Name: Aducal, John Mark S.	**Date Performed: 09 / 13 / 2022**
Course/Section: CPE232 – CPE31S24	**Date Submitted: 09 / 14 / 2022**
Instructor: Engr. Jonathan V. Taylar	**Semester and SY: 1st Sem SY ’22-‘23**

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.

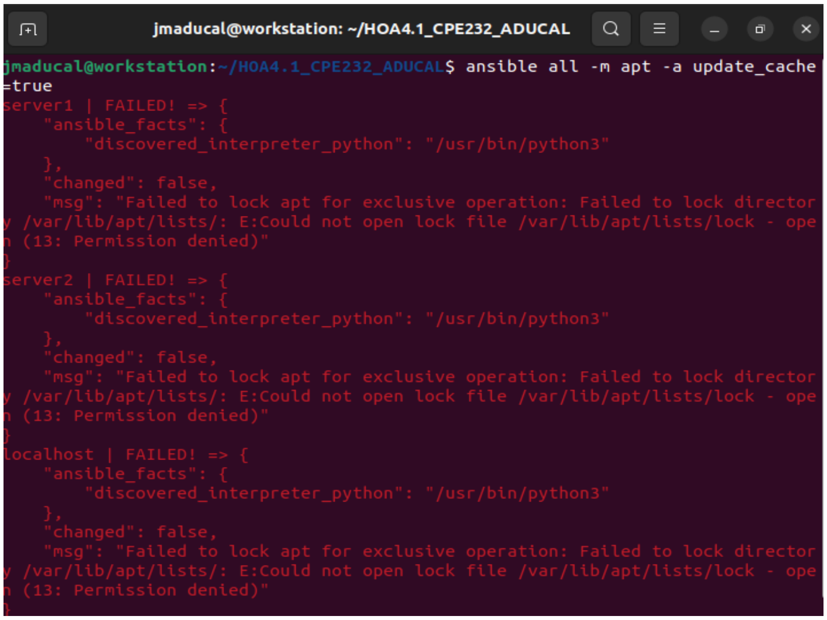
Playbooks record and execute **Ansible**’s configuration, deployment, and orchestration functions. They can describe a policy you want your remote systems to enforce, or a set of steps in a general IT process. If Ansible modules are the tools in your workshop, playbooks are your instruction manuals,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

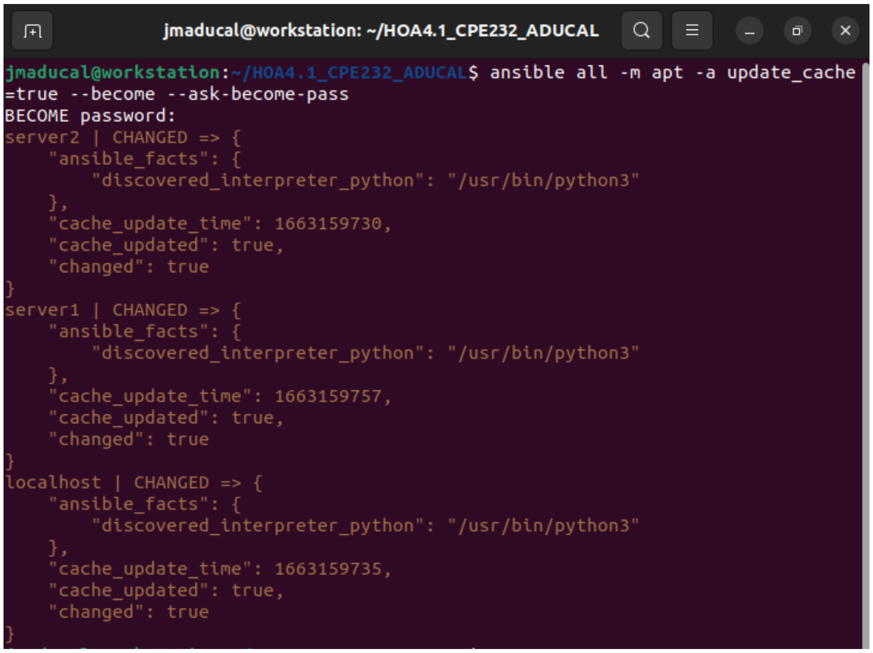
   ```
   ansible all -m apt -a update_cache=true
   ```

What is the result of the command? Is it successful? No, the result of the command displayed an error and permission denied.



```
jmaducal@workstation:~/HOA4.1_CPE232_ADUCAL$ ansible all -m apt -a update_cache
=true
server1 | 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)"
}
server2 | 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)"
}
localhost | 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)"
}
```

