Name: Jozshua Amiel A. Alonzo	**Date Performed: September 10, 2022**
Course/Section: CPE31S23	**Date Submitted: September 13, 2022**
Instructor: Dr. Jonathan Taylar	**Semester and SY: 2022-2023**

Activity 4: Running Elevated Ad hoc Commands

1. Objectives:

1.1 Use commands that makes changes to remote machines

1.2 Use playbook in automating ansible commands

2. Discussion:

GITHUB LINK: https://github.com/jozshua/CPE232_Jozshua.git

Elevated Ad hoc commands

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.

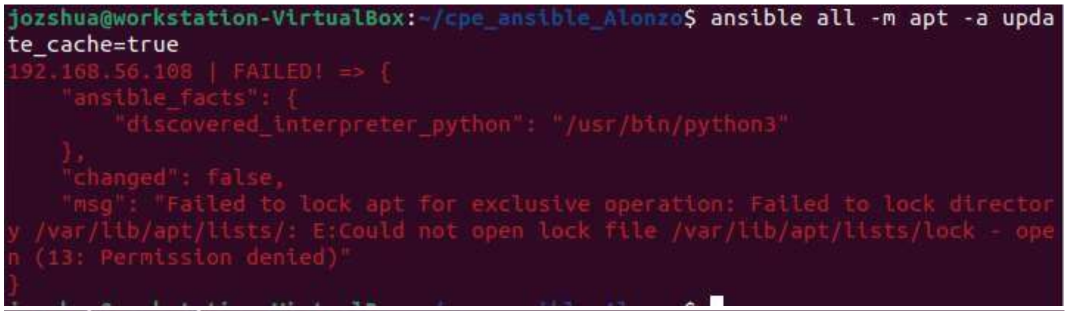
Playbooks record and execute **Ansible**'s configuration, deployment, and orchestration functions. They can describe a policy you want your remote systems to enforce, or a set of steps in a general IT process. If Ansible modules are the tools in your workshop, playbooks are your instruction manuals, 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. Working with playbooks — Ansible Documentation

Task 1: Run elevated ad hoc commands

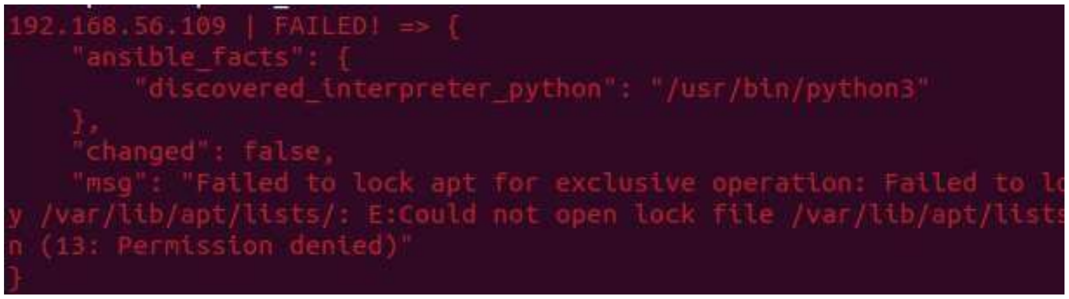
1. Locally, we use the command *sudo apt update* when we want to download package information from all configured resources. The sources often defined in /etc/apt/sources.list file and other files located in /etc/apt/sources.list.d/ directory. So, when you run update command, it downloads the package 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?



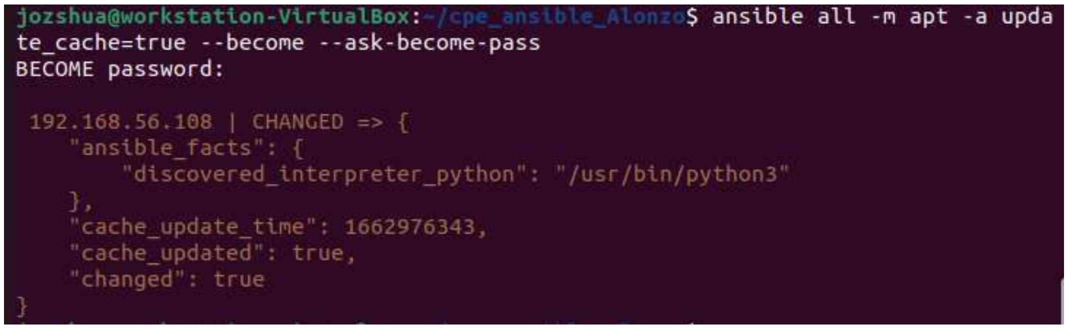
```
jozshua@workstation-VirtualBox:~/cpe_ansible_Alonzo$ ansible all -m apt -a upda
te_cache=true
192.168.56.108 | FAILED! => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "msg": "Failed to lock apt for exclusive operation: Failed to lock director
y /var/lib/apt/lists/: E:Could not open lock file /var/lib/apt/lists/lock - ope
n (13: Permission denied)"
}
```



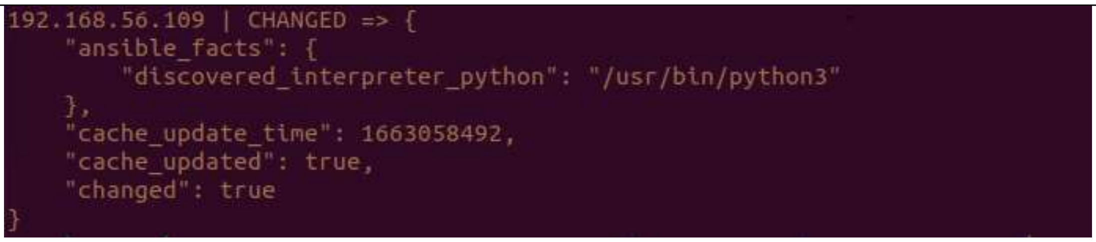
```
192.168.56.109 | FAILED! => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "msg": "Failed to lock apt for exclusive operation: Failed to lo
y /var/lib/apt/lists/: E:Could not open lock file /var/lib/apt/lists
n (13: Permission denied)"
}
```

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.

