Name: Clarence B. Chavez	**Date Performed:** Sept. 11, 2024
Course/Section: CPE31S2	**Date Submitted:** Sept. 11, 2024
Instructor: Engr. Robin Valenzuela	**Semester and SY:** 1st Sem, 2024-2025

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:

Provide screenshots for each task.

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.

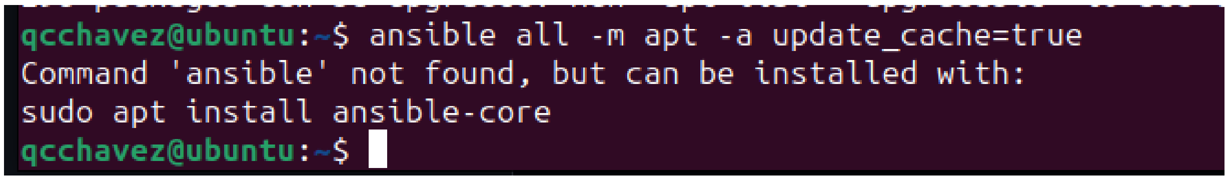
Playbooks record and execute **Ansible**'s configuration, deployment, and orchestration functions. They can describe a policy you want your remote systems to enforce, or a set of steps in a general IT process. If Ansible modules are the tools in your workshop, playbooks are your instruction manuals,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

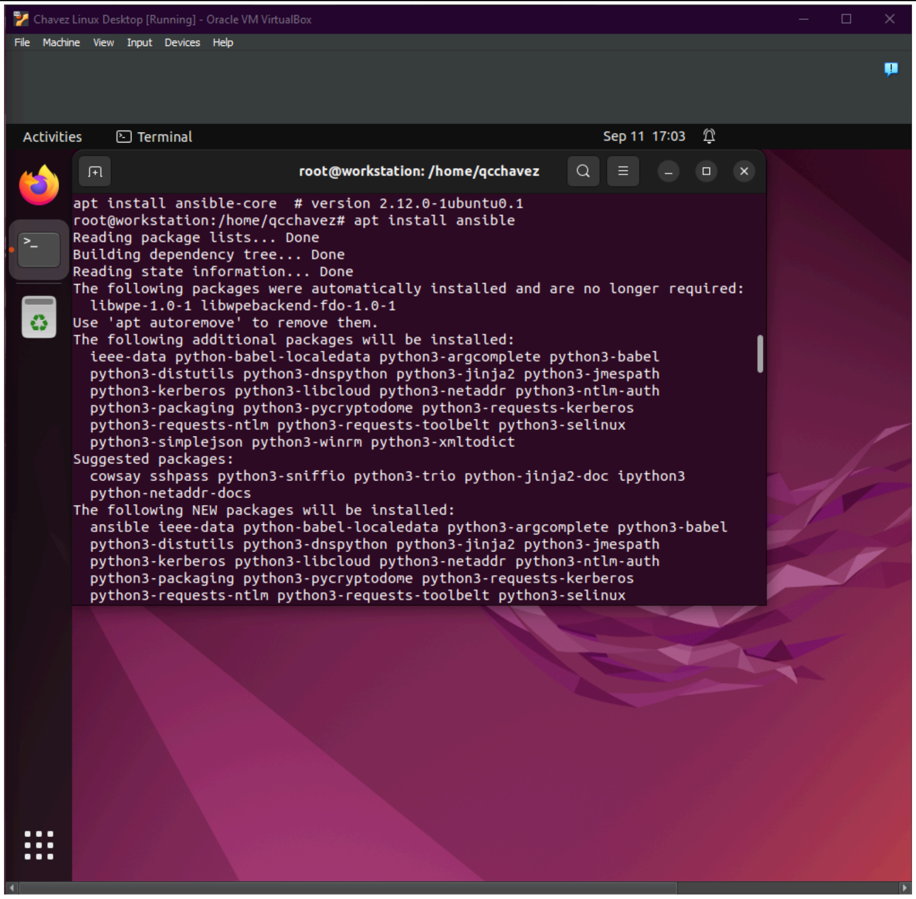
1. Locally, we use the command *sudo apt update* when we want to download package information from all configured resources. The sources often defined in /etc/apt/sources.list file and other files located in /etc/apt/sources.list.d/ directory. So, when you run update command, it downloads the package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

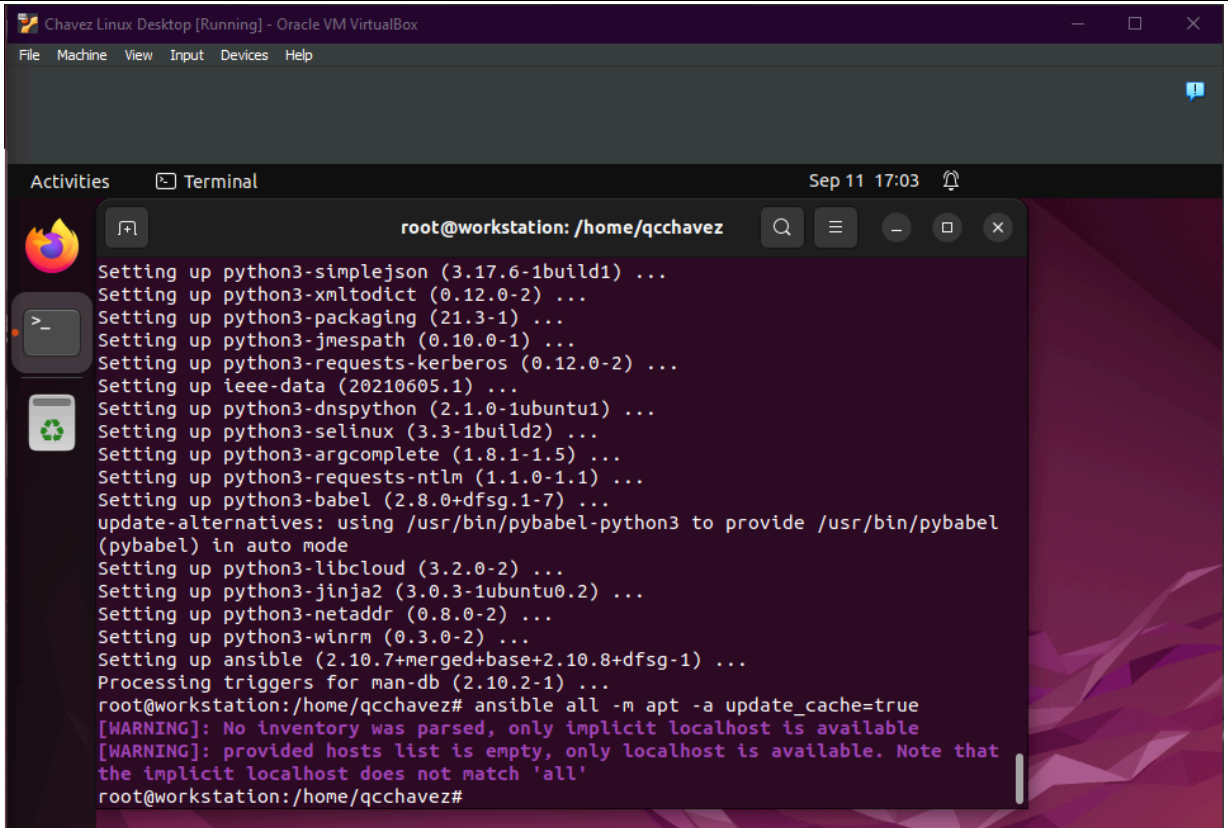
What is the result of the command? Is it successful?



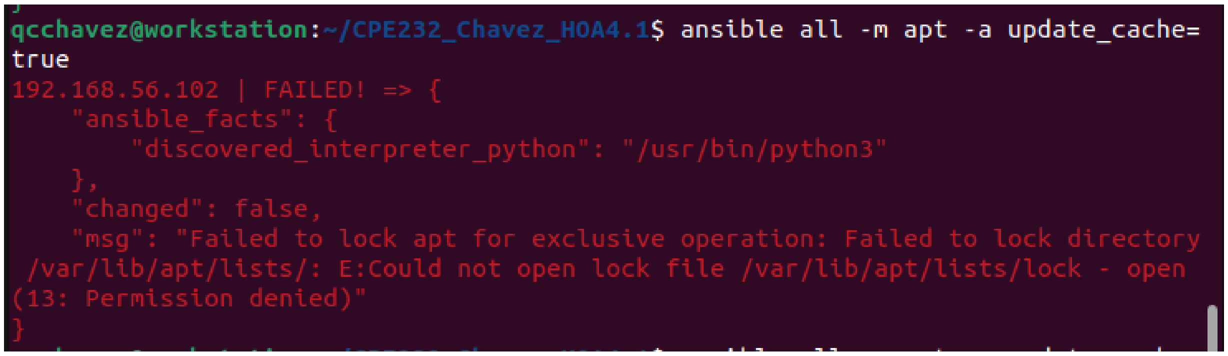
```
qcchavez@ubuntu:~$ ansible all -m apt -a update_cache=true
Command 'ansible' not found, but can be installed with:
sudo apt install ansible-core
qcchavez@ubuntu:~$ 
```


Chavez Linux Desktop [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Terminal
Sep 11 17:03
root@workstation: /home/qcchavez

```
apt install ansible-core  # version 2.12.0-1ubuntu0.1
root@workstation:/home/qcchavez# apt install ansible
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libwpe-1.0-1 libwpebackend-fdo-1.0-1
Use 'apt autoremove' to remove them.
The following additional packages will be installed:
  ieee-data python-babel-localedata python3-argcomplete python3-babel
  python3-distutils python3-dnspython python3-jinja2 python3-jmespath
  python3-kerberos python3-libcloud python3-netaddr python3-ntlm-auth
  python3-packaging python3-pycryptodome python3-requests-kerberos
  python3-requests-ntlm python3-requests-toolbelt python3-selinux
  python3-simplejson python3-winrm python3-xmltodict
Suggested packages:
  cowsay sshpass python3-sniffio python3-trio python-jinja2-doc ipython3
  python-netaddr-docs
The following NEW packages will be installed:
  ansible ieee-data python-babel-localedata python3-argcomplete python3-babel
  python3-distutils python3-dnspython python3-jinja2 python3-jmespath
  python3-kerberos python3-libcloud python3-netaddr python3-ntlm-auth
  python3-packaging python3-pycryptodome python3-requests-kerberos
  python3-requests-ntlm python3-requests-toolbelt python3-selinux
```