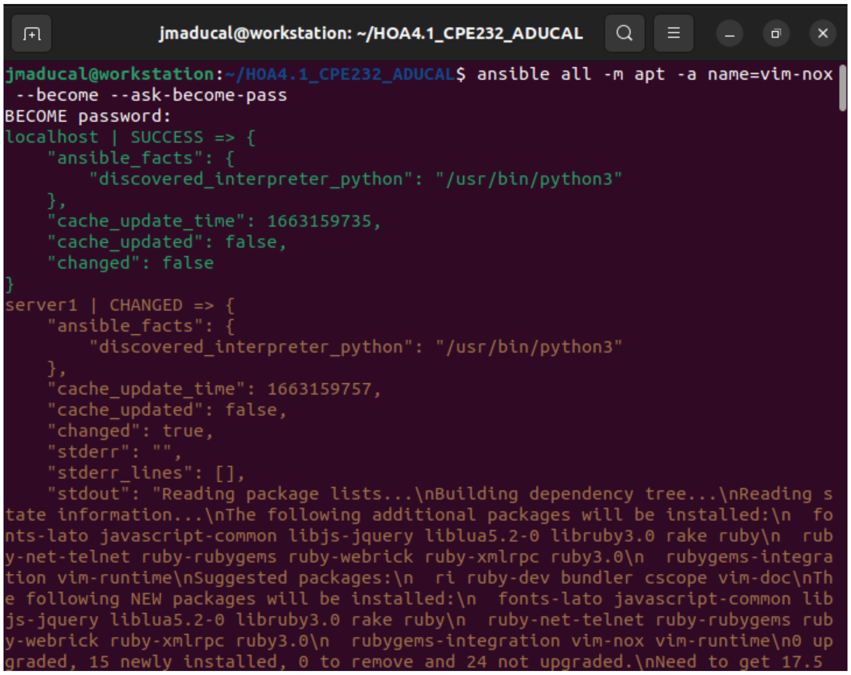
Try editing the command and add something that would elevate the privilege. Issue the command *ansible all -m apt -a update_cache=true --become --ask-become-pass.* Enter the sudo password when prompted. You will notice now that the output of this command is a success. The *update_cache=true* is the same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



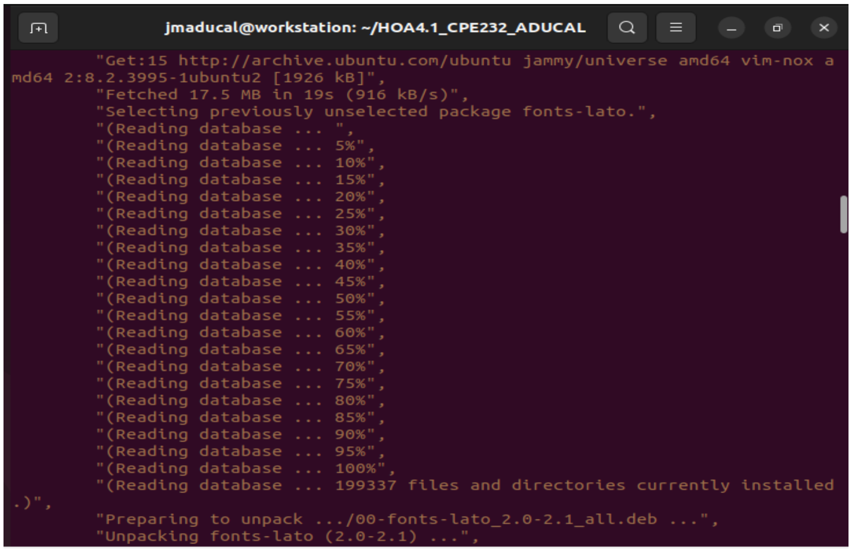
```
jmaducal@workstation:~/HOA4.1_CPE232_ADUCAL$ ansible all -m apt -a update_cache
=true --become --ask-become-pass
BECOME password:
server2 | CHANGED => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "cache_update_time": 1663159730,
    "cache_updated": true,
    "changed": true
}
server1 | CHANGED => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "cache_update_time": 1663159757,
    "cache_updated": true,
    "changed": true
}
localhost | CHANGED => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "cache_update_time": 1663159735,
    "cache_updated": true,
    "changed": true
}
```