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

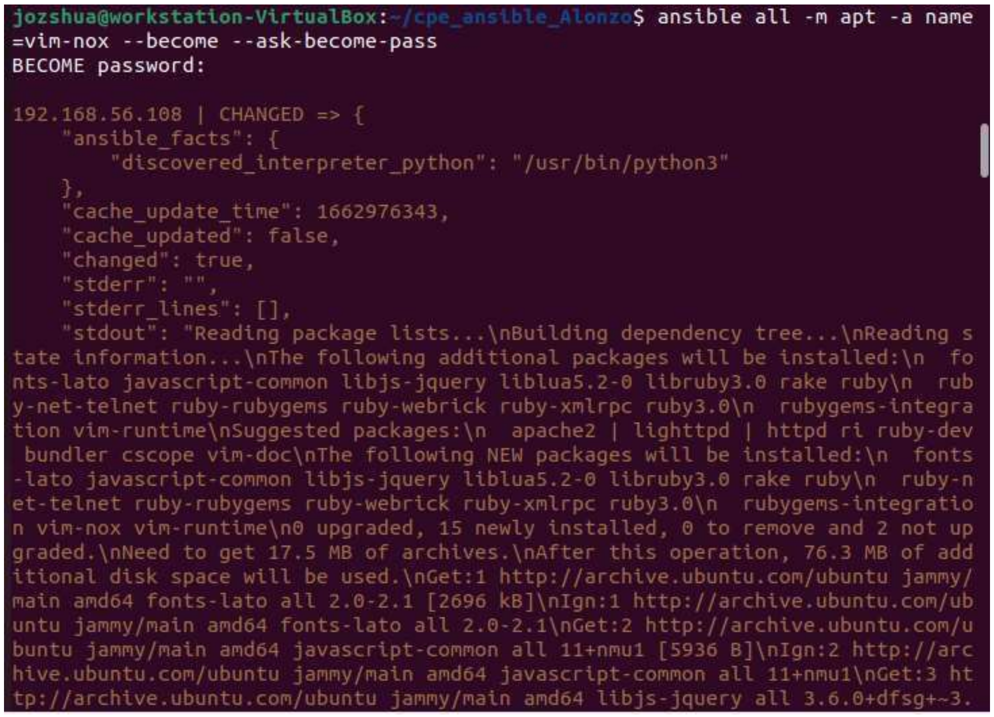


```
jozshua@workstation-VirtualBox:~/cpe_ansible_Alonzo$ ansible all -m apt -a upda
te_cache=true --become --ask-become-pass
BECOME password:

 192.168.56.108 | CHANGED => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "cache_update_time": 1662976343,
    "cache_updated": true,
    "changed": true
}
```

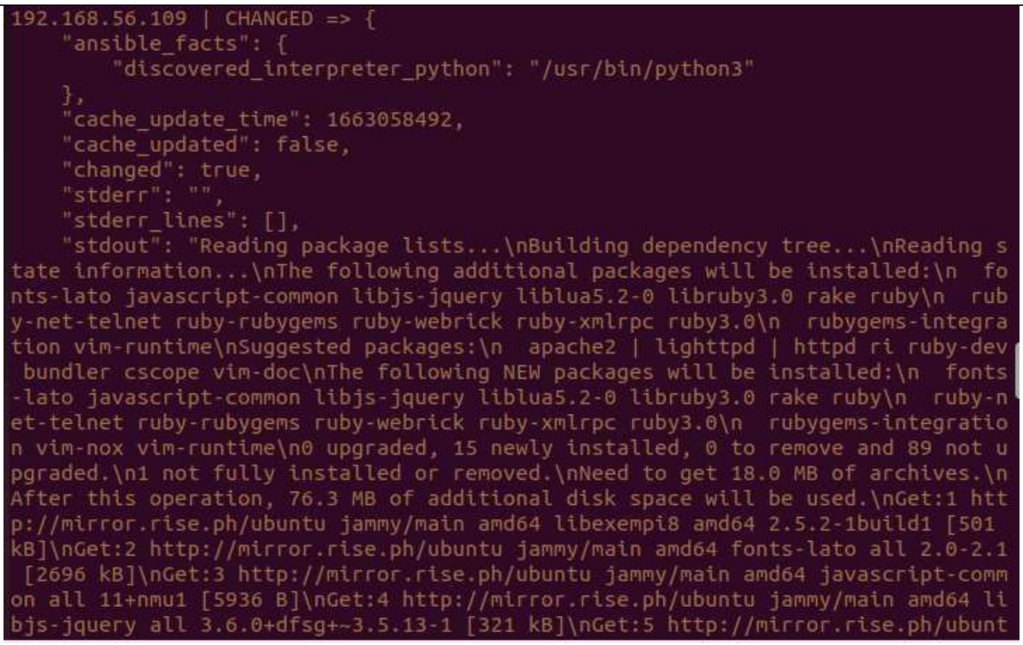


```
192.168.56.109 | CHANGED => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "cache_update_time": 1663058492,
    "cache_updated": true,
    "changed": true
}
```

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.

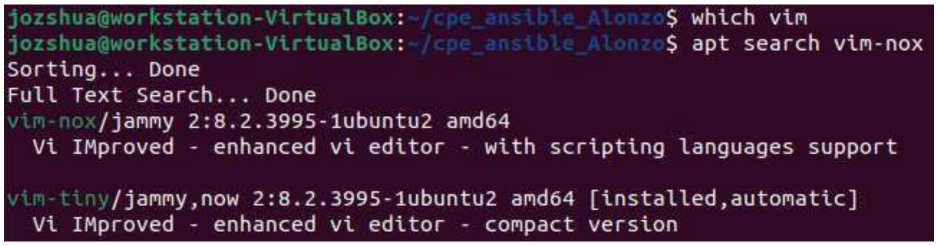


```
jozshua@workstation-VirtualBox:~/cpe_ansible_Alonzo$ ansible all -m apt -a name
=vim-nox --become --ask-become-pass
BECOME password:

192.168.56.108 | CHANGED => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "cache_update_time": 1662976343,
    "cache_updated": false,
    "changed": true,
    "stderr": "",
    "stderr_lines": [],
    "stdout": "Reading package lists...\nBuilding dependency tree...\nReading s
tate information...\nThe following additional packages will be installed:\n  fo
nts-lato javascript-common libjs-jquery liblua5.2-0 libruby3.0 rake ruby\n  rub
y-net-telnet ruby-rubygems ruby-webrick ruby-xmlrpc ruby3.0\n  rubygems-integra
tion 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 liblua5.2-0 libruby3.0 rake ruby\n  ruby-n
et-telnet ruby-rubygems ruby-webrick ruby-xmlrpc ruby3.0\n  rubygems-integratio
n vim-nox vim-runtime\n0 upgraded, 15 newly installed, 0 to remove and 2 not up
graded.\nNeed to get 17.5 MB of archives.\nAfter this operation, 76.3 MB of add
itional disk space will be used.\nGet:1 http://archive.ubuntu.com/ubuntu jammy/
main amd64 fonts-lato all 2.0-2.1 [2696 kB]\nIgn:1 http://archive.ubuntu.com/ub
untu jammy/main amd64 fonts-lato all 2.0-2.1\nGet:2 http://archive.ubuntu.com/u
buntu jammy/main amd64 javascript-common all 11+nmu1 [5936 B]\nIgn:2 http://arc
hive.ubuntu.com/ubuntu jammy/main amd64 javascript-common all 11+nmu1\nGet:3 ht
tp://archive.ubuntu.com/ubuntu jammy/main amd64 libjs-jquery all 3.6.0+dfsg+~3.
```