Chavez Linux Desktop [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Terminal
Sep 11 17:03
root@workstation: /home/qcchavez

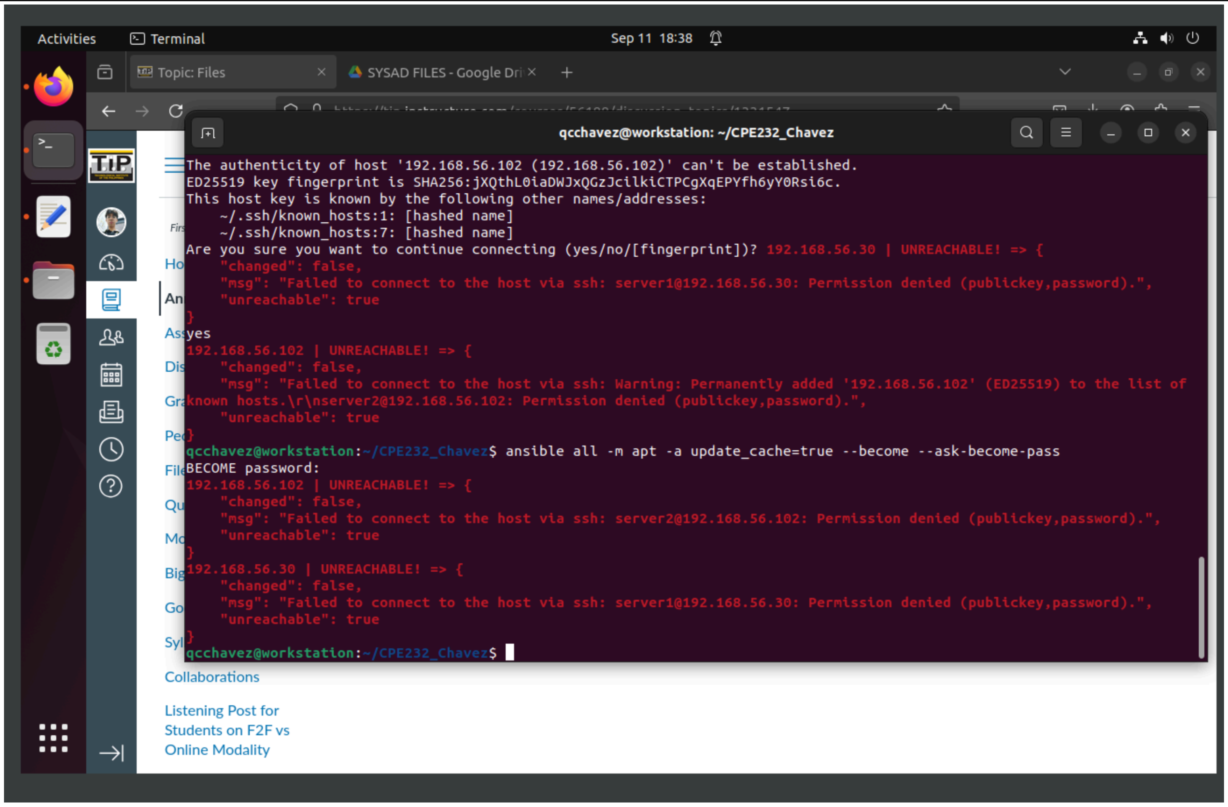
```
Setting up python3-simplejson (3.17.6-1build1) ...
Setting up python3-xmltodict (0.12.0-2) ...
Setting up python3-packaging (21.3-1) ...
Setting up python3-jmespath (0.10.0-1) ...
Setting up python3-requests-kerberos (0.12.0-2) ...
Setting up ieee-data (20210605.1) ...
Setting up python3-dnspython (2.1.0-1ubuntu1) ...
Setting up python3-selinux (3.3-1build2) ...
Setting up python3-argcomplete (1.8.1-1.5) ...
Setting up python3-requests-ntlm (1.1.0-1.1) ...
Setting up python3-babel (2.8.0+dfsg.1-7) ...
update-alternatives: using /usr/bin/pybabel-python3 to provide /usr/bin/pybabel
(pybabel) in auto mode
Setting up python3-libcloud (3.2.0-2) ...
Setting up python3-jinja2 (3.0.3-1ubuntu0.2) ...
Setting up python3-netaddr (0.8.0-2) ...
Setting up python3-winrm (0.3.0-2) ...
Setting up ansible (2.10.7+merged+base+2.10.8+dfsg-1) ...
Processing triggers for man-db (2.10.2-1) ...
root@workstation:/home/qcchavez# ansible all -m apt -a update_cache=true
[WARNING]: No inventory was parsed, only implicit localhost is available
[WARNING]: provided hosts list is empty, only localhost is available. Note that
the implicit localhost does not match 'all'
root@workstation:/home/qcchavez#
```



```
qcchavez@workstation:~/CPE232_Chavez_HOA4.1$ ansible all -m apt -a update_cache=
true
192.168.56.102 | 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)"
}
```

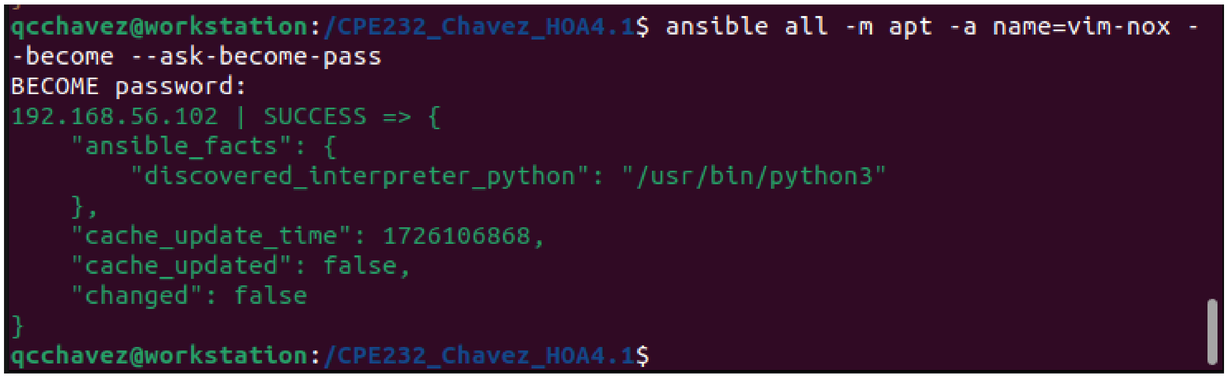
Try editing the command and add something that would elevate the privilege. Issue the command *ansible all -m apt -a update_cache=true --become --ask-become-pass.* Enter the sudo password when prompted. You will notice now that the output of this command is a success. The *update_cache=true* is the same 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.


Activities
Terminal
Sep 11 18:38
Topic: Files
SYSAD FILES - Google Dri
qcchavez@workstation: ~/CPE232_Chavez
Collaborations
Listening Post for Students on F2F vs Online Modality

```
The authenticity of host '192.168.56.102 (192.168.56.102)' can't be established.
ED25519 key fingerprint is SHA256:jXQthL0iaDWJxQGzJcilkiCTPCgXqEPYfh6yY0Rsi6c.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:1: [hashed name]
    ~/.ssh/known_hosts:7: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? 192.168.56.30 | UNREACHABLE! => {
    "changed": false,
    "msg": "Failed to connect to the host via ssh: server1@192.168.56.30: Permission denied (publickey,password).",
    "unreachable": true
}
yes
192.168.56.102 | UNREACHABLE! => {
    "changed": false,
    "msg": "Failed to connect to the host via ssh: Warning: Permanently added '192.168.56.102' (ED25519) to the list of known hosts.\r\nserver2@192.168.56.102: Permission denied (publickey,password).",
    "unreachable": true
}
qcchavez@workstation:~/CPE232_Chavez$ ansible all -m apt -a update_cache=true --become --ask-become-pass
BECOME password:
192.168.56.102 | UNREACHABLE! => {
    "changed": false,
    "msg": "Failed to connect to the host via ssh: server2@192.168.56.102: Permission denied (publickey,password).",
    "unreachable": true
}
192.168.56.30 | UNREACHABLE! => {
    "changed": false,
    "msg": "Failed to connect to the host via ssh: server1@192.168.56.30: Permission denied (publickey,password).",
    "unreachable": true
}
qcchavez@workstation:~/CPE232_Chavez$
```

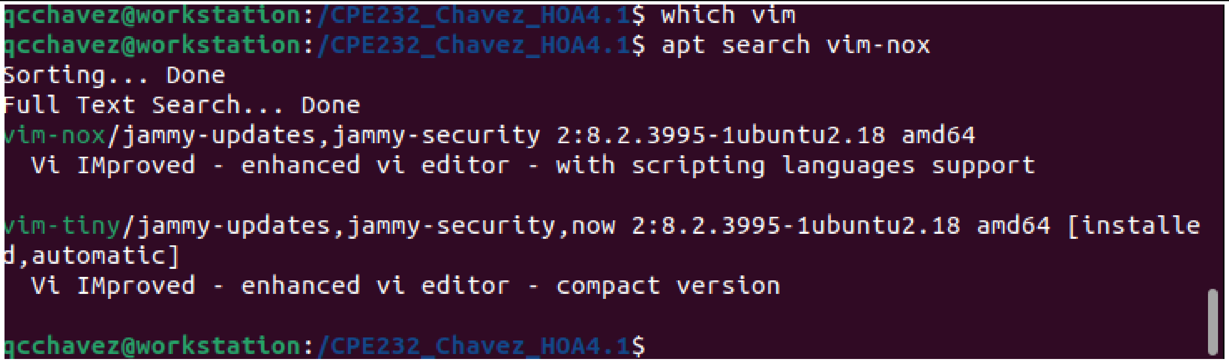
2. Let's try to install VIM, which is an almost compatible version of the UNIX editor Vi. To do this, we will just change the module part in 1.1 instruction. Here is the command: *ansible all -m apt -a name=vim-nox --become --ask-become-pass.* The command would take some time after typing the password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



```
qcchavez@workstation:/CPE232_Chavez_HOA4.1$ ansible all -m apt -a name=vim-nox -
-become --ask-become-pass
BECOME password:
192.168.56.102 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "cache_update_time": 1726106868,
    "cache_updated": false,
    "changed": false
}
qcchavez@workstation:/CPE232_Chavez_HOA4.1$
```

