-lato javascript-common libjs-jquery liblua5.1-0 libruby libruby3.2\n libsodium23 rake ruby ruby-net-telnet ruby-rubygems ruby-sdbm ruby-webrick\n ruby-xmlrpc ruby3.2 rubygems-integration vim-nox vim-runtime\n0 upgraded, 18 newly installed, 0 to remove and 0 not upgraded.\nNeed to get 18.6 MB of archives.\nAfter this operation, 84.2 MB of additional disk space will be used.\nGet:1 http://ph.archive.ubuntu.com/ubuntu noble/main amd64 fonts-lato all 2.
```

2.1 Verify that you have installed the package in the remote servers. Issue the command *which vim* and the command *apt search vim-nox* respectively. Was the command successful?



2.2 Check the logs in the servers using the following commands: `cd /var/log`. After this, issue the command `ls`, go to the folder `apt` and open `history.log`. Describe what you see in the `history.log`.

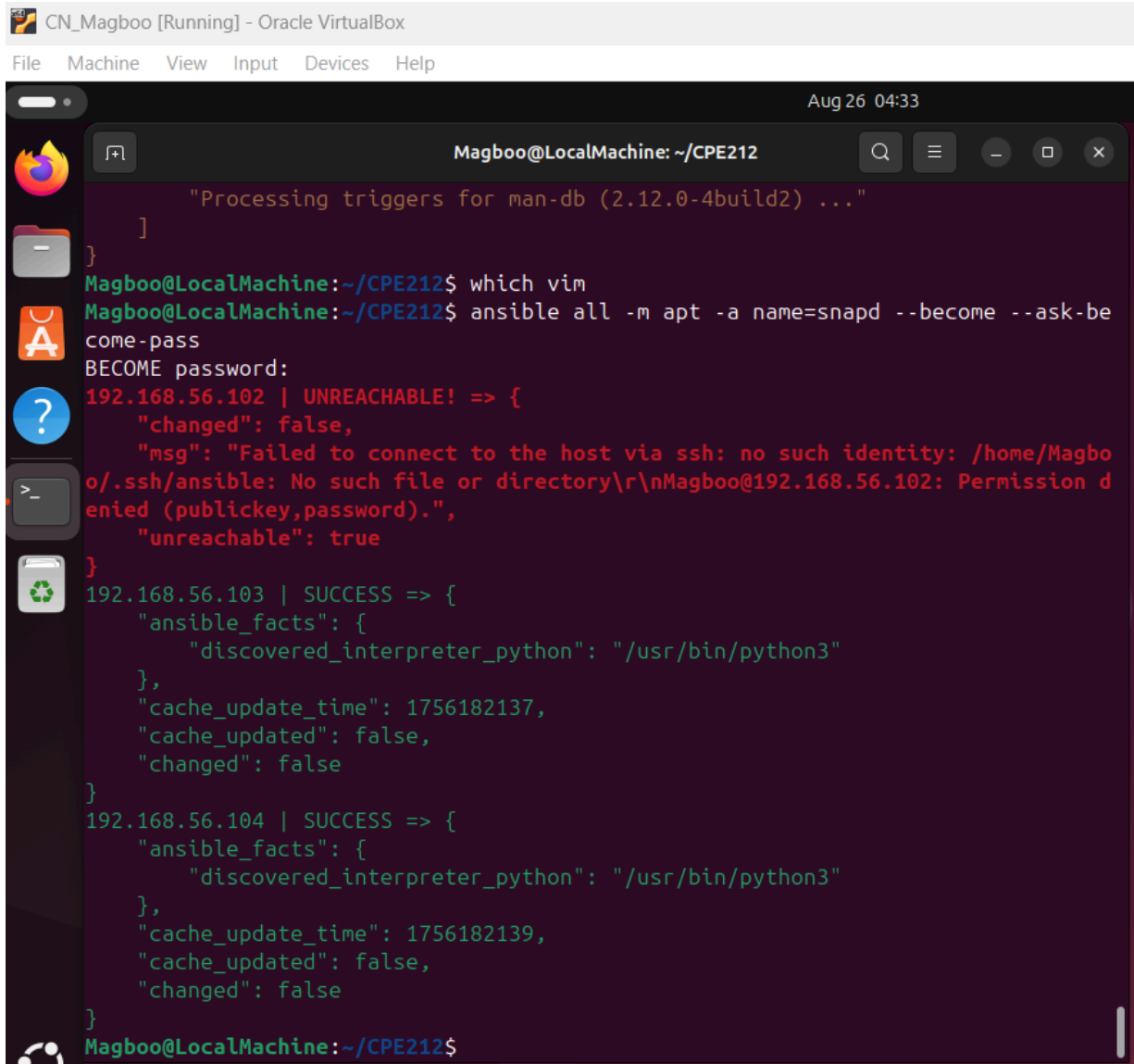


its all confirmation for install update upgrade.

3. This time, we will install a package called snapd. Snap is pre-installed in Ubuntu system. However, our goal is to create a command that checks for the latest installation package.

3.1 Issue the command: *ansible all -m apt -a name=snapd --become --ask-become-pass*

Can you describe the result of this command? Is it a success? Did it change anything in the remote servers?

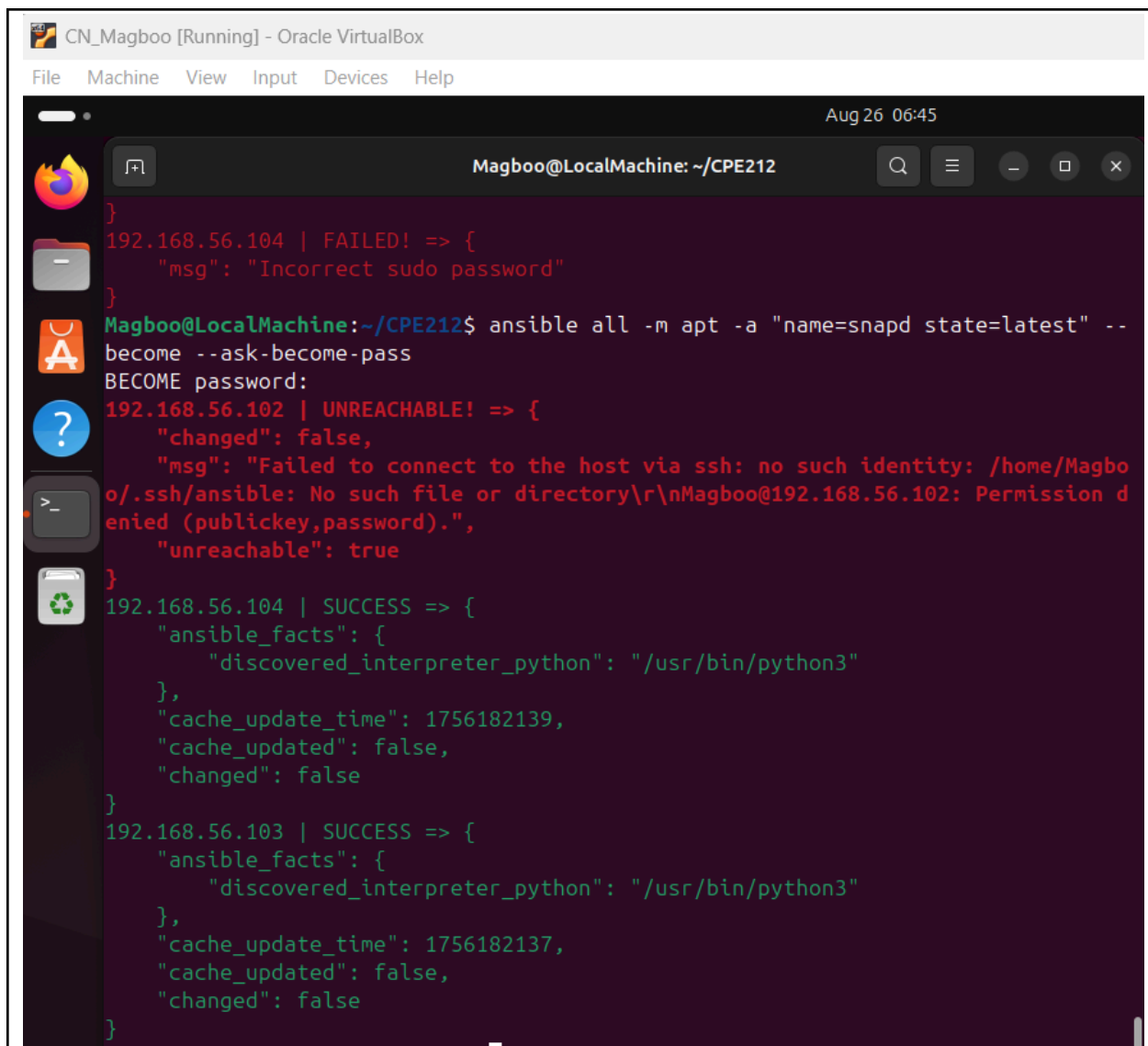