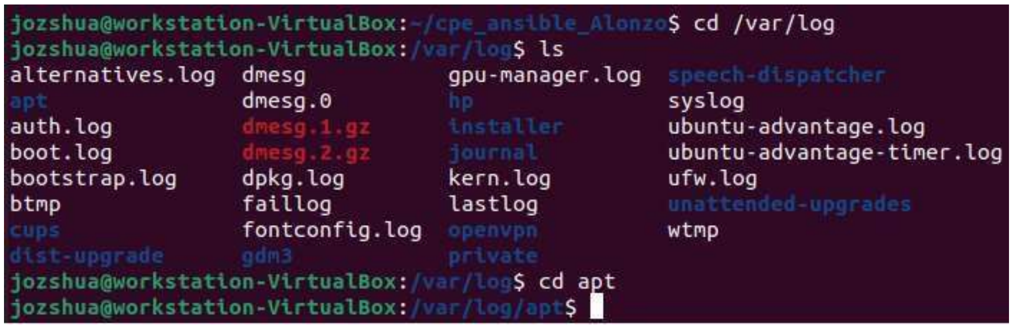
```
192.168.56.109 | CHANGED => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "cache_update_time": 1663058492,
    "cache_updated": false,
    "changed": true,
    "stderr": "",
    "stderr_lines": [],
    "stdout": "Reading package lists...\nBuilding dependency tree...\nReading s
tate information...\nThe following additional packages will be installed:\n  fo
nts-lato javascript-common libjs-jquery liblua5.2-0 libruby3.0 rake ruby\n  rub
y-net-telnet ruby-rubygems ruby-webrick ruby-xmlrpc ruby3.0\n  rubygems-integra
tion 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 liblua5.2-0 libruby3.0 rake ruby\n  ruby-n
et-telnet ruby-rubygems ruby-webrick ruby-xmlrpc ruby3.0\n  rubygems-integratio
n vim-nox vim-runtime\n0 upgraded, 15 newly installed, 0 to remove and 89 not u
pgraded.\n1 not fully installed or removed.\nNeed to get 18.0 MB of archives.\n
After this operation, 76.3 MB of additional disk space will be used.\nGet:1 htt
p://mirror.rise.ph/ubuntu jammy/main amd64 libexempi8 amd64 2.5.2-1build1 [501
kB]\nGet:2 http://mirror.rise.ph/ubuntu jammy/main amd64 fonts-lato all 2.0-2.1
 [2696 kB]\nGet:3 http://mirror.rise.ph/ubuntu jammy/main amd64 javascript-comm
on all 11+nmu1 [5936 B]\nGet:4 http://mirror.rise.ph/ubuntu jammy/main amd64 li
bjs-jquery all 3.6.0+dfsg+~3.5.13-1 [321 kB]\nGet:5 http://mirror.rise.ph/ubunt
```

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? Yes it is, The command shown the installed vim-nox information.



```
jozshua@workstation-VirtualBox:~/cpe_ansible_Alonzo$ which vim
jozshua@workstation-VirtualBox:~/cpe_ansible_Alonzo$ apt search vim-nox
Sorting... Done
Full Text Search... Done
vim-nox/jammy 2:8.2.3995-1ubuntu2 amd64
  Vi IMproved - enhanced vi editor - with scripting languages support

vim-tiny/jammy,now 2:8.2.3995-1ubuntu2 amd64 [installed,automatic]
  Vi IMproved - enhanced vi editor - compact version
```

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.

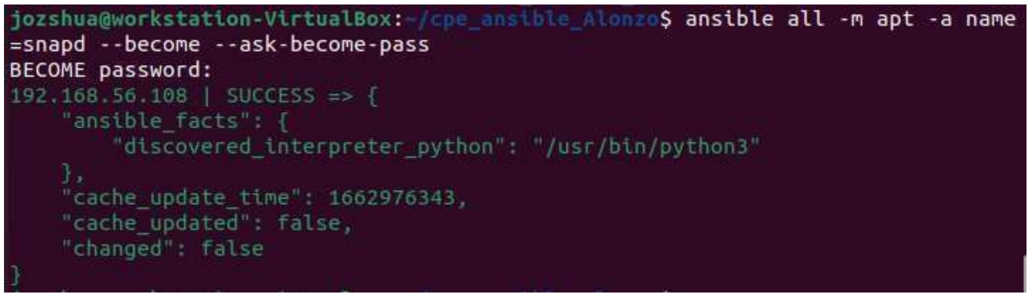


```
jozshua@workstation-VirtualBox:~/cpe_ansible_Alonzo$ cd /var/log
jozshua@workstation-VirtualBox:/var/log$ ls
alternatives.log  dmesg           gpu-manager.log  speech-dispatcher
apt               dmesg.0         hp               syslog
auth.log          dmesg.1.gz      installer        ubuntu-advantage.log
boot.log          dmesg.2.gz      journal          ubuntu-advantage-timer.log
bootstrap.log     dpkg.log        kern.log         ufw.log
btmp              faillog         lastlog          unattended-upgrades
cups              fontconfig.log  openvpn          wtmp
dist-upgrade      gdm3            private
jozshua@workstation-VirtualBox:/var/log$ cd apt
jozshua@workstation-VirtualBox:/var/log/apt$ 
```

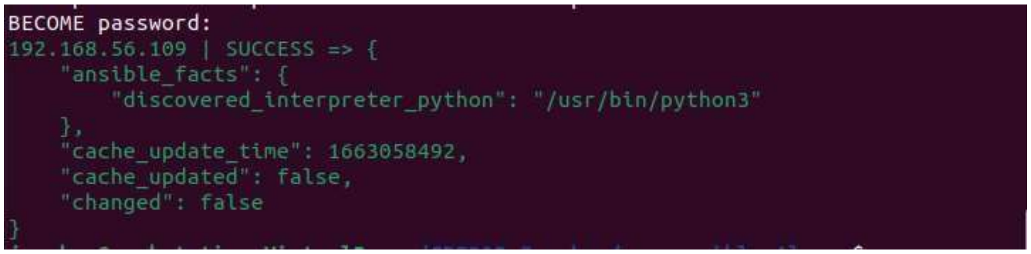
I have seen the process commands that I have run such as updates, upgrades and the installed package to use this ansible commands.

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*



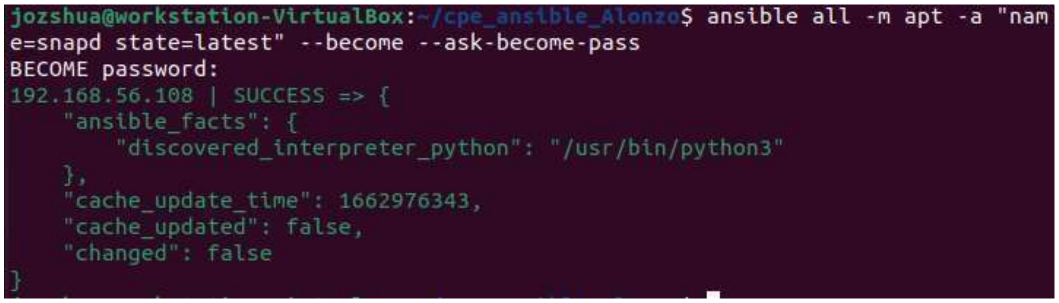
```
jozshua@workstation-VirtualBox:~/cpe_ansible_Alonzo$ ansible all -m apt -a name
=snapd --become --ask-become-pass
BECOME password:
192.168.56.108 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "cache_update_time": 1662976343,
    "cache_updated": false,
    "changed": false
}
```