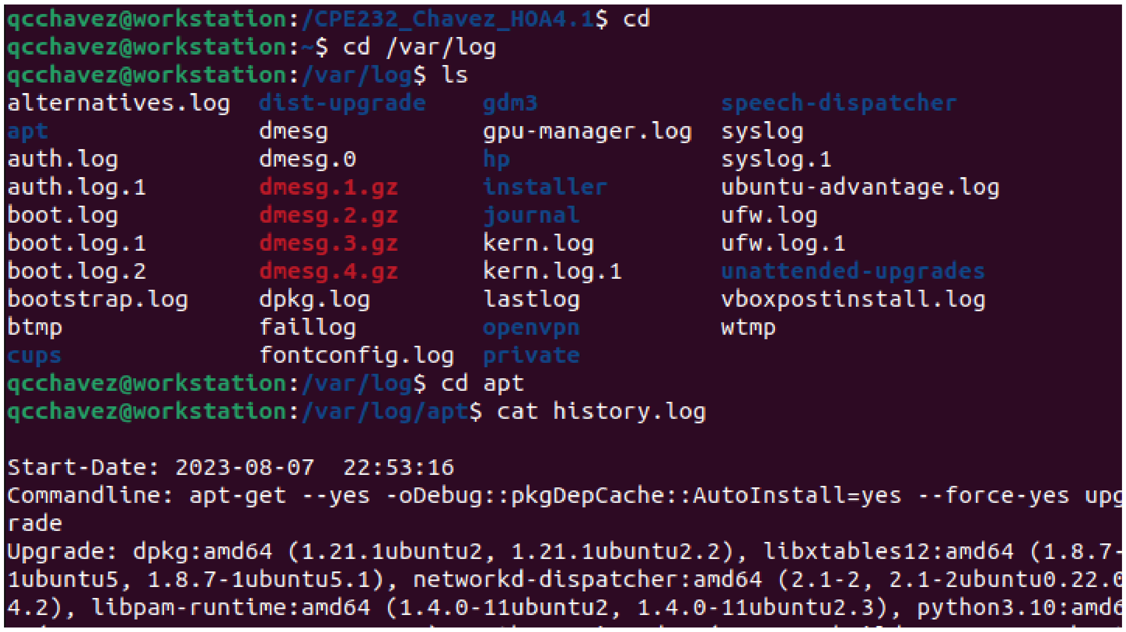
2.1 Verify that you have installed the package in the remote servers. Issue the command *which vim* and the command *apt search vim-nox* respectively. Was the command successful?

- The command **apt search vim-nox** was successful.



```
qcchavez@workstation:/CPE232_Chavez_HOA4.1$ which vim
qcchavez@workstation:/CPE232_Chavez_HOA4.1$ apt search vim-nox
Sorting... Done
Full Text Search... Done
vim-nox/jammy-updates,jammy-security 2:8.2.3995-1ubuntu2.18 amd64
  Vi IMproved - enhanced vi editor - with scripting languages support

vim-tiny/jammy-updates,jammy-security,now 2:8.2.3995-1ubuntu2.18 amd64 [installe
d,automatic]
  Vi IMproved - enhanced vi editor - compact version

qcchavez@workstation:/CPE232_Chavez_HOA4.1$
```

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.

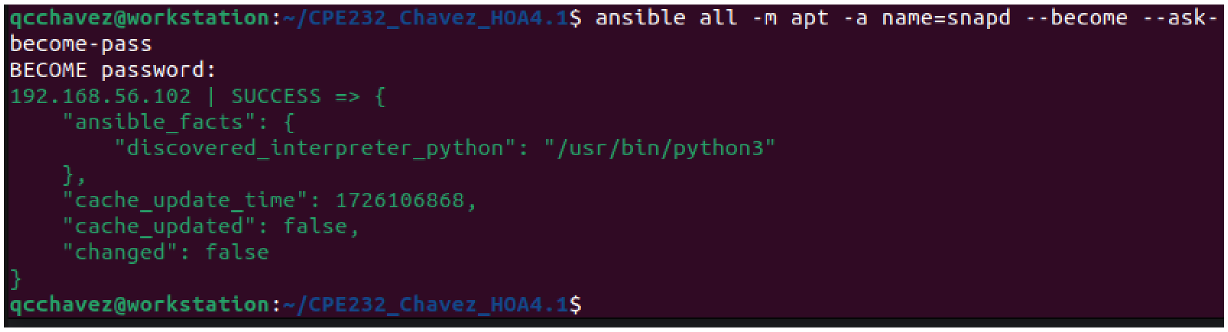


```
qcchavez@workstation:/CPE232_Chavez_HOA4.1$ cd
qcchavez@workstation:~$ cd /var/log
qcchavez@workstation:/var/log$ ls
alternatives.log  dist-upgrade    gdm3              speech-dispatcher
apt               dmesg           gpu-manager.log   syslog
auth.log          dmesg.0         hp                syslog.1
auth.log.1        dmesg.1.gz      installer         ubuntu-advantage.log
boot.log          dmesg.2.gz      journal           ufw.log
boot.log.1        dmesg.3.gz      kern.log          ufw.log.1
boot.log.2        dmesg.4.gz      kern.log.1        unattended-upgrades
bootstrap.log     dpkg.log        lastlog           vboxpostinstall.log
btmp              faillog         openvpn           wtmp
cups              fontconfig.log  private
qcchavez@workstation:/var/log$ cd apt
qcchavez@workstation:/var/log/apt$ cat history.log

Start-Date: 2023-08-07  22:53:16
Commandline: apt-get --yes -oDebug::pkgDepCache::AutoInstall=yes --force-yes upg
rade
Upgrade: dpkg:amd64 (1.21.1ubuntu2, 1.21.1ubuntu2.2), libxtables12:amd64 (1.8.7-
1ubuntu5, 1.8.7-1ubuntu5.1), networkd-dispatcher:amd64 (2.1-2, 2.1-2ubuntu0.22.0
4.2), libpam-runtime:amd64 (1.4.0-11ubuntu2, 1.4.0-11ubuntu2.3), python3.10:amd6
```

- In the **history.log**, I can see the **starting dates of the commands**, and **what were the upgrades** done to the linux system prompted by the user.

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.

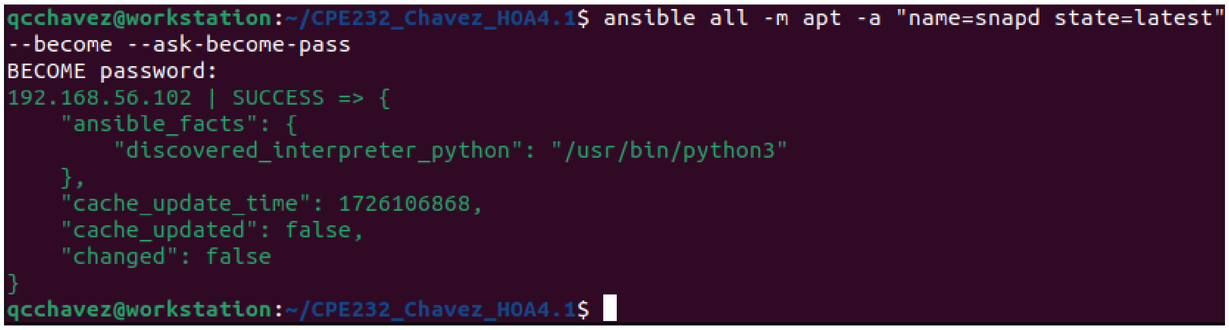
   3.1 Issue the command: *ansible all -m apt -a name=snapd --become --ask-become-pass*
   Can you describe the result of this command? Is it a success? Did it change anything in the remote servers?



```
qcchavez@workstation:~/CPE232_Chavez_HOA4.1$ ansible all -m apt -a name=snapd --become --ask-become-pass
BECOME password:
192.168.56.102 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "cache_update_time": 1726106868,
    "cache_updated": false,
    "changed": false
}
qcchavez@workstation:~/CPE232_Chavez_HOA4.1$
```

- By prompting the command, it was a successful process, and it has also installed **snapd package** in the remote server.