```

CN_Magboo [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Aug 26 04:33
Magboo@LocalMachine: ~/CPE212
"Processing triggers for man-db (2.12.0-4build2) ..."
]
Magboo@LocalMachine:~/CPE212$ which vim
Magboo@LocalMachine:~/CPE212$ ansible all -m apt -a name=snapd --become --ask-become-pass
BECOME password:
192.168.56.102 | UNREACHABLE! => {
  "changed": false,
  "msg": "Failed to connect to the host via ssh: no such identity: /home/Magboo/.ssh/ansible: No such file or directory\r\nMagboo@192.168.56.102: Permission denied (publickey,password).",
  "unreachable": true
}
192.168.56.103 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "cache_update_time": 1756182137,
  "cache_updated": false,
  "changed": false
}
192.168.56.104 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "cache_update_time": 1756182139,
  "cache_updated": false,
  "changed": false
}
Magboo@LocalMachine:~/CPE212$
```

nothing changed because both ip address showed "changed": false

3.2 Now, try to issue this command: *ansible all -m apt -a "name=snapd state=latest" --become --ask-become-pass*



The screenshot shows a terminal window titled "Magboo@LocalMachine: ~/CPE212" with a timestamp of "Aug 26 06:45". The terminal displays the output of an Ansible command: `ansible all -m apt -a "name=snapd state=latest" --become --ask-become-pass`. The output shows three hosts: 192.168.56.104 (FAILED), 192.168.56.102 (UNREACHABLE), and 192.168.56.103 (SUCCESS). The failed host 192.168.56.104 has a message: "Incorrect sudo password". The unreachable host 192.168.56.102 has a message: "Failed to connect to the host via ssh: no such identity: /home/Magboo/.ssh/ansible: No such file or directory\r\nMagboo@192.168.56.102: Permission denied (publickey,password).". The successful host 192.168.56.103 shows the fact that the package is installed and the interpreter is /usr/bin/python3.

```
}
192.168.56.104 | FAILED! => {
  "msg": "Incorrect sudo password"
}
Magboo@LocalMachine:~/CPE212$ ansible all -m apt -a "name=snapd state=latest" --become --ask-become-pass
BECOME password:
192.168.56.102 | UNREACHABLE! => {
  "changed": false,
  "msg": "Failed to connect to the host via ssh: no such identity: /home/Magboo/.ssh/ansible: No such file or directory\r\nMagboo@192.168.56.102: Permission denied (publickey,password).",
  "unreachable": true
}
192.168.56.104 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "cache_update_time": 1756182139,
  "cache_updated": false,
  "changed": false
}
192.168.56.103 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "cache_update_time": 1756182137,
  "cache_updated": false,
  "changed": false
}
```

Describe the output of this command. Notice how we added the command `state=latest` and placed them in double quotations.

4. At this point, make sure to commit all changes to GitHub.

The screenshot shows a GitHub repository page for 'CPE232' by user 'MagbooMattClemence'. The repository is public and has 1 branch (main) and 0 tags. The commit history shows 3 commits. The files listed are README.md, ansible.cfg, inventory.yaml, and readme.md. The README file is selected, showing the title 'CPE232'.

File	Commit	Time
README.md	Initial commit	yesterday
ansible.cfg	activity4 task1	now
inventory.yaml	activity4 task1	now
readme.md	hehe	yesterday

Below the file list, the README content is displayed, showing the title 'CPE232'.

## Task 2: Writing our First Playbook

1. With ad hoc commands, we can simplify the administration of remote servers. For example, we can install updates, packages, and applications, etc. However, the real strength of ansible comes from its playbooks. When we write a playbook, we can define the state that we want our servers to be in and the place or commands that ansible will carry out to bring to that state. You can use an editor to create a playbook. Before we proceed, make sure that you are in the directory of the repository that we use in the previous activities (*CPE232\_yourname*). Issue the command *nano install\_apache.yml*. This will create a playbook file called *install\_apache.yml*. The .yml is the basic standard extension for playbook files.

When the editor appears, type the following:



```
GNU nano 4.8          install_apache.yml
--
- hosts: all
  become: true
  tasks:

  - name: install apache2 package
    apt:
      name: apache2
```

```
Magboo@LocalMachine: ~/CPE232
GNU nano 7.2          install_apache.yml
--
- hosts: all
  become: true
  tasks:

  - name: install apache2 package
    apt:
      name: apache2
```

Make sure to save the file. Take note also of the alignments of the texts.

2. Run the yml file using the command: *ansible-playbook --ask-become-pass install\_apache.yml*. Describe the result of this command.