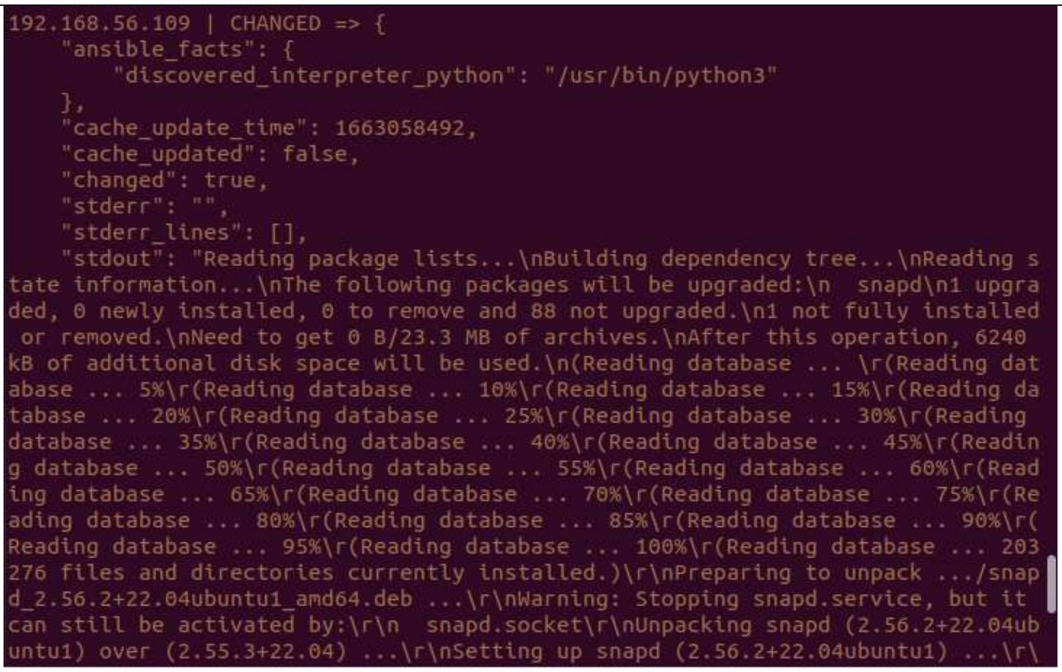
```
BECOME password:
192.168.56.109 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "cache_update_time": 1663058492,
    "cache_updated": false,
    "changed": false
}
```

   Can you describe the result of this command? Is it a success? Did it change anything in the remote servers? It was a success and it didn’t change anything from the remote servers.

   3.2 Now, try to issue this command: *ansible all -m apt -a "name=snapd state=latest" --become --ask-become-pass*
   Describe the output of this command. Notice how we added the command *state=latest* and placed them in double quotations.

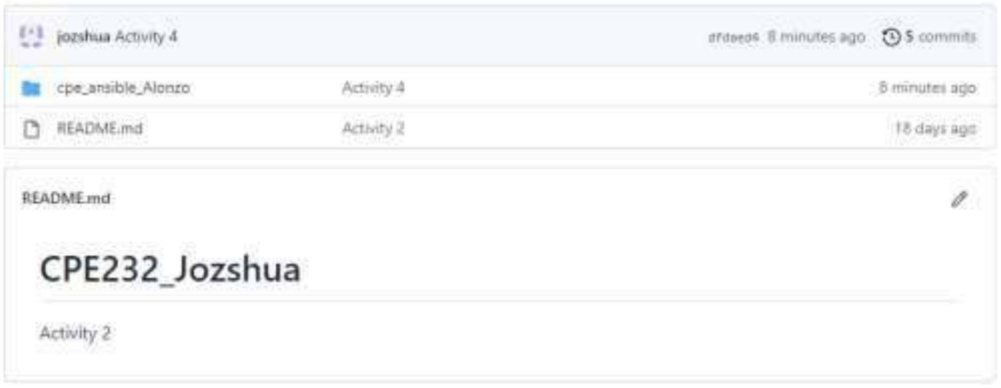


```
jozshua@workstation-VirtualBox:~/cpe_ansible_Alonzo$ ansible all -m apt -a "nam
e=snapd state=latest" --become --ask-become-pass
BECOME password:
192.168.56.108 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "cache_update_time": 1662976343,
    "cache_updated": false,
    "changed": false
}
```



```
192.168.56.109 | CHANGED => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "cache_update_time": 1663058492,
    "cache_updated": false,
    "changed": true,
    "stderr": "",
    "stderr_lines": [],
    "stdout": "Reading package lists...\nBuilding dependency tree...\nReading s
tate information...\nThe following packages will be upgraded:\n  snapd\n1 upgra
ded, 0 newly installed, 0 to remove and 88 not upgraded.\n1 not fully installed
 or removed.\nNeed to get 0 B/23.3 MB of archives.\nAfter this operation, 6240
kB of additional disk space will be used.\n(Reading database ... \r(Reading dat
abase ... 5%\r(Reading database ... 10%\r(Reading database ... 15%\r(Reading da
tabase ... 20%\r(Reading database ... 25%\r(Reading database ... 30%\r(Reading
database ... 35%\r(Reading database ... 40%\r(Reading database ... 45%\r(Readin
g database ... 50%\r(Reading database ... 55%\r(Reading database ... 60%\r(Read
ing database ... 65%\r(Reading database ... 70%\r(Reading database ... 75%\r(Re
ading database ... 80%\r(Reading database ... 85%\r(Reading database ... 90%\r(
Reading database ... 95%\r(Reading database ... 100%\r(Reading database ... 203
276 files and directories currently installed.)\r\nPreparing to unpack .../snap
d_2.56.2+22.04ubuntu1_amd64.deb ...\r\nWarning: Stopping snapd.service, but it
can still be activated by:\r\n  snapd.socket\r\nUnpacking snapd (2.56.2+22.04ub
untu1) over (2.55.3+22.04) ...\r\nSetting up snapd (2.56.2+22.04ubuntu1) ...\r\
```

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



jozshua Activity 4 — dfdeed6 8 minutes ago — 5 commits

cpe_ansible_Alonzo	Activity 4	8 minutes ago
README.md	Activity 2	18 days ago

README.md

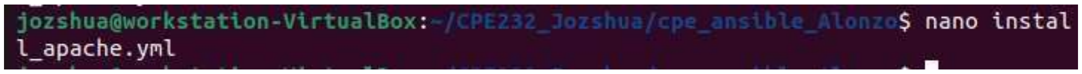
CPE232_Jozshua

Activity 2

Task 2: Writing our First Playbook

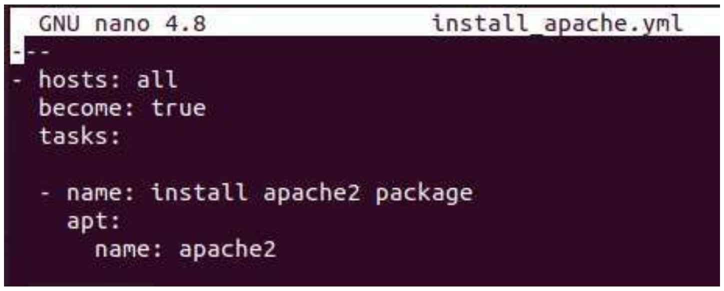
1. With ad hoc commands, we can simplify the administration of remote servers. For example, we can install updates, packages, and applications, etc. However, the real strength of ansible comes from its playbooks. When we write a playbook, we can define the state that we want our servers to be in and 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.



```
jozshua@workstation-VirtualBox:~/CPE232_Jozshua/cpe_ansible_Alonzo$ nano instal
l_apache.yml
```