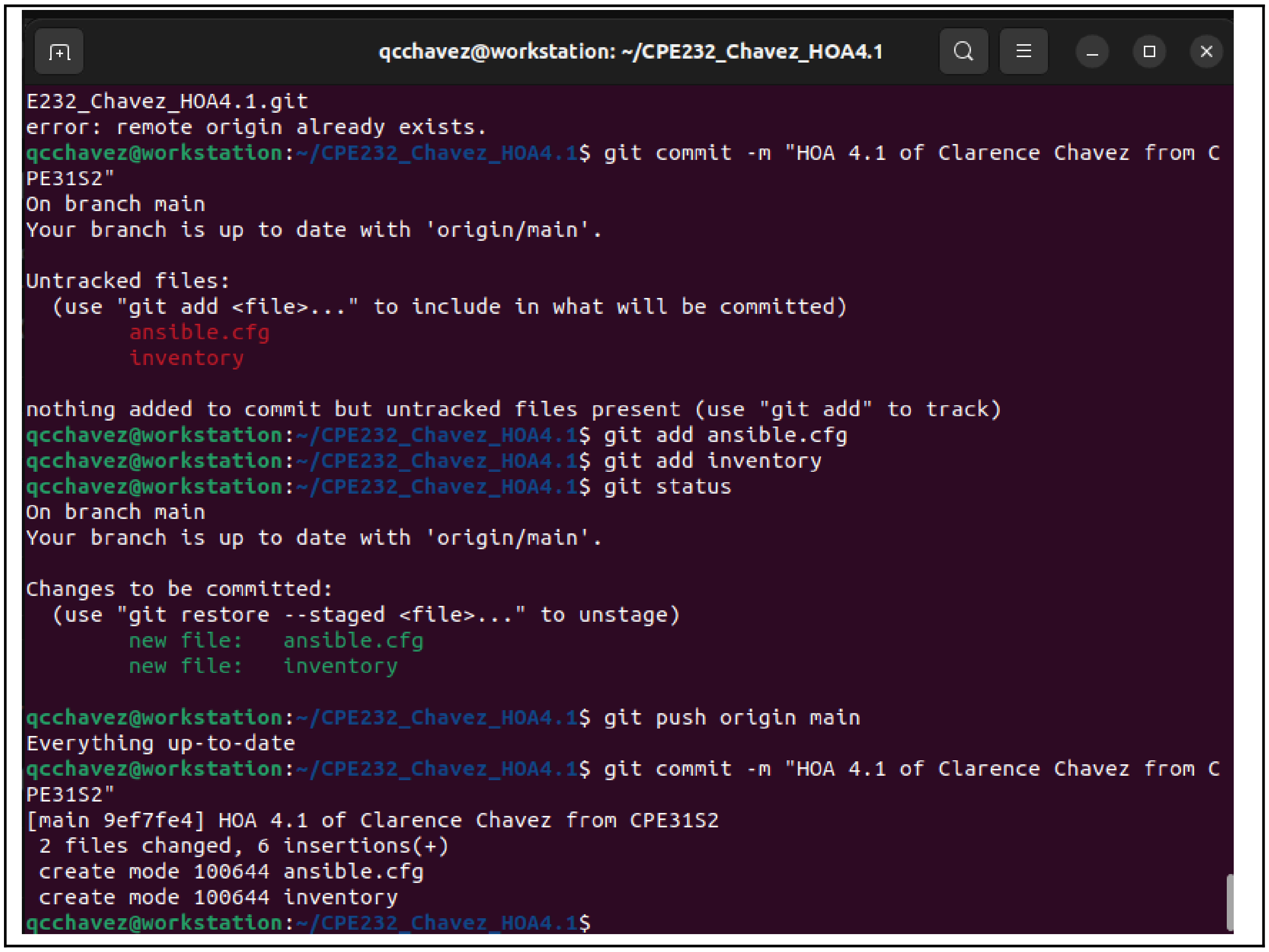
   3.2 Now, try to issue this command: *ansible all -m apt -a "name=snapd state=latest" --become --ask-become-pass*
   Describe the output of this command. Notice how we added the command *state=latest* and placed them in double quotations.



```
qcchavez@workstation:~/CPE232_Chavez_HOA4.1$ ansible all -m apt -a "name=snapd state=latest"
--become --ask-become-pass
BECOME password:
192.168.56.102 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "cache_update_time": 1726106868,
    "cache_updated": false,
    "changed": false
}
qcchavez@workstation:~/CPE232_Chavez_HOA4.1$
```

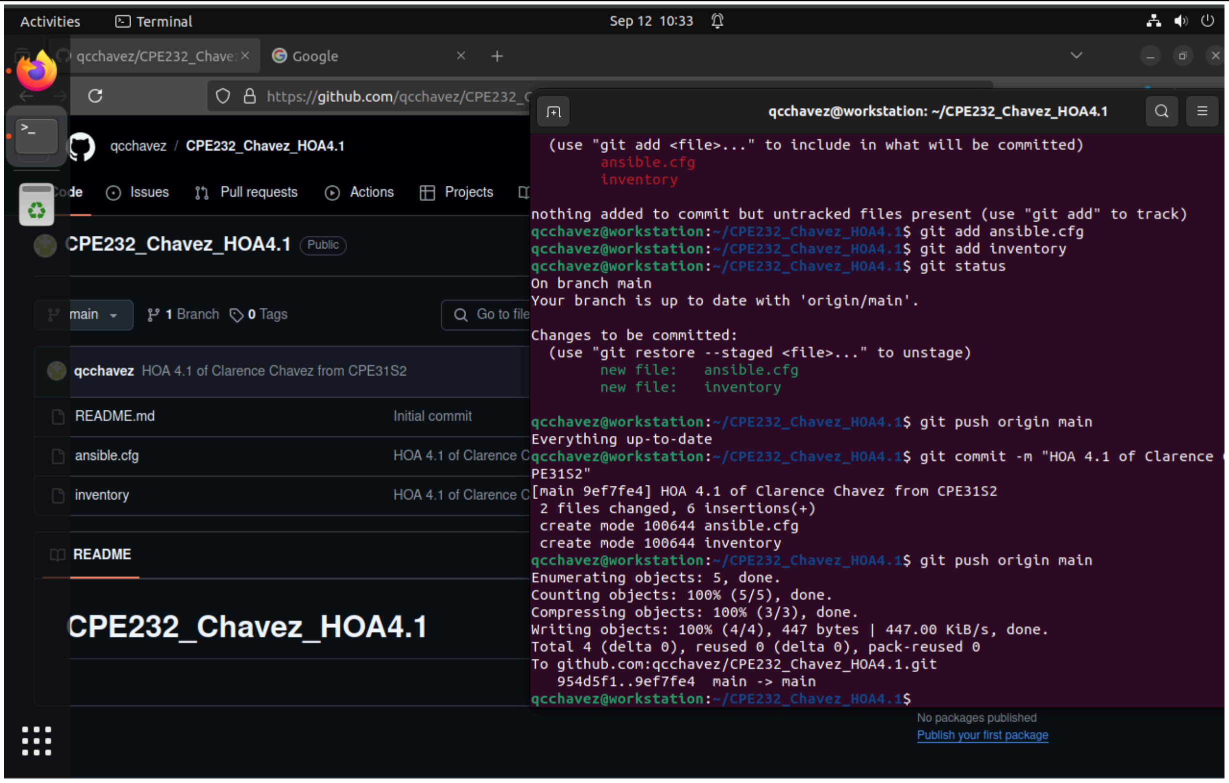
- By issuing the command, it was a successful process and I've noticed that the command **state=latest** checks the snapd package if it has the latest version installed in the Linux Ubuntu OS.

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



qcchavez@workstation: ~/CPE232_Chavez_HOA4.1

```
E232_Chavez_HOA4.1.git
error: remote origin already exists.
qcchavez@workstation:~/CPE232_Chavez_HOA4.1$ git commit -m "HOA 4.1 of Clarence Chavez from C
PE31S2"
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)
        ansible.cfg
        inventory

nothing added to commit but untracked files present (use "git add" to track)
qcchavez@workstation:~/CPE232_Chavez_HOA4.1$ git add ansible.cfg
qcchavez@workstation:~/CPE232_Chavez_HOA4.1$ git add inventory
qcchavez@workstation:~/CPE232_Chavez_HOA4.1$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   ansible.cfg
        new file:   inventory

qcchavez@workstation:~/CPE232_Chavez_HOA4.1$ git push origin main
Everything up-to-date
qcchavez@workstation:~/CPE232_Chavez_HOA4.1$ git commit -m "HOA 4.1 of Clarence Chavez from C
PE31S2"
[main 9ef7fe4] HOA 4.1 of Clarence Chavez from CPE31S2
 2 files changed, 6 insertions(+)
 create mode 100644 ansible.cfg
 create mode 100644 inventory
qcchavez@workstation:~/CPE232_Chavez_HOA4.1$
```

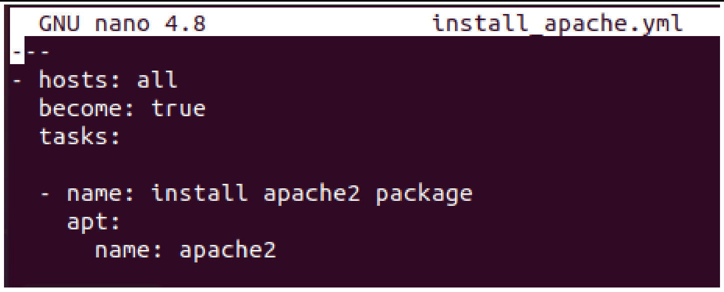


```
  (use "git add <file>..." to include in what will be committed)
        ansible.cfg
        inventory

nothing added to commit but untracked files present (use "git add" to track)
qcchavez@workstation:~/CPE232_Chavez_HOA4.1$ git add ansible.cfg
qcchavez@workstation:~/CPE232_Chavez_HOA4.1$ git add inventory
qcchavez@workstation:~/CPE232_Chavez_HOA4.1$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   ansible.cfg
        new file:   inventory

qcchavez@workstation:~/CPE232_Chavez_HOA4.1$ git push origin main
Everything up-to-date
qcchavez@workstation:~/CPE232_Chavez_HOA4.1$ git commit -m "HOA 4.1 of Clarence C
PE31S2"
[main 9ef7fe4] HOA 4.1 of Clarence Chavez from CPE31S2
 2 files changed, 6 insertions(+)
 create mode 100644 ansible.cfg
 create mode 100644 inventory
qcchavez@workstation:~/CPE232_Chavez_HOA4.1$ git push origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 447 bytes | 447.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:qcchavez/CPE232_Chavez_HOA4.1.git
   954d5f1..9ef7fe4  main -> main
qcchavez@workstation:~/CPE232_Chavez_HOA4.1$
```