```

Magboo@LocalMachine:~/CPE232$ nano install_apache.yml
Magboo@LocalMachine:~/CPE232$ ansible-playbook --ask-become-pass install_apache.
yml
BECOME password:

PLAY [all] *****

TASK [Gathering Facts] *****
fatal: [192.168.56.102]: UNREACHABLE! => {"changed": false, "msg": "Failed to co
nnect to the host via ssh: no such identity: /home/Magboo/.ssh/ansible: No such
file or directory\r\nMagboo@192.168.56.102: Permission denied (publickey,password).", "unreachable": true}
ok: [192.168.56.104]
ok: [192.168.56.103]

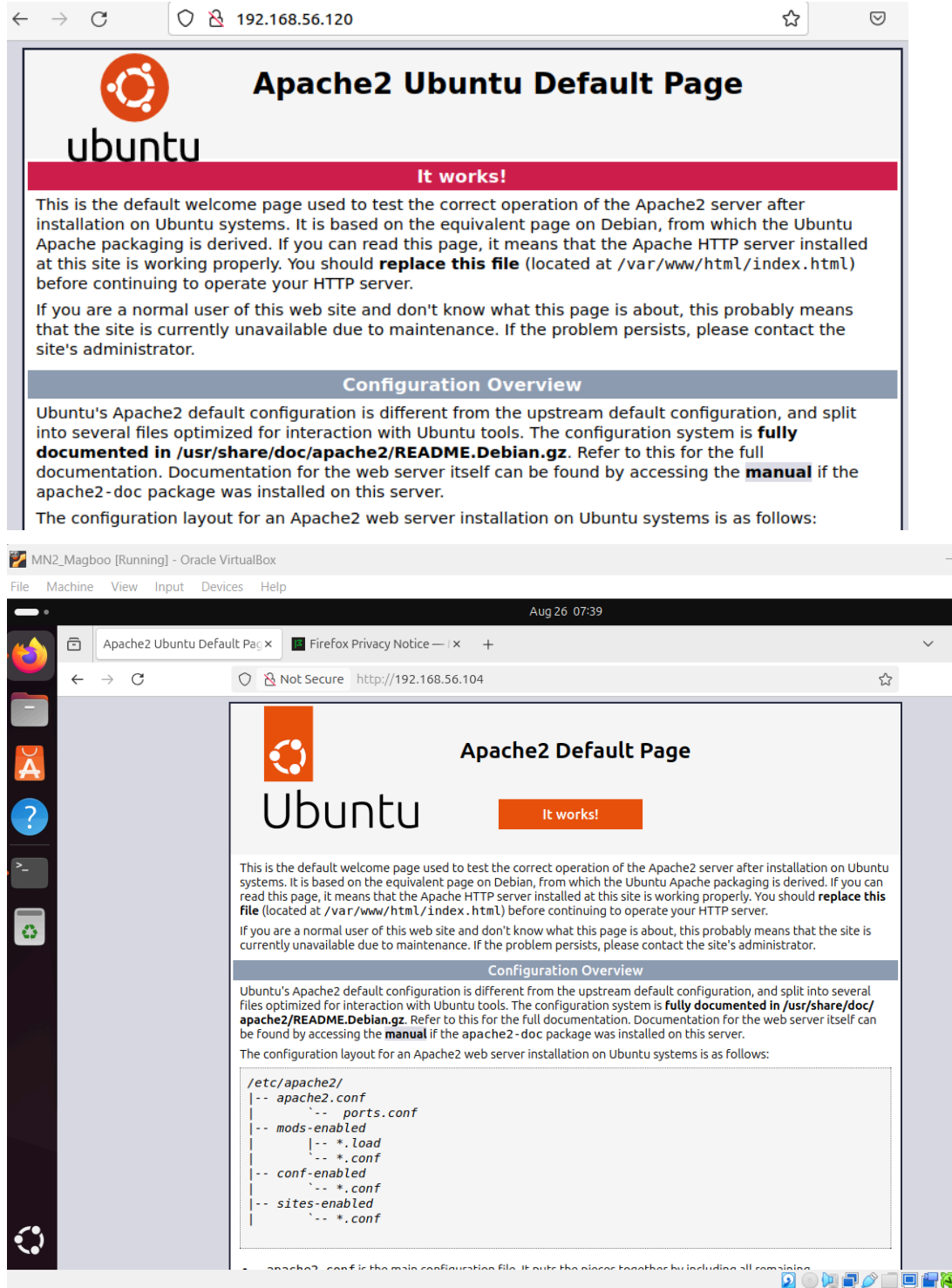
TASK [install apache2 package] *****
changed: [192.168.56.104]
changed: [192.168.56.103]