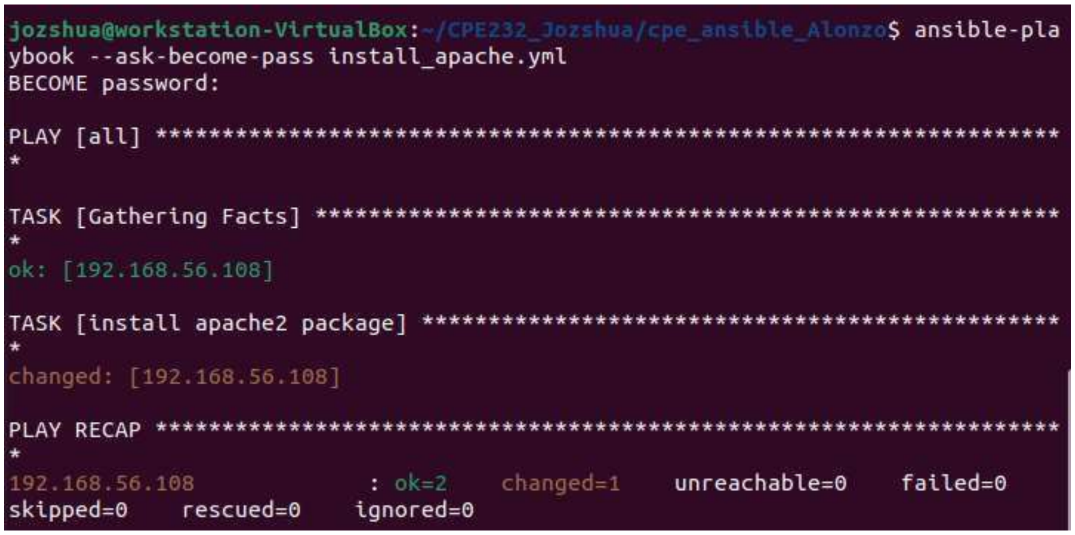
When the editor appears, type the following:



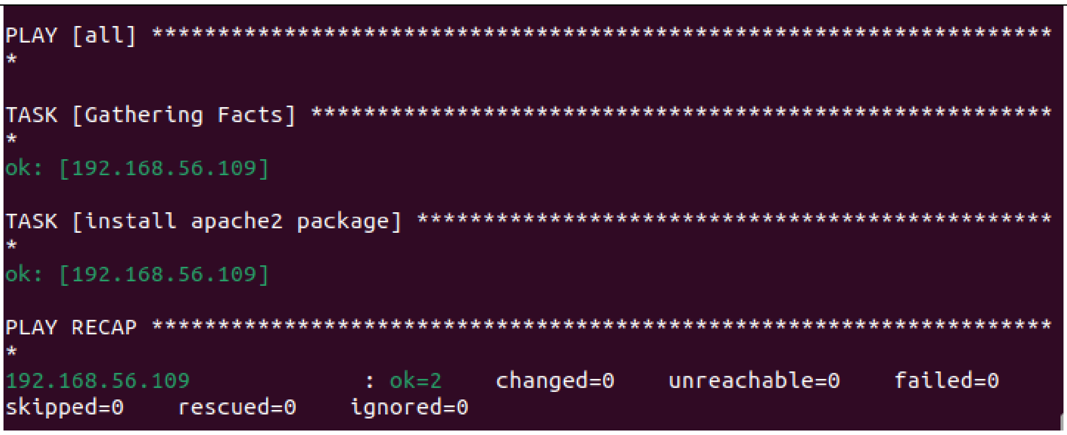
```
  GNU nano 4.8                    install_apache.yml
---
- hosts: all
  become: true
  tasks:

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

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

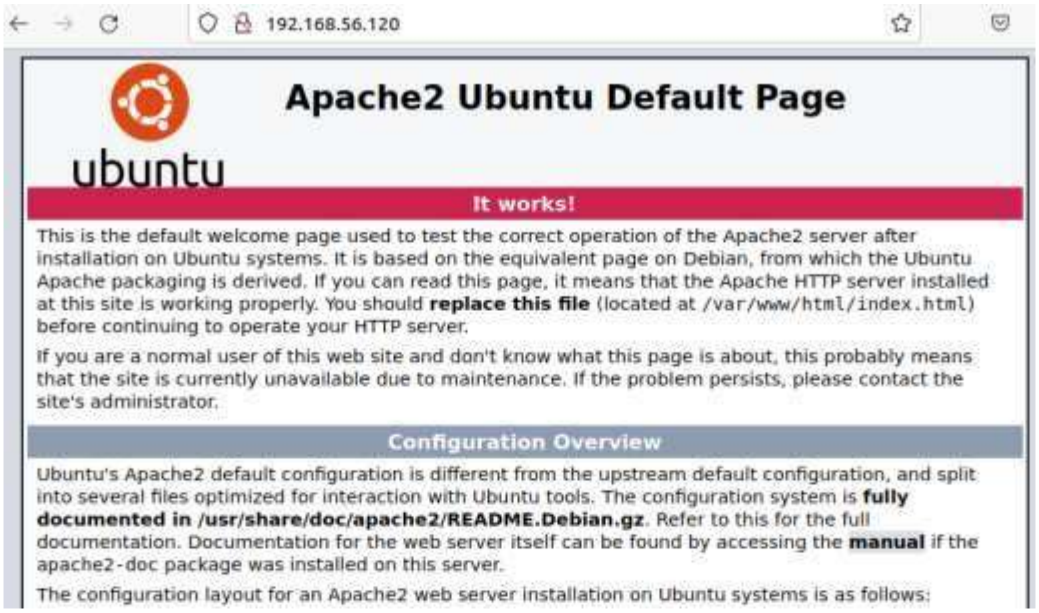
2. Run the yml file using the command: *ansible-playbook --ask-become-pass install_apache.yml.* Describe the result of this command. *This commands shows inside the playbook which is working and it does affect the remote servers.*



```
jozshua@workstation-VirtualBox:~/CPE232_Jozshua/cpe_ansible_Alonzo$ ansible-pla
ybook --ask-become-pass install_apache.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************
*
ok: [192.168.56.108]

TASK [install apache2 package] *************************************************
*
changed: [192.168.56.108]

PLAY RECAP *********************************************************************
*
192.168.56.108             : ok=2    changed=1    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
```


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

TASK [Gathering Facts] ********************************************************
*
ok: [192.168.56.109]

TASK [install apache2 package] ************************************************
*
ok: [192.168.56.109]

PLAY RECAP ********************************************************************
*
192.168.56.109             : ok=2    changed=0    unreachable=0    failed=0    
skipped=0    rescued=0    ignored=0
```

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.