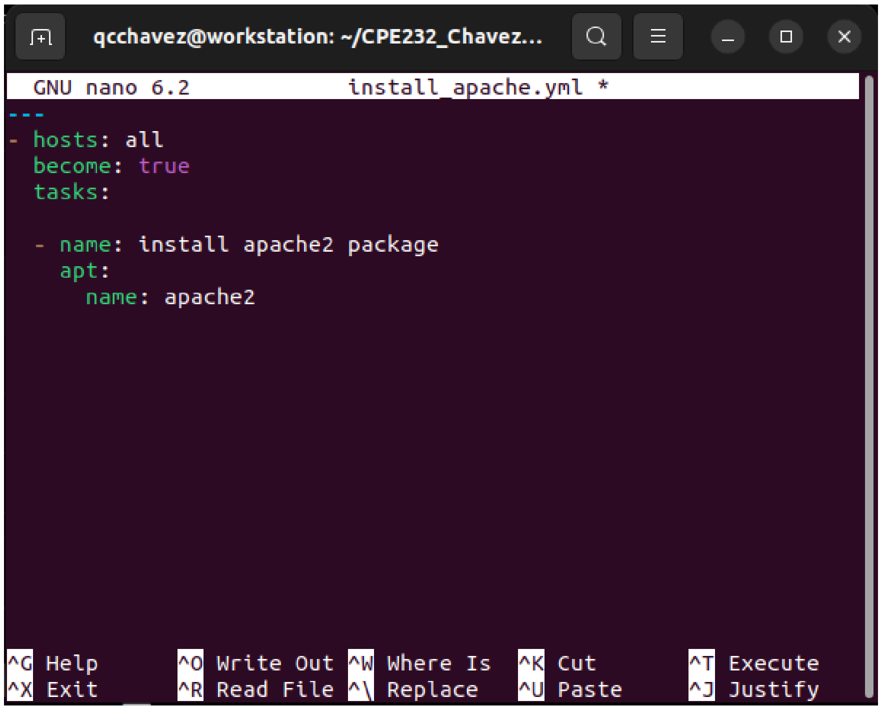
Task 2: Writing our First Playbook

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:



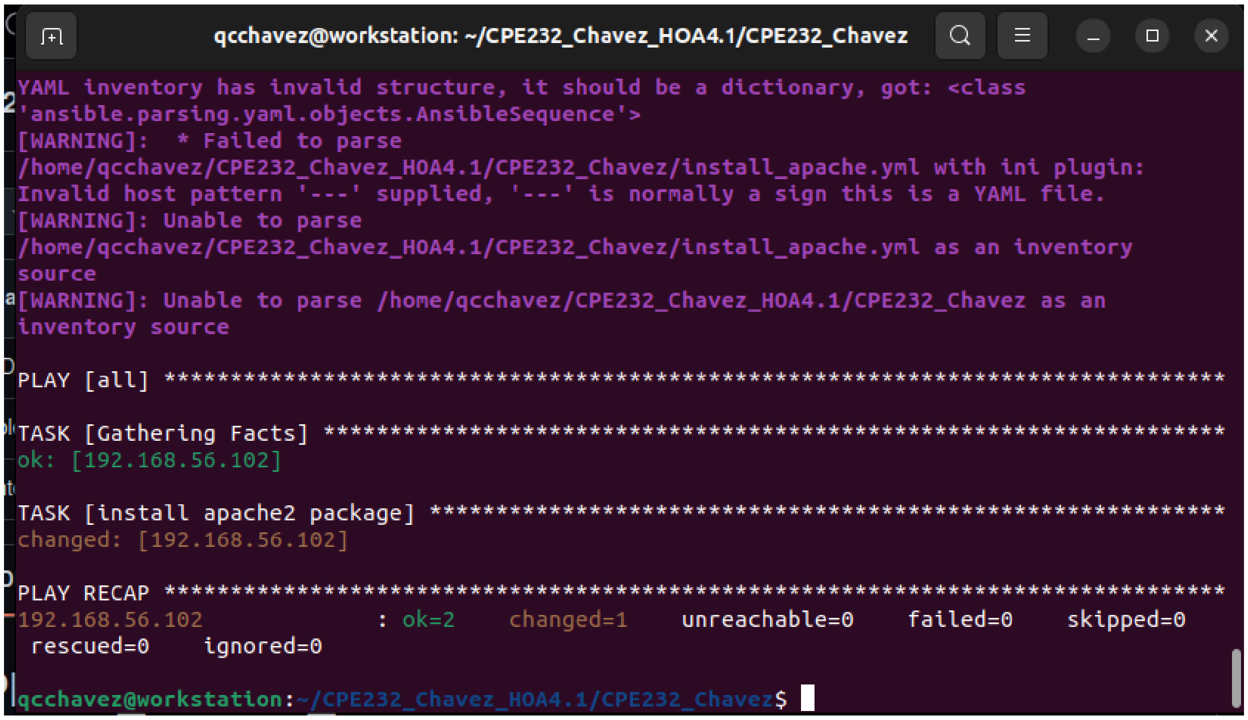
```
GNU nano 4.8                          install_apache.yml
---
- hosts: all
  become: true
  tasks:

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



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

  - name: install apache2 package
    apt:
      name: apache2













^G Help        ^O Write Out ^W Where Is  ^K Cut         ^T Execute
^X Exit        ^R Read File ^\ Replace   ^U Paste       ^J Justify
```

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

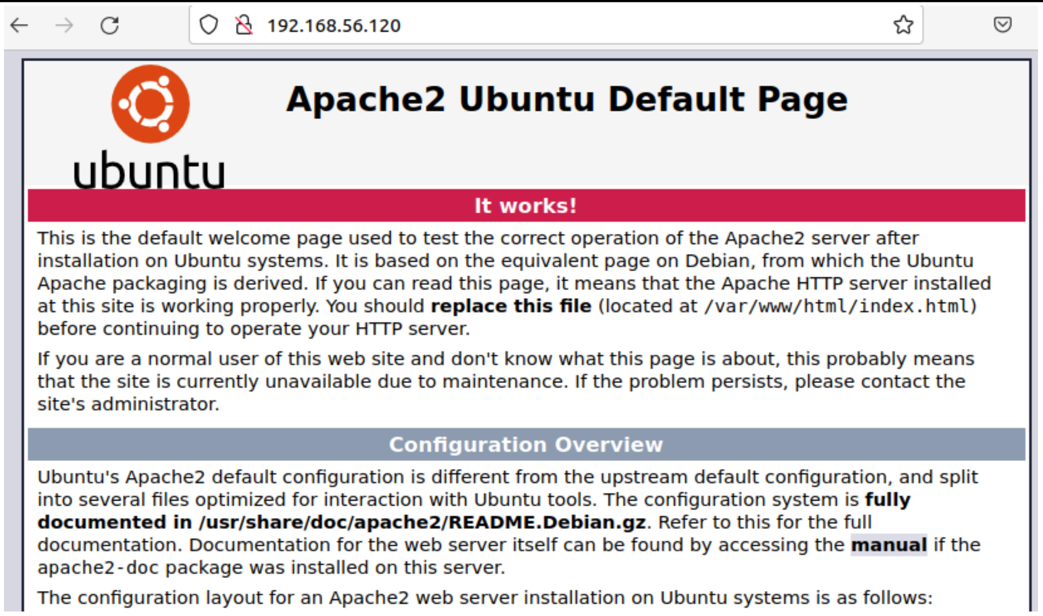
2. Run the yml file using the command: *ansible-playbook --ask-become-pass install_apache.yml*. Describe the result of this command.
   - *The result of the command means that the apache2 package has been altered or changed in some ways and also has applied these changes to the remote server.*



qcchavez@workstation: ~/CPE232_Chavez_HOA4.1/CPE232_Chavez

```
YAML inventory has invalid structure, it should be a dictionary, got: <class
'ansible.parsing.yaml.objects.AnsibleSequence'>
[WARNING]:  * Failed to parse
/home/qcchavez/CPE232_Chavez_HOA4.1/CPE232_Chavez/install_apache.yml with ini plugin:
Invalid host pattern '---' supplied, '---' is normally a sign this is a YAML file.
[WARNING]: Unable to parse
/home/qcchavez/CPE232_Chavez_HOA4.1/CPE232_Chavez/install_apache.yml as an inventory
source
[WARNING]: Unable to parse /home/qcchavez/CPE232_Chavez_HOA4.1/CPE232_Chavez as an
inventory source

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

TASK [Gathering Facts] *************************************************************
ok: [192.168.56.102]

TASK [install apache2 package] *****************************************************
changed: [192.168.56.102]

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

qcchavez@workstation:~/CPE232_Chavez_HOA4.1/CPE232_Chavez$
```

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.



192.168.56.120

Apache2 Ubuntu Default Page

It works!

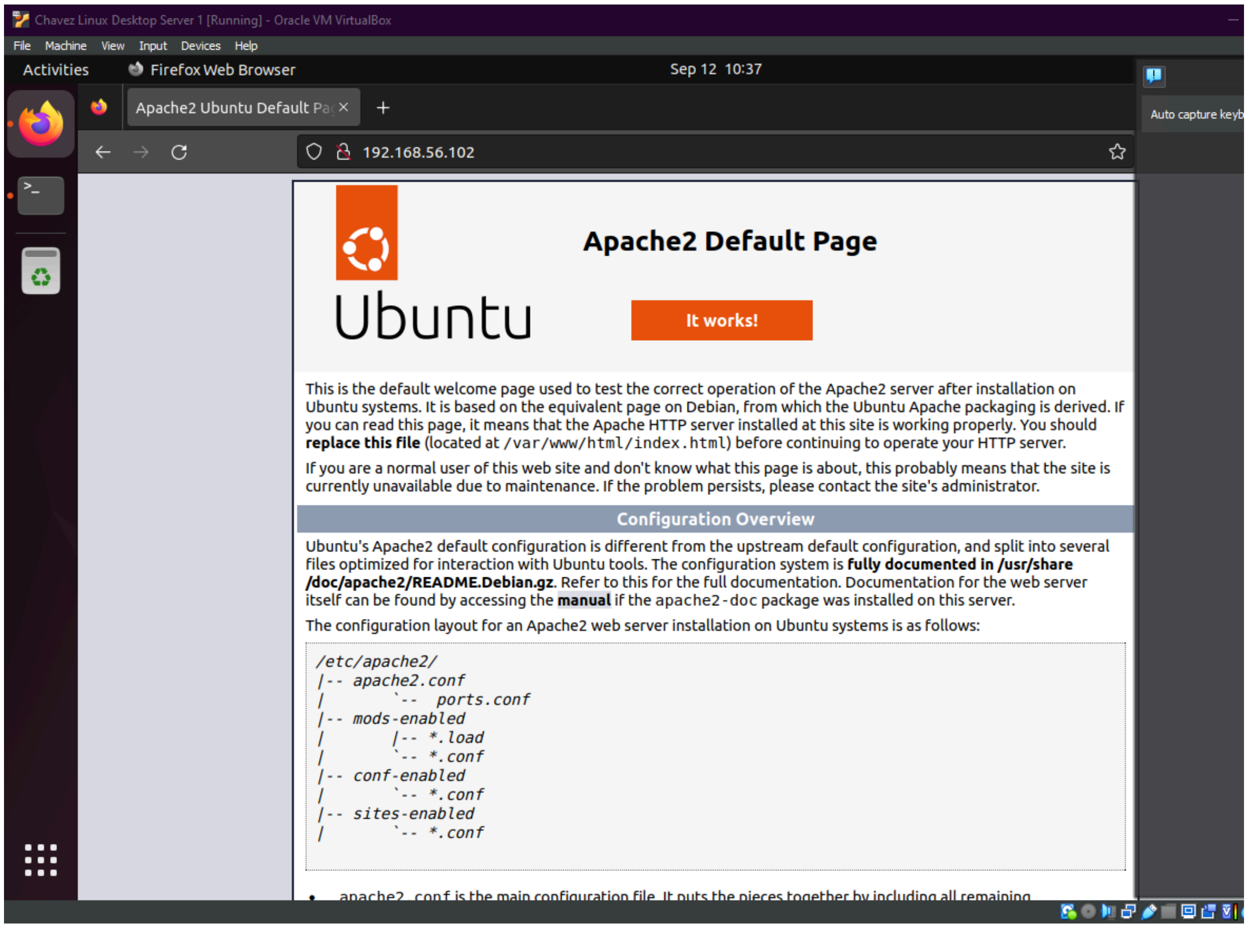
This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at 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:


Chavez Linux Desktop Server 1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Firefox Web Browser
Sep 12 10:37
Apache2 Ubuntu Default Pa
Auto capture keyb

192.168.56.102

Apache2 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

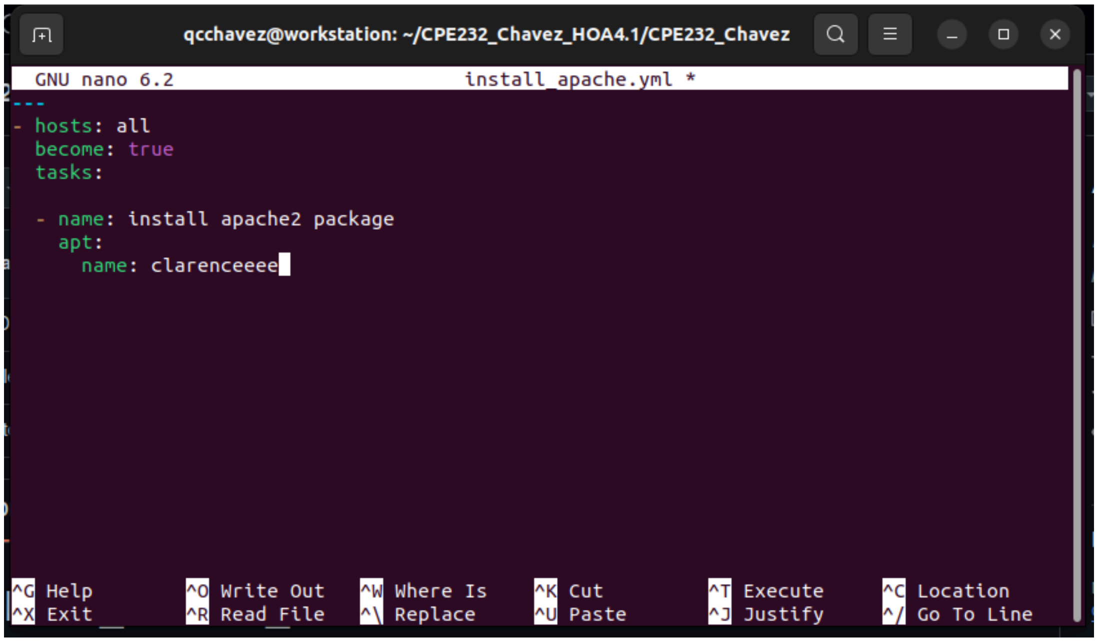
Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. 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:

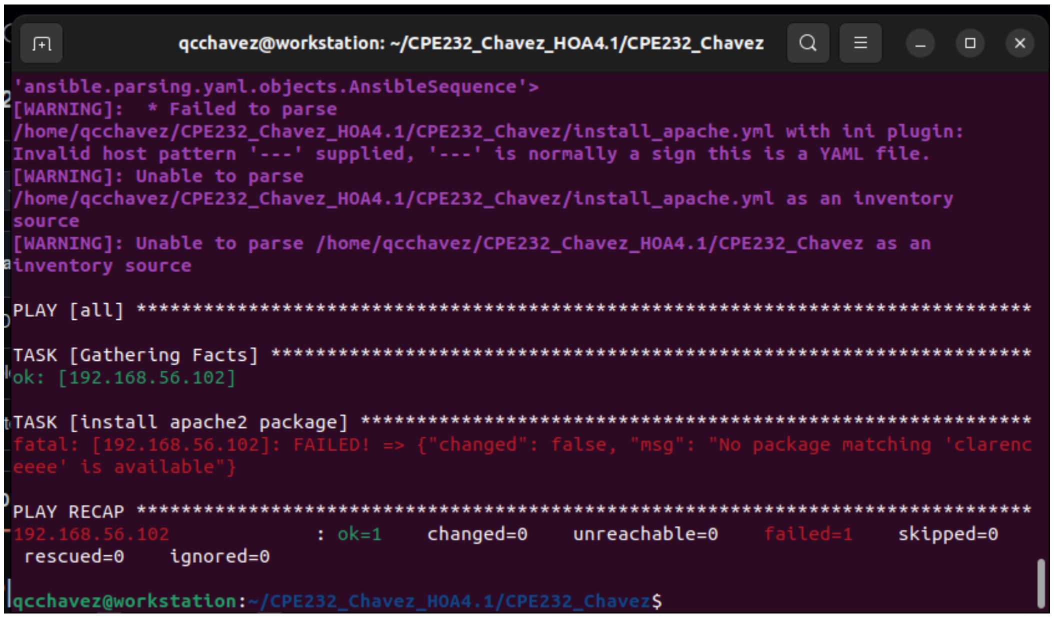
```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```

- apache2.conf is the main configuration file. It puts the pieces together by including all remaining

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?



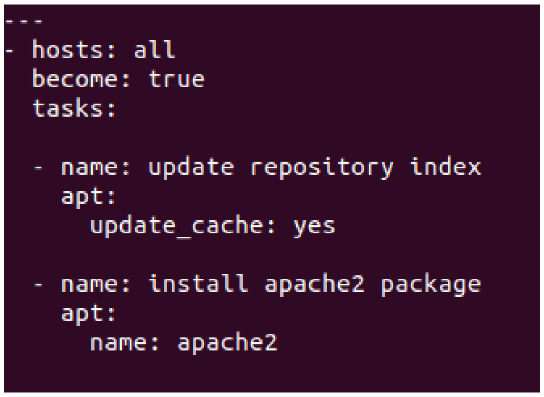
```
qcchavez@workstation: ~/CPE232_Chavez_HOA4.1/CPE232_Chavez
  GNU nano 6.2                          install_apache.yml *
---
- hosts: all
  become: true
  tasks:

  - name: install apache2 package
    apt:
      name: clarenceeee

^G Help          ^O Write Out     ^W Where Is      ^K Cut           ^T Execute       ^C Location
^X Exit          ^R Read File     ^\ Replace       ^U Paste         ^J Justify       ^/ Go To Line
```



```
qcchavez@workstation: ~/CPE232_Chavez_HOA4.1/CPE232_Chavez
'ansible.parsing.yaml.objects.AnsibleSequence'>
[WARNING]:  * Failed to parse
/home/qcchavez/CPE232_Chavez_HOA4.1/CPE232_Chavez/install_apache.yml with ini plugin:
Invalid host pattern '---' supplied, '---' is normally a sign this is a YAML file.
[WARNING]: Unable to parse
/home/qcchavez/CPE232_Chavez_HOA4.1/CPE232_Chavez/install_apache.yml as an inventory
source
[WARNING]: Unable to parse /home/qcchavez/CPE232_Chavez_HOA4.1/CPE232_Chavez as an
inventory source

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

TASK [Gathering Facts] *****************************************************************
ok: [192.168.56.102]

TASK [install apache2 package] *********************************************************
fatal: [192.168.56.102]: FAILED! => {"changed": false, "msg": "No package matching 'clarenc
eeee' is available"}