PLAY RECAP *****
192.168.56.102      : ok=0    changed=0    unreachable=1    failed=0    s
kipped=0    rescued=0    ignored=0
192.168.56.103      : ok=2    changed=1    unreachable=0    failed=0    s
kipped=0    rescued=0    ignored=0
192.168.56.104      : ok=2    changed=1    unreachable=0    failed=0    s
kipped=0    rescued=0    ignored=0

Magboo@LocalMachine:~/CPE232$

```

3. To verify that apache2 was installed automatically in the remote servers, go to the web browsers on each server and type its IP address. You should see something like this.



4. Try to edit the *install\_apache.yml* and change the name of the package to any name that will not be recognized. What is the output?



Magboo@LocalMachine: ~/CPE232

GNU nano 7.2

install\_apache.yml

```
--
- hosts: all
  become: true
  tasks:

    - name: install apache2 package
      apt:
        name: wowowin
```

```
Magboo@LocalMachine:~/CPE232$ ansible-playbook --ask-become-pass install_apache.
yml
```

```
BECOME password:
```

```
PLAY [all] *****
```

```
TASK [Gathering Facts] *****
```

```
fatal: [192.168.56.102]: UNREACHABLE! => {"changed": false, "msg": "Failed to co
nnect to the host via ssh: no such identity: /home/Magboo/.ssh/ansible: No such
file or directory\r\nMagboo@192.168.56.102: Permission denied (publickey,passwor
d).", "unreachable": true}
```

```
ok: [192.168.56.103]
```

```
ok: [192.168.56.104]
```

```
TASK [install apache2 package] *****
```

```
fatal: [192.168.56.103]: FAILED! => {"changed": false, "msg": "No package matchi
ng 'wowowin' is available"}
```

```
fatal: [192.168.56.104]: FAILED! => {"changed": false, "msg": "No package matchi
ng 'wowowin' is available"}
```

```
PLAY RECAP *****
```

192.168.56.102	: ok=0	changed=0	unreachable=1	failed=0	s
kipped=0	rescued=0	ignored=0			
192.168.56.103	: ok=1	changed=0	unreachable=0	failed=1	s
kipped=0	rescued=0	ignored=0			
192.168.56.104	: ok=1	changed=0	unreachable=0	failed=1	s
kipped=0	rescued=0	ignored=0			