Apache2 Ubuntu Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

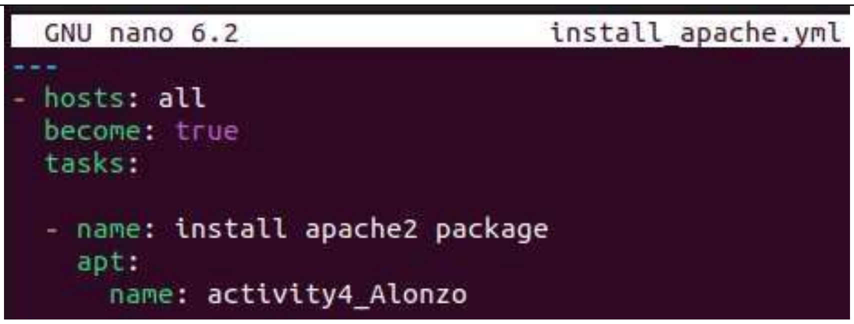
If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

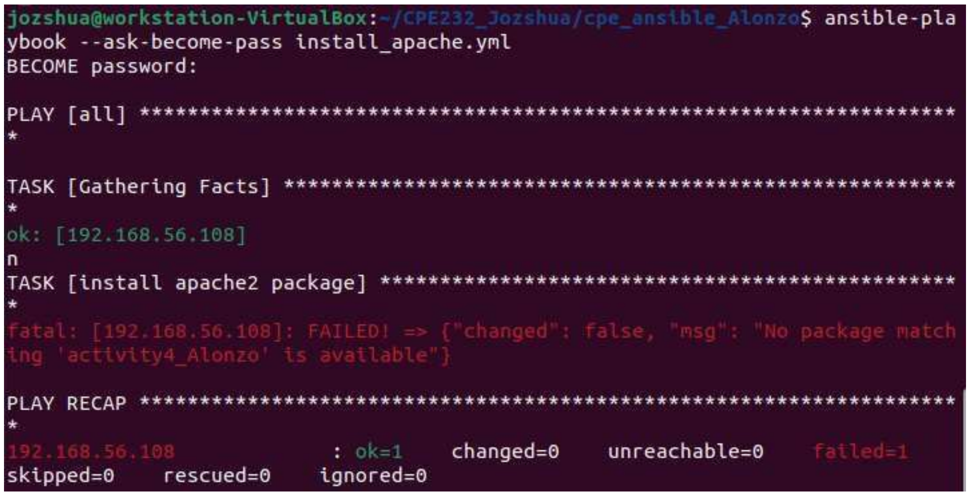
Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the apache2-doc package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

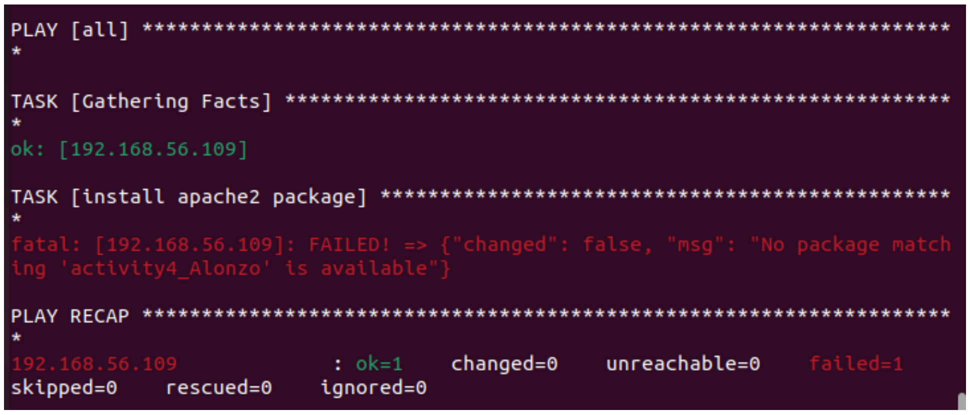
4. Try to edit the *install_apache.yml* and change the name of the package to any name that will not be recognized. What is the output? The output shows no package that is matched from the edited name of the package.



```
  GNU nano 6.2                              install_apache.yml
---
- hosts: all
  become: true
  tasks:

  - name: install apache2 package
    apt:
      name: activity4_Alonzo
```

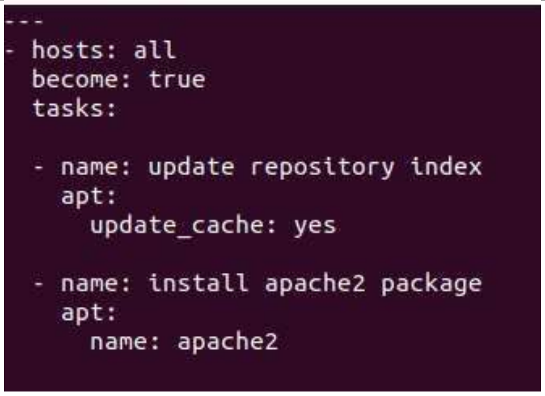


```
jozshua@workstation-VirtualBox:~/CPE232_Jozshua/cpe_ansible_Alonzo$ ansible-pla
ybook --ask-become-pass install_apache.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************
*
ok: [192.168.56.108]
n
TASK [install apache2 package] *************************************************
*
fatal: [192.168.56.108]: FAILED! => {"changed": false, "msg": "No package match
ing 'activity4_Alonzo' is available"}

PLAY RECAP *********************************************************************
*
192.168.56.108             : ok=1    changed=0    unreachable=0    failed=1
skipped=0    rescued=0    ignored=0
```



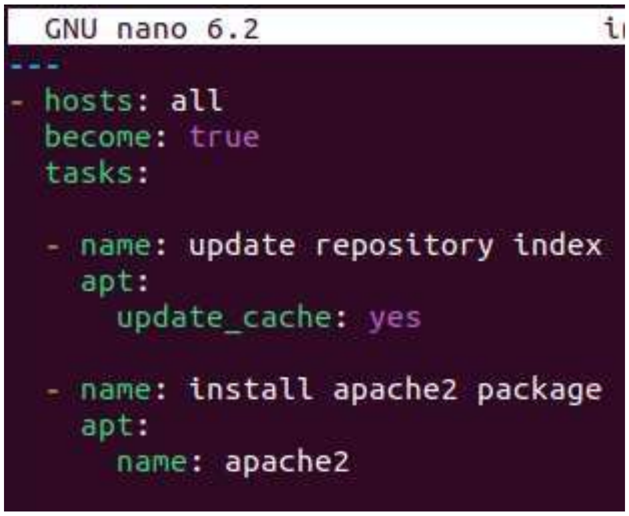
```
PLAY [all] *********************************************************************
*

TASK [Gathering Facts] *********************************************************
*
ok: [192.168.56.109]

TASK [install apache2 package] *************************************************
*
fatal: [192.168.56.109]: FAILED! => {"changed": false, "msg": "No package match
ing 'activity4_Alonzo' is available"}

PLAY RECAP *********************************************************************
*
192.168.56.109             : ok=1    changed=0    unreachable=0    failed=1
skipped=0    rescued=0    ignored=0
```