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

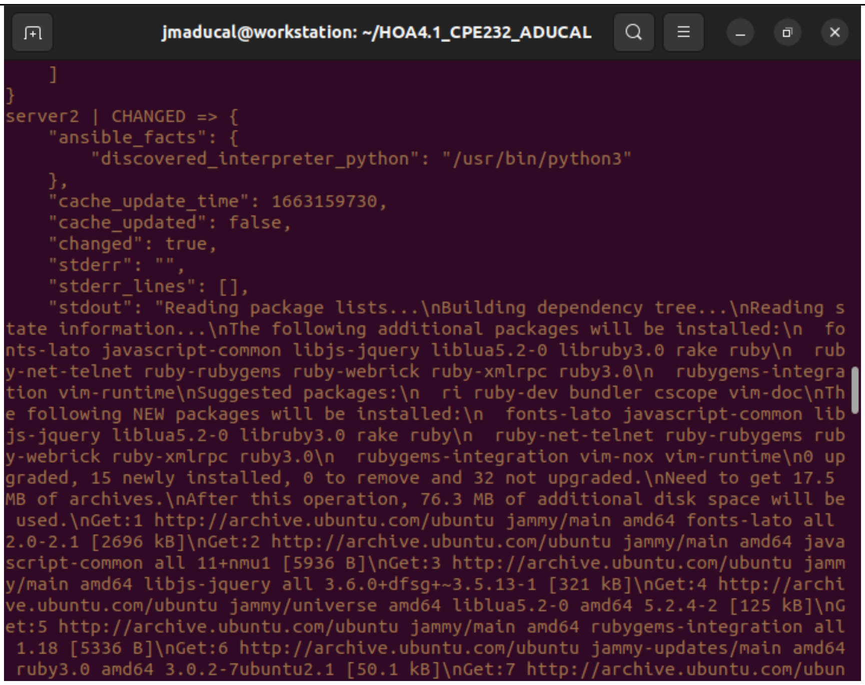
2. Let's try to install VIM, which is an almost compatible version of the UNIX editor Vi. To do this, we will just changed the module part in 1.1 instruction. Here is the command: *ansible all -m apt -a name=vim-nox --become --ask-become-pass.* The command would take some time after typing the password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



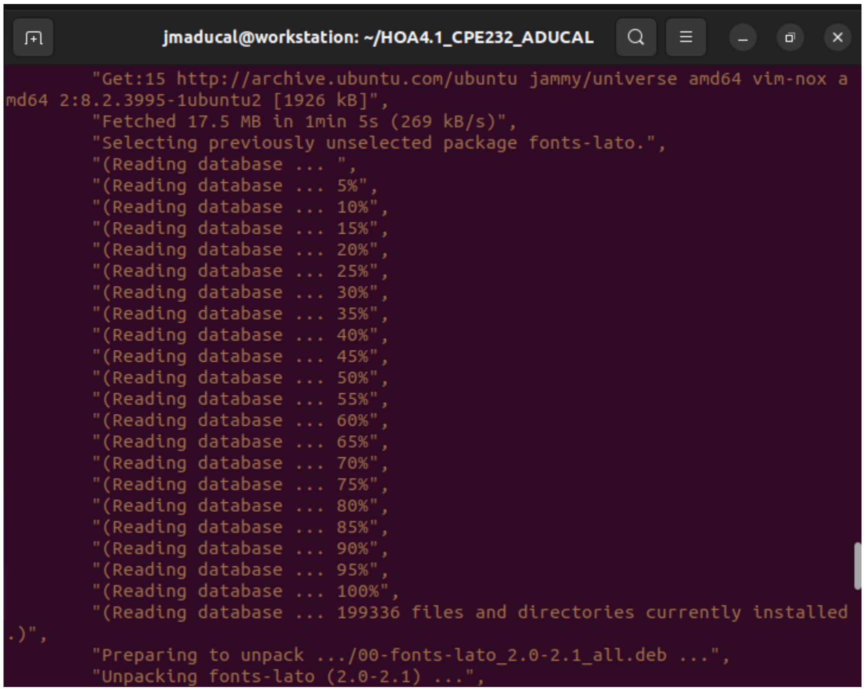
```
jmaducal@workstation:~/HOA4.1_CPE232_ADUCAL$ ansible all -m apt -a name=vim-nox
 --become --ask-become-pass
BECOME password:
localhost | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "cache_update_time": 1663159735,
    "cache_updated": false,
    "changed": false
}
server1 | CHANGED => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "cache_update_time": 1663159757,
    "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  ri ruby-dev bundler cscope vim-doc\nTh
e following NEW packages will be installed:\n  fonts-lato javascript-common lib
js-jquery liblua5.2-0 libruby3.0 rake ruby\n  ruby-net-telnet ruby-rubygems rub
y-webrick ruby-xmlrpc ruby3.0\n  rubygems-integration vim-nox vim-runtime\n0 up
graded, 15 newly installed, 0 to remove and 24 not upgraded.\nNeed to get 17.5
```



```
        "Get:15 http://archive.ubuntu.com/ubuntu jammy/universe amd64 vim-nox a
md64 2:8.2.3995-1ubuntu2 [1926 kB]",
        "Fetched 17.5 MB in 19s (916 kB/s)",
        "Selecting previously unselected package fonts-lato.",
        "(Reading database ... ",
        "(Reading database ... 5%",
        "(Reading database ... 10%",
        "(Reading database ... 15%",
        "(Reading database ... 20%",
        "(Reading database ... 25%",
        "(Reading database ... 30%",
        "(Reading database ... 35%",
        "(Reading database ... 40%",
        "(Reading database ... 45%",
        "(Reading database ... 50%",
        "(Reading database ... 55%",
        "(Reading database ... 60%",
        "(Reading database ... 65%",
        "(Reading database ... 70%",
        "(Reading database ... 75%",
        "(Reading database ... 80%",
        "(Reading database ... 85%",
        "(Reading database ... 90%",
        "(Reading database ... 95%",
        "(Reading database ... 100%",
        "(Reading database ... 199337 files and directories currently installed
.)",
        "Preparing to unpack .../00-fonts-lato_2.0-2.1_all.deb ...",
        "Unpacking fonts-lato (2.0-2.1) ...",
```