```
Magboo@LocalMachine:~/CPE232$
```

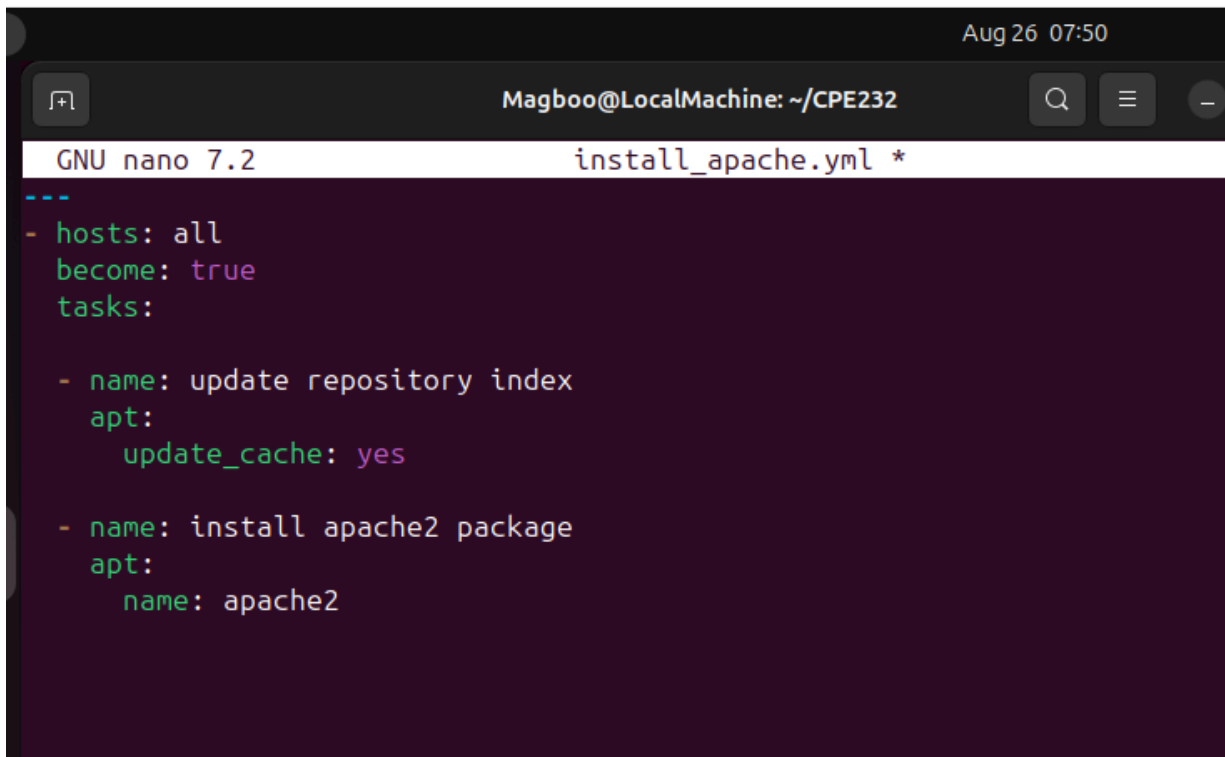
5. This time, we are going to put additional task to our playbook. Edit the *install\_apache.yml*. As you can see, we are now adding an additional command, which is the *update\_cache*. This command updates existing package-indexes on a supporting distro but not upgrading installed-packages (utilities) that were being installed.

```
---
- hosts: all
  become: true
  tasks:

    - name: update repository index
      apt:
        update_cache: yes

    - name: install apache2 package
      apt:
        name: apache2
```

Save the changes to this file and exit.



```
Aug 26 07:50
Magboo@LocalMachine: ~/CPE232
GNU nano 7.2 install_apache.yml *
---
- hosts: all
  become: true
  tasks:

    - name: update repository index
      apt:
        update_cache: yes

    - name: install apache2 package
      apt:
        name: apache2
```

6. Run the playbook and describe the output. Did the new command change anything on the remote servers?

```

Magboo@LocalMachine:~/CPE232$ ansible-playbook --ask-become-pass install_apache.
.yml
BECOME password:

PLAY [all] *****

TASK [Gathering Facts] *****
fatal: [192.168.56.102]: UNREACHABLE! => {"changed": false, "msg": "Failed to co
nnect to the host via ssh: no such identity: /home/Magboo/.ssh/ansible: No such
file or directory\r\nMagboo@192.168.56.102: Permission denied (publickey,password).", "unreachable": true}
ok: [192.168.56.103]
ok: [192.168.56.104]

TASK [update repository index] *****
changed: [192.168.56.103]
changed: [192.168.56.104]

TASK [install apache2 package] *****
ok: [192.168.56.103]
ok: [192.168.56.104]

PLAY RECAP *****
192.168.56.102      : ok=0    changed=0    unreachable=1    failed=0    s
kipped=0    rescued=0    ignored=0
192.168.56.103      : ok=3    changed=1    unreachable=0    failed=0    s
kipped=0    rescued=0    ignored=0
192.168.56.104      : ok=3    changed=1    unreachable=0    failed=0    s
kipped=0    rescued=0    ignored=0

```

7. Edit again the *install\_apache.yml*. This time, we are going to add a PHP support for the apache package we installed earlier.