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
```

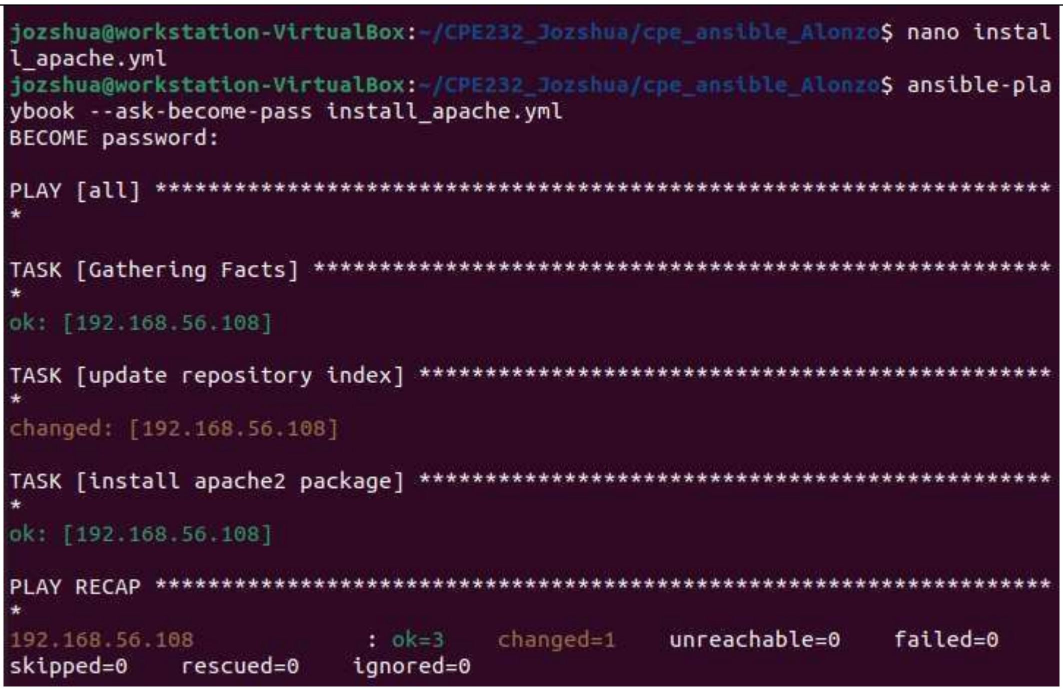
Save the changes to this file and exit.



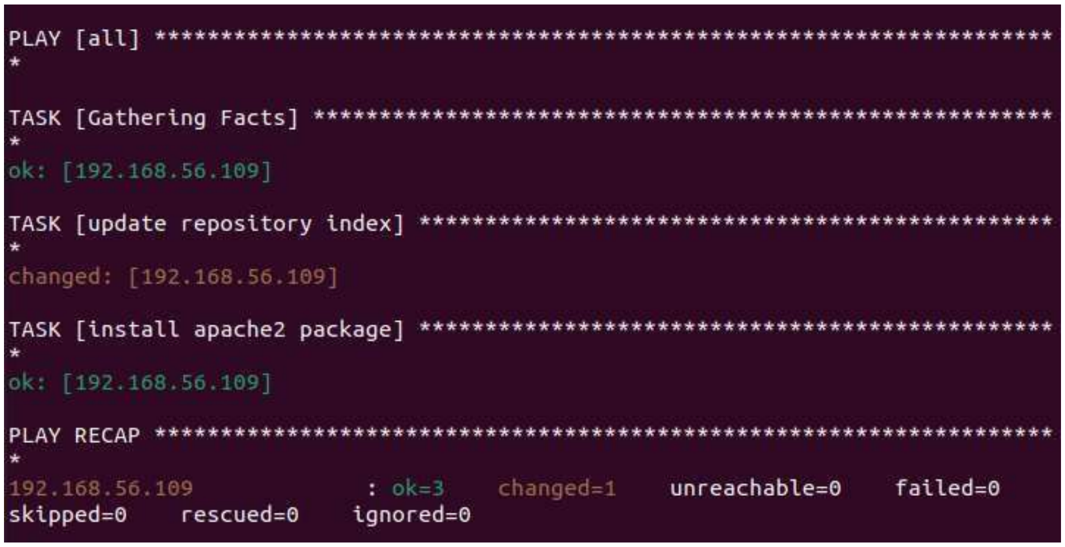
```
  GNU nano 6.2                                  i
---
- hosts: all
  become: true
  tasks:

  - 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?

   The command shows new task display wherein update repository index and below it was indicated that it was changed with its ip address.

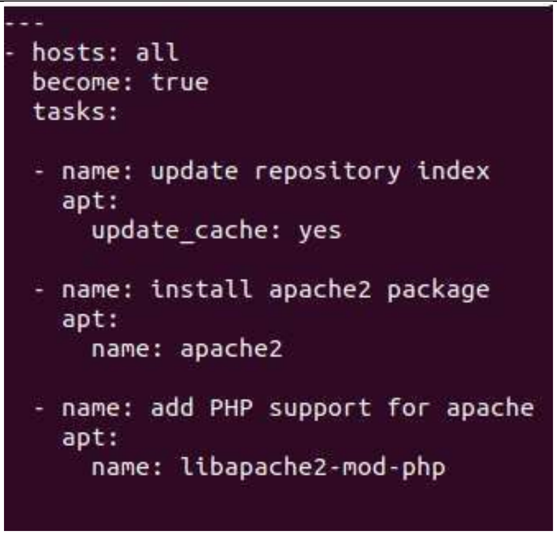


```
jozshua@workstation-VirtualBox:~/CPE232_Jozshua/cpe_ansible_Alonzo$ nano instal
l_apache.yml
jozshua@workstation-VirtualBox:~/CPE232_Jozshua/cpe_ansible_Alonzo$ ansible-pla
ybook --ask-become-pass install_apache.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************
*
ok: [192.168.56.108]

TASK [update repository index] *************************************************
*
changed: [192.168.56.108]

TASK [install apache2 package] *************************************************
*
ok: [192.168.56.108]

PLAY RECAP *********************************************************************
*
192.168.56.108             : ok=3    changed=1    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
```



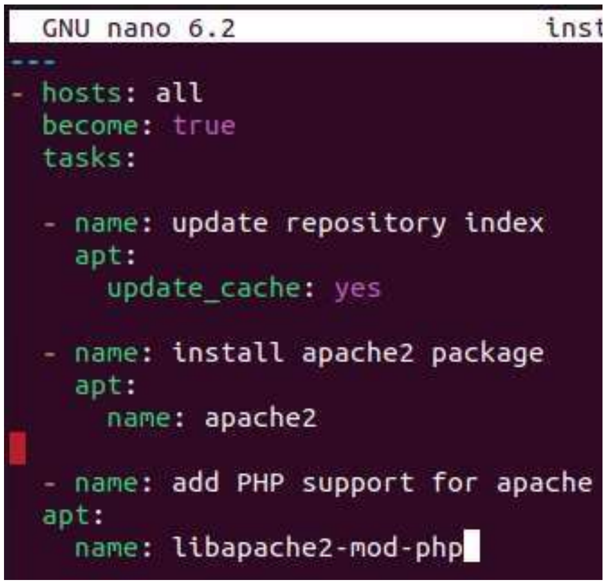
```
PLAY [all] *********************************************************************
*

TASK [Gathering Facts] *********************************************************
*
ok: [192.168.56.109]

TASK [update repository index] *************************************************
*
changed: [192.168.56.109]

TASK [install apache2 package] *************************************************
*
ok: [192.168.56.109]

PLAY RECAP *********************************************************************
*
192.168.56.109             : ok=3    changed=1    unreachable=0    failed=0
skipped=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
```