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

qcchavez@workstation:~/CPE232_Chavez_HOA4.1/CPE232_Chavez$
```

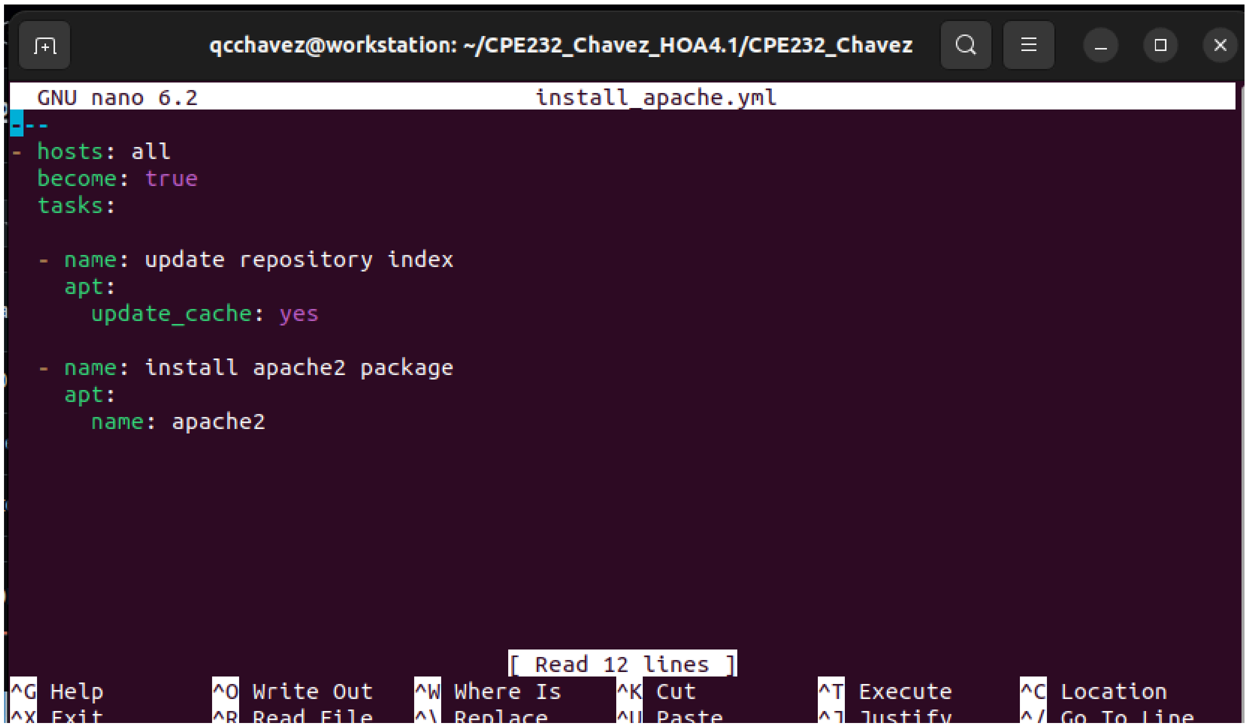
- The error states that the state changed from “**true**” to “**false**” since no “**clarenceeee**” package exists.

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.



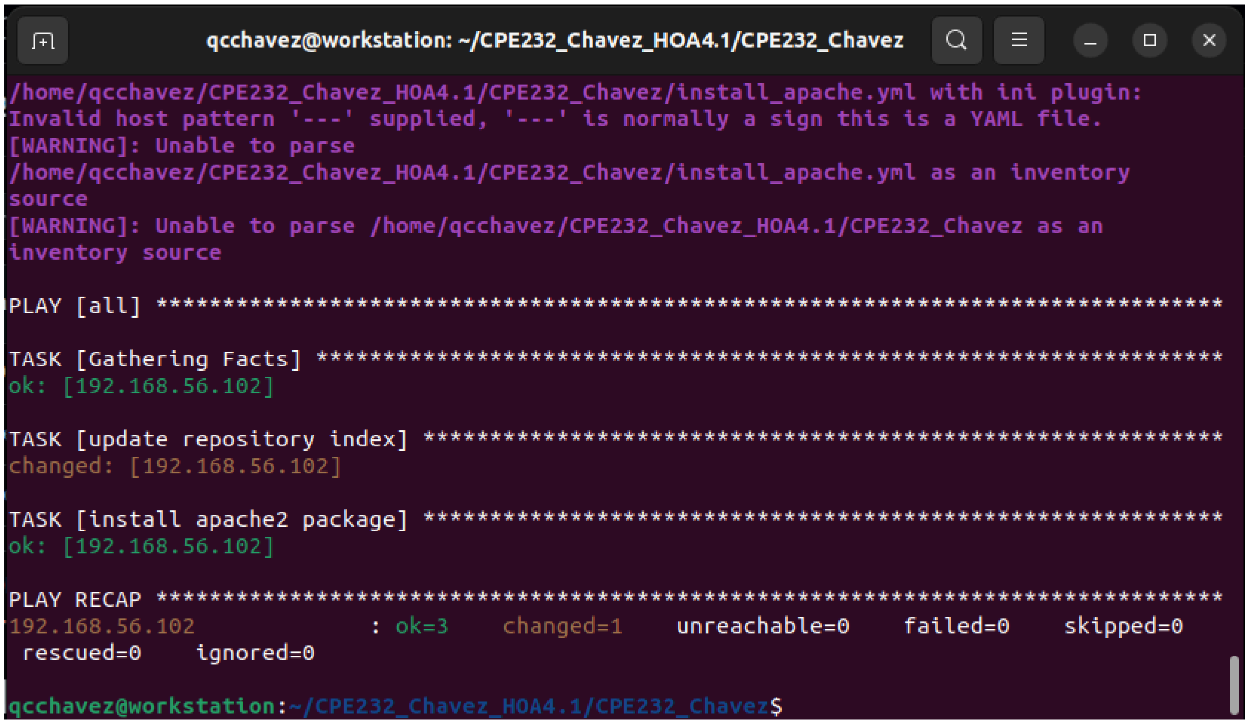
```
---
- hosts: all
  become: true
  tasks:

  - name: update repository index
    apt:
      update_cache: yes

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

Save the changes to this file and exit.



```
qcchavez@workstation: ~/CPE232_Chavez_HOA4.1/CPE232_Chavez
  GNU nano 6.2                          install_apache.yml
---
- hosts: all
  become: true
  tasks:

  - name: update repository index
    apt:
      update_cache: yes

  - name: install apache2 package
    apt:
      name: apache2





                              [ Read 12 lines ]
^G Help         ^O Write Out    ^W Where Is     ^K Cut          ^T Execute      ^C Location
^X Exit         ^R Read File    ^\ Replace      ^U Paste        ^J Justify      ^/ Go To Line
```

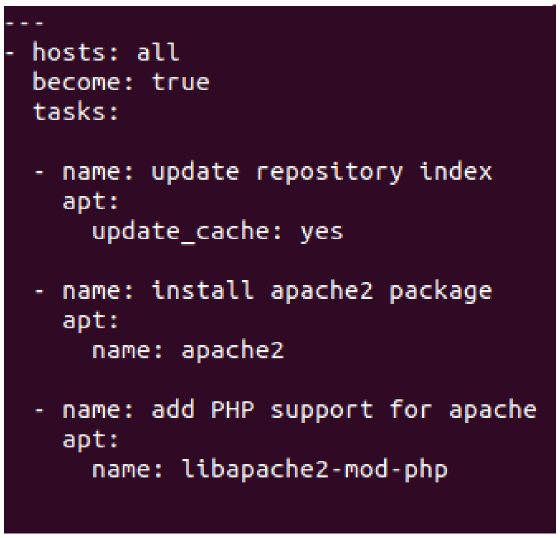
6. Run the playbook and describe the output. Did the new command change anything on the remote servers?



```
/home/qcchavez/CPE232_Chavez_HOA4.1/CPE232_Chavez/install_apache.yml with ini plugin:
Invalid host pattern '---' supplied, '---' is normally a sign this is a YAML file.
[WARNING]: Unable to parse
/home/qcchavez/CPE232_Chavez_HOA4.1/CPE232_Chavez/install_apache.yml as an inventory
source
[WARNING]: Unable to parse /home/qcchavez/CPE232_Chavez_HOA4.1/CPE232_Chavez as an
inventory source

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

TASK [Gathering Facts] ***************************************************************
ok: [192.168.56.102]

TASK [update repository index] *******************************************************
changed: [192.168.56.102]

TASK [install apache2 package] *******************************************************
ok: [192.168.56.102]

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

qcchavez@workstation:~/CPE232_Chavez_HOA4.1/CPE232_Chavez$
```

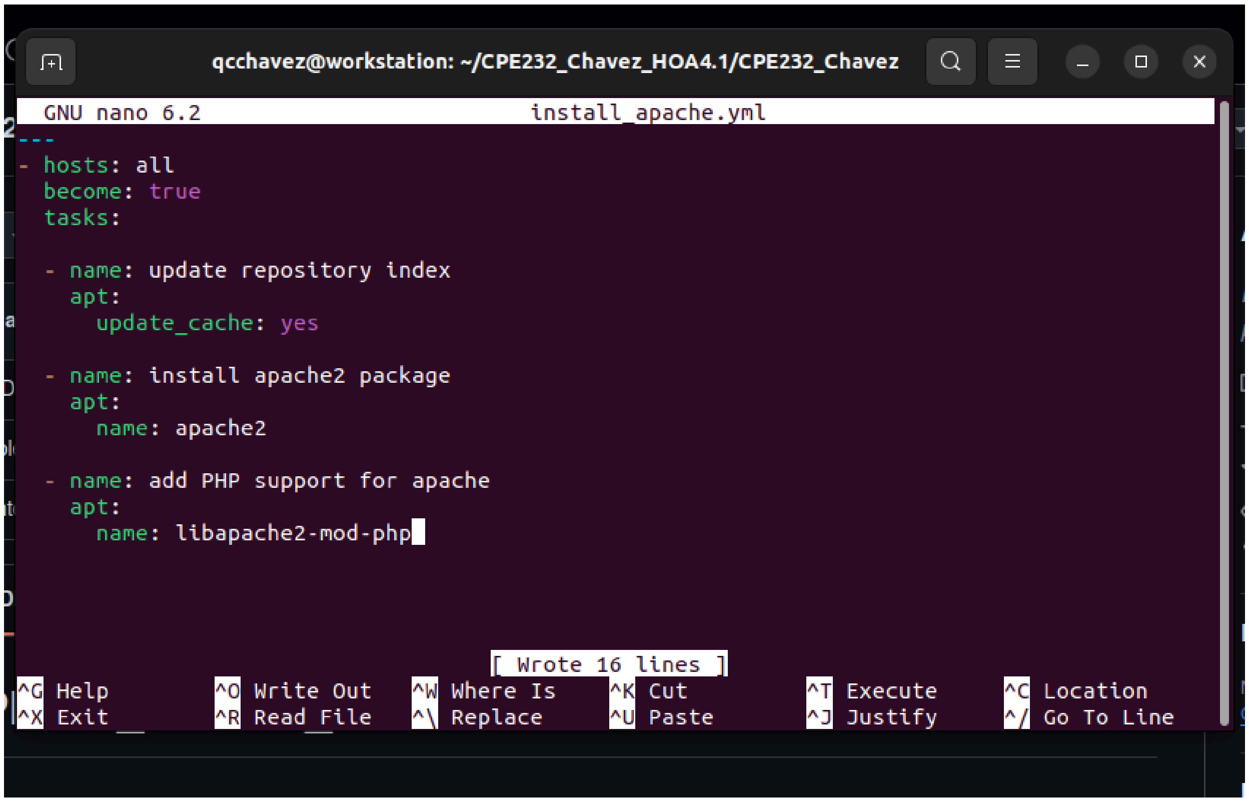
- Based on the screenshot and on the output, the new command did change something to the remote servers. It updated the repository index, and it installed also the apache2 package.