```

---
- hosts: all
  become: true
  tasks:

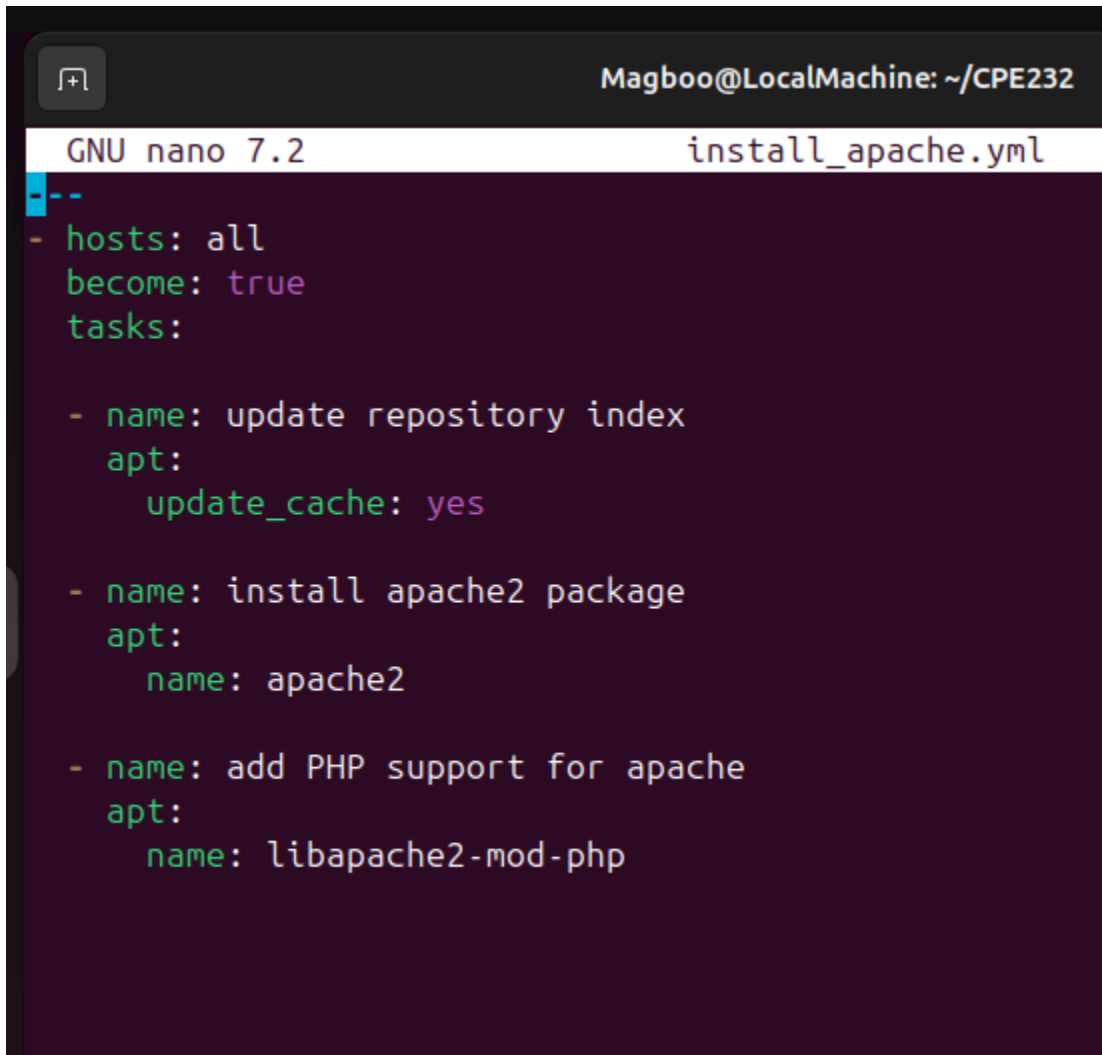
    - name: update repository index
      apt:
        update_cache: yes

    - name: install apache2 package
      apt:
        name: apache2

    - name: add PHP support for apache
      apt:
        name: libapache2-mod-php

```

Save the changes to this file and exit.



```
Magboo@LocalMachine: ~/CPE232
GNU nano 7.2 install_apache.yml
---
- hosts: all
  become: true
  tasks:

    - name: update repository index
      apt:
        update_cache: yes

    - name: install apache2 package
      apt:
        name: apache2

    - name: add PHP support for apache
      apt:
        name: libapache2-mod-php
```

8. Run the playbook and describe the output. Did the new command change anything on the remote servers?

Aug 26 08:10

Magboo@LocalMachine: ~/CPE232

PLAY [all] \*\*\*\*\*

TASK [Gathering Facts] \*\*\*\*\*