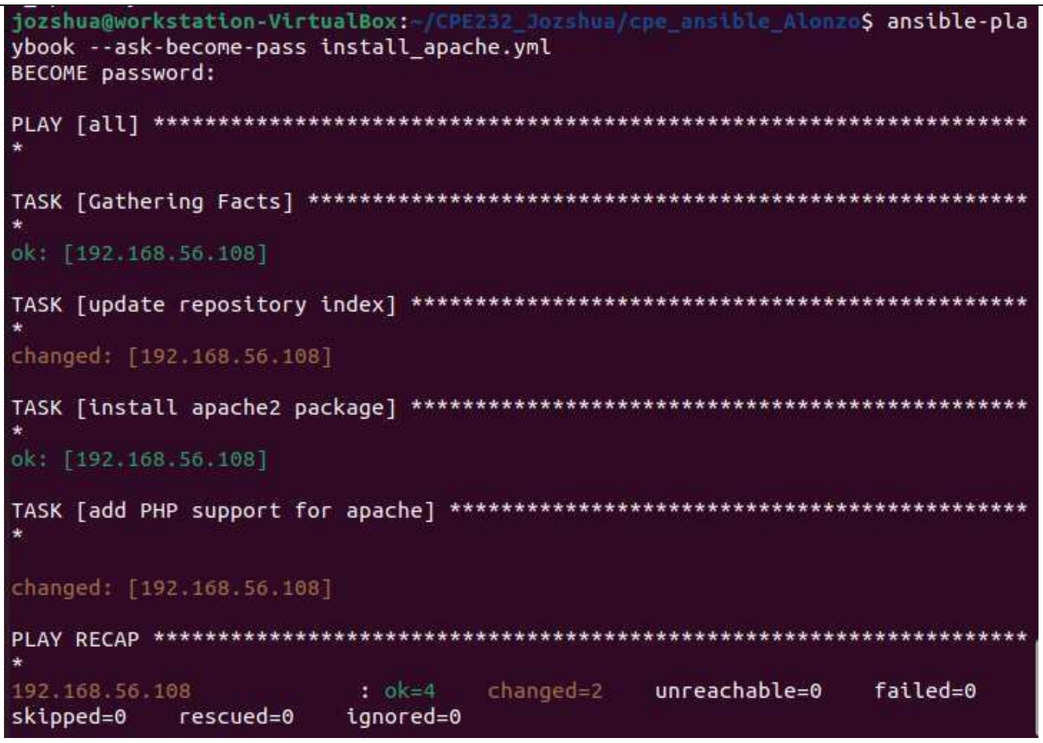
Save the changes to this file and exit.



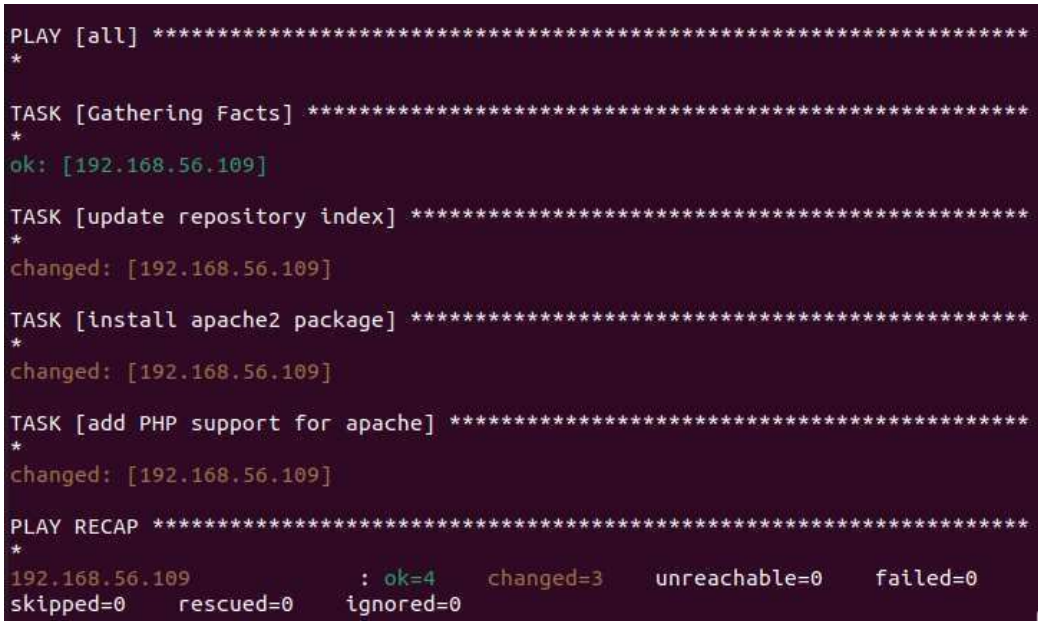
```
  GNU nano 6.2                   inst
---
- hosts: all
  become: true
  tasks:

  - 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?

   The command shows new task display wherein add PHP support for apache2 and below it was indicated that is was changed with its ip address.

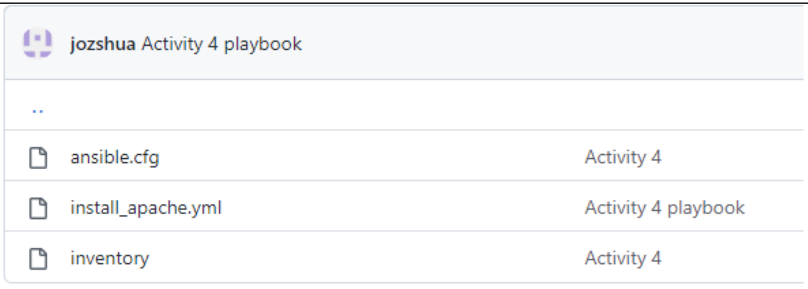


```
jozshua@workstation-VirtualBox:~/CPE232_Jozshua/cpe_ansible_Alonzo$ ansible-pla
ybook --ask-become-pass install_apache.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************
*
ok: [192.168.56.108]

TASK [update repository index] *************************************************
*
changed: [192.168.56.108]

TASK [install apache2 package] *************************************************
*
ok: [192.168.56.108]

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

changed: [192.168.56.108]

PLAY RECAP *********************************************************************
*
192.168.56.108             : ok=4    changed=2    unreachable=0    failed=0    
skipped=0    rescued=0    ignored=0
```



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

TASK [Gathering Facts] *********************************************************
*
ok: [192.168.56.109]

TASK [update repository index] *************************************************
*
changed: [192.168.56.109]

TASK [install apache2 package] *************************************************
*
changed: [192.168.56.109]

TASK [add PHP support for apache] **********************************************
*
changed: [192.168.56.109]

PLAY RECAP *********************************************************************
*
192.168.56.109             : ok=4    changed=3    unreachable=0    failed=0    
skipped=0    rescued=0    ignored=0
```

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



jozshua Activity 4 playbook

..

ansible.cfg	Activity 4
install_apache.yml	Activity 4 playbook
inventory	Activity 4

Reflections:

Answer the following:

1. What is the importance of using a playbook?

   The playbook is important because of instead of manually going through the process each time you start a server, you may use ansible playbooks.

2. Summarize what we have done on this activity.

   To summarize from what I did for this activity first command is I allowed the unauthenticated upgrade. Next is I installed curl to use it for the installation of the pip.py then we get the python3. For the last installation I get the ansible for this activity. After the installation I make cpe_ansible directory to have inventory and ansible.cfg folders. I used ssh-keygen and ssh copy id to ssh login from the servers and to have connection to the local host on every ansible command. I also installed playbook with the install_apache.yml folder to edit every output from the playbook command. Lastly we commited every change to the personal github account.