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



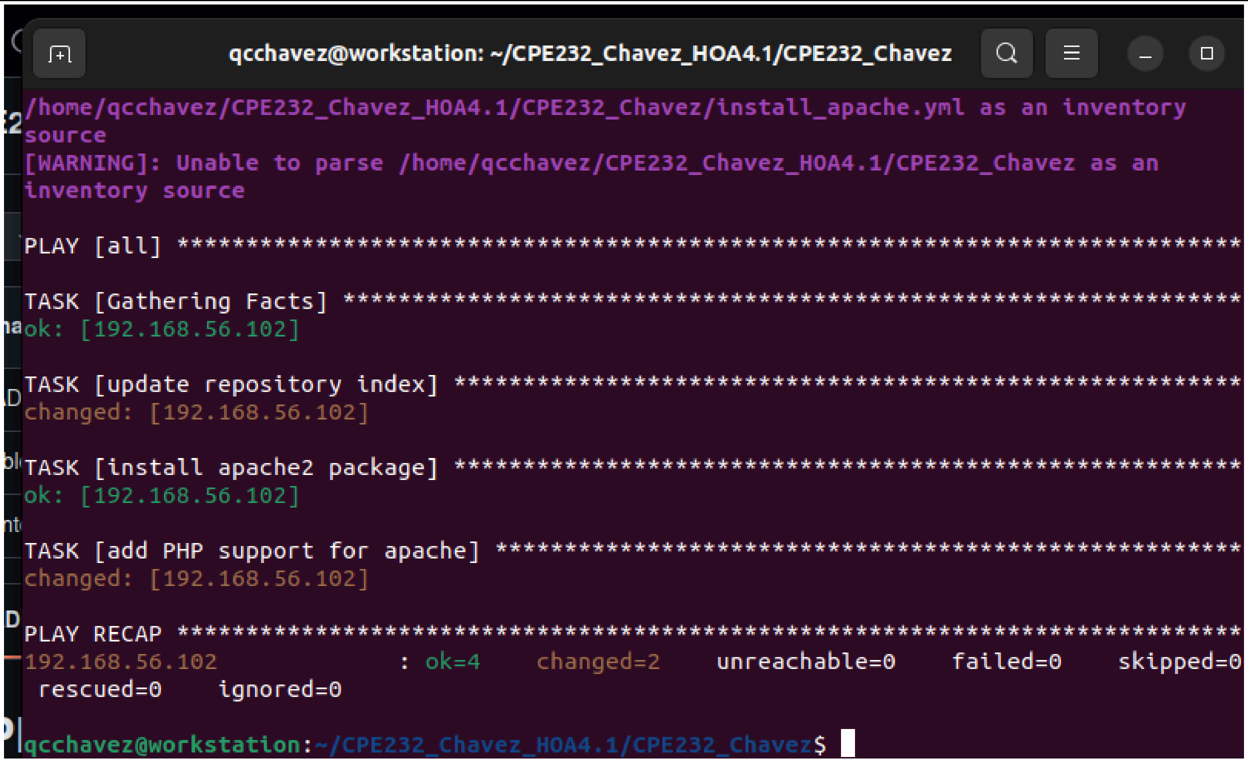
```
---
- hosts: all
  become: true
  tasks:

  - 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                     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



                              [ Wrote 16 lines ]
^G Help        ^O Write Out   ^W Where Is    ^K Cut         ^T Execute     ^C Location
^X Exit        ^R Read File   ^\ Replace     ^U Paste       ^J Justify     ^/ Go To Line
```

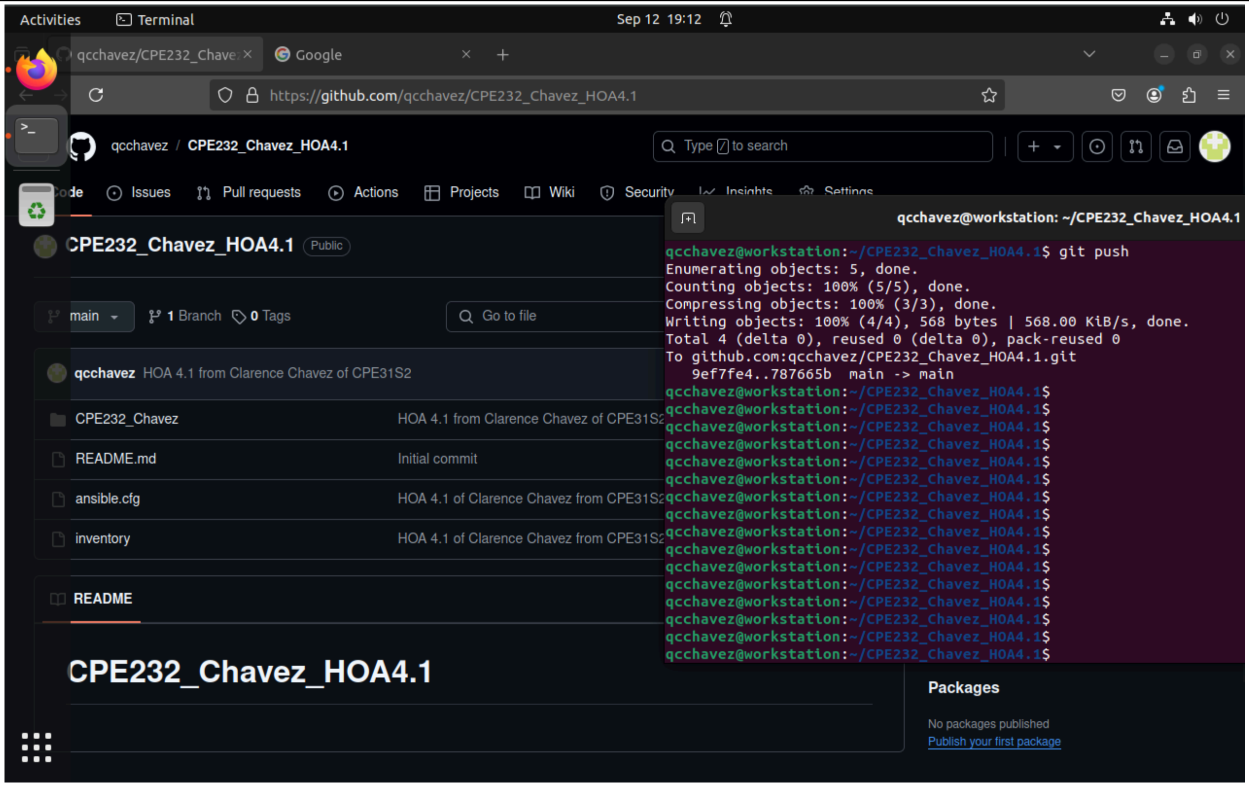
8. Run the playbook and describe the output. Did the new command change anything on the remote servers?



```
/home/qcchavez/CPE232_Chavez_HOA4.1/CPE232_Chavez/install_apache.yml as an inventory
source
[WARNING]: Unable to parse /home/qcchavez/CPE232_Chavez_HOA4.1/CPE232_Chavez as an
inventory source

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

TASK [Gathering Facts] *************************************************************
ok: [192.168.56.102]

TASK [update repository index] *****************************************************
changed: [192.168.56.102]

TASK [install apache2 package] *****************************************************
ok: [192.168.56.102]

TASK [add PHP support for apache] **************************************************
changed: [192.168.56.102]

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

qcchavez@workstation:~/CPE232_Chavez_HOA4.1/CPE232_Chavez$
```

- After making changes in the .yaml file and running the command, the remote server's update repository was changed, the installation of apache2 package was done, and the addition of PHP support for apache was changed.

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



qcchavez HOA 4.1 from Clarence Chavez of CPE31S2	
CPE232_Chavez	HOA 4.1 from Clarence Chavez of CPE31S2
README.md	Initial commit
ansible.cfg	HOA 4.1 of Clarence Chavez from CPE31S2
inventory	HOA 4.1 of Clarence Chavez from CPE31S2

CPE232_Chavez_HOA4.1

qcchavez@workstation: ~/CPE232_Chavez_HOA4.1

```
qcchavez@workstation:~/CPE232_Chavez_HOA4.1$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 568 bytes | 568.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:qcchavez/CPE232_Chavez_HOA4.1.git
   9ef7fe4..787665b  main -> main
qcchavez@workstation:~/CPE232_Chavez_HOA4.1$
qcchavez@workstation:~/CPE232_Chavez_HOA4.1$
qcchavez@workstation:~/CPE232_Chavez_HOA4.1$
qcchavez@workstation:~/CPE232_Chavez_HOA4.1$
qcchavez@workstation:~/CPE232_Chavez_HOA4.1$
qcchavez@workstation:~/CPE232_Chavez_HOA4.1$
qcchavez@workstation:~/CPE232_Chavez_HOA4.1$
qcchavez@workstation:~/CPE232_Chavez_HOA4.1$
qcchavez@workstation:~/CPE232_Chavez_HOA4.1$
qcchavez@workstation:~/CPE232_Chavez_HOA4.1$
qcchavez@workstation:~/CPE232_Chavez_HOA4.1$
qcchavez@workstation:~/CPE232_Chavez_HOA4.1$
qcchavez@workstation:~/CPE232_Chavez_HOA4.1$
qcchavez@workstation:~/CPE232_Chavez_HOA4.1$
qcchavez@workstation:~/CPE232_Chavez_HOA4.1$
qcchavez@workstation:~/CPE232_Chavez_HOA4.1$
```

Reflections:

Answer the following:

1. What is the importance of using a playbook?
   - The importance of using playbook is that it provides useful feature when managing systems especially remotely such as automation of codes, efficiency of remote usage, and provides seamless tasks.

2. Summarize what we have done on this activity.
   - In this activity, we have use several Ansible commands and managed to learn how to use playbooks. We used the workstation and the remote server. Mostly, the workstation was used and the remote server is only used for making changes using the workstation. We have configured several files such as inventory and ansible.cfg.