```
fatal: [192.168.56.102]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: no such identity: /home/Magboo/.ssh/ansible: No such file or directory\r\nMagboo@192.168.56.102: Permission denied (publickey,password).", "unreachable": true}
```

ok: [192.168.56.104]

ok: [192.168.56.103]

TASK [update repository index] \*\*\*\*\*

changed: [192.168.56.104]

changed: [192.168.56.103]

TASK [install apache2 package] \*\*\*\*\*

ok: [192.168.56.103]

ok: [192.168.56.104]

TASK [add PHP support for apache] \*\*\*\*\*

changed: [192.168.56.103]

changed: [192.168.56.104]

PLAY RECAP \*\*\*\*\*

192.168.56.102	: ok=0	changed=0	unreachable=1	failed=0	s
----------------	--------	-----------	---------------	----------	---

kipped=0	rescued=0	ignored=0
----------	-----------	-----------

192.168.56.103	: ok=4	changed=2	unreachable=0	failed=0	s
----------------	--------	-----------	---------------	----------	---

kipped=0	rescued=0	ignored=0
----------	-----------	-----------

192.168.56.104	: ok=4	changed=2	unreachable=0	failed=0	s
----------------	--------	-----------	---------------	----------	---

kipped=0	rescued=0	ignored=0
----------	-----------	-----------

Magboo@LocalMachine: ~/CPE232\$



MN1\_Magboo [Running] - Oracle VirtualBox

File Machine View Input Devices Help

Aug 26 08:16

Magboo@Server1: /var/log/apt

GNU nano 7.2 history.log

```
Start-Date: 2025-08-05 16:48:47
Commandline: apt-get -y --fix-policy install
Install: libgpg-error-l10n:amd64 (1.47-3build2, automatic), e2fsprogs-l10n:amd64 (1.47-3build2, automatic)
End-Date: 2025-08-05 16:48:49

Start-Date: 2025-08-05 16:49:12
Commandline: apt-get --yes -oDebug::pkgDepCache::AutoInstall=yes --force-yes upgrade
Upgrade: libgpg-error-l10n:amd64 (1.47-3build2, 1.47-3build2.1), dpkg:amd64 (1.47-3build2, 1.47-3build2.1)
End-Date: 2025-08-05 16:49:31

Start-Date: 2025-08-05 16:49:32
Commandline: apt-get --yes -oDebug::pkgDepCache::AutoInstall=yes --force-yes dist-upgrade
Install: ethtool:amd64 (1:6.7-1build1, automatic)
Upgrade: netplan-generator:amd64 (1.0-2ubuntu1, 1.1.2-2~ubuntu24.04.2), python3-netplan (1.1.2-2~ubuntu24.04.2, 1.1.2-2~ubuntu24.04.2)
End-Date: 2025-08-05 16:49:33

Start-Date: 2025-08-05 16:50:02
Commandline: apt-get --yes -oDebug::pkgDepCache::AutoInstall=yes install accountsservice
Install: kerneloops:amd64 (0.12+git20140509-6ubuntu8), openvpn:amd64 (2.6.14-0ubuntu1)
End-Date: 2025-08-05 16:50:02
```

9. Finally, make sure that we are in sync with GitHub. Provide the link of your GitHub repository.








```

Magboo@LocalMachine:~/CPE232$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        install_apache.yml

nothing added to commit but untracked files present (use "git add" to track)
Magboo@LocalMachine:~/CPE232$ git add install_apache.yml
Magboo@LocalMachine:~/CPE232$ git commit -m "activity4 task 2"
[main 8a8266b] activity4 task 2
 1 file changed, 16 insertions(+)
 create mode 100644 install_apache.yml
Magboo@LocalMachine:~/CPE232$ git push origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 10 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 413 bytes | 413.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:MagbooMattClemence/CPE232.git
 4a649bf..8a8266b  main -> main

```

 <b>MagbooMattClemence</b> activity4 task 2		8a8266b · now	 4 Commits
	README.md	Initial commit	yesterday
	ansible.cfg	activity4 task1	1 hour ago
	install_apache.yml	activity4 task 2	now
	inventory.yml	activity4 task1	1 hour ago
	readme.md	hehe	yesterday

github: [MagbooMattClemence/CPE232](https://github.com/MagbooMattClemence/CPE232)

### Reflections:

Answer the following:

1. What is the importance of using a playbook?

playbook is important because it can be use repeatedly making automation faster and easier with only ansible it would control nodes on a single manage node all at the same time making it easy.

2. Summarize what we have done on this activity.

i installed ansible in the machine then installed some necessary programs to servers(manage nodes) added support too for the ansible updated the repository index for both with a simple automation from install\_apache.yml