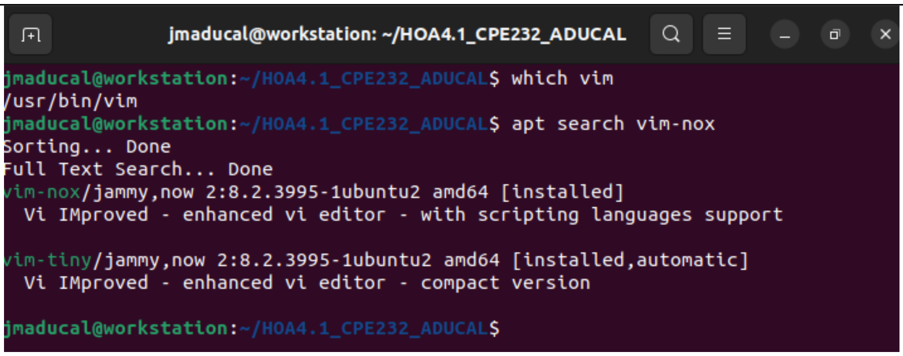


```
    ]
}
server2 | CHANGED => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "cache_update_time": 1663159730,
    "cache_updated": false,
    "changed": true,
    "stderr": "",
    "stderr_lines": [],
    "stdout": "Reading package lists...\nBuilding 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  ri ruby-dev bundler cscope vim-doc\nTh
e following NEW packages will be installed:\n  fonts-lato javascript-common lib
js-jquery liblua5.2-0 libruby3.0 rake ruby\n  ruby-net-telnet ruby-rubygems rub
y-webrick ruby-xmlrpc ruby3.0\n  rubygems-integration vim-nox vim-runtime\n0 up
graded, 15 newly installed, 0 to remove and 32 not upgraded.\nNeed to get 17.5
MB of archives.\nAfter this operation, 76.3 MB of additional disk space will be
 used.\nGet:1 http://archive.ubuntu.com/ubuntu jammy/main amd64 fonts-lato all
2.0-2.1 [2696 kB]\nGet:2 http://archive.ubuntu.com/ubuntu jammy/main amd64 java
script-common all 11+nmu1 [5936 B]\nGet:3 http://archive.ubuntu.com/ubuntu jamm
y/main amd64 libjs-jquery all 3.6.0+dfsg+~3.5.13-1 [321 kB]\nGet:4 http://archi
ve.ubuntu.com/ubuntu jammy/universe amd64 liblua5.2-0 amd64 5.2.4-2 [125 kB]\nG
et:5 http://archive.ubuntu.com/ubuntu jammy/main amd64 rubygems-integration all
 1.18 [5336 B]\nGet:6 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64
 ruby3.0 amd64 3.0.2-7ubuntu2.1 [50.1 kB]\nGet:7 http://archive.ubuntu.com/ubun
```



```
        "Get:15 http://archive.ubuntu.com/ubuntu jammy/universe amd64 vim-nox a
md64 2:8.2.3995-1ubuntu2 [1926 kB]",
        "Fetched 17.5 MB in 1min 5s (269 kB/s)",
        "Selecting previously unselected package fonts-lato.",
        "(Reading database ... ",
        "(Reading database ... 5%",
        "(Reading database ... 10%",
        "(Reading database ... 15%",
        "(Reading database ... 20%",
        "(Reading database ... 25%",
        "(Reading database ... 30%",
        "(Reading database ... 35%",
        "(Reading database ... 40%",
        "(Reading database ... 45%",
        "(Reading database ... 50%",
        "(Reading database ... 55%",
        "(Reading database ... 60%",
        "(Reading database ... 65%",
        "(Reading database ... 70%",
        "(Reading database ... 75%",
        "(Reading database ... 80%",
        "(Reading database ... 85%",
        "(Reading database ... 90%",
        "(Reading database ... 95%",
        "(Reading database ... 100%",
        "(Reading database ... 199336 files and directories currently installed
.)",
        "Preparing to unpack .../00-fonts-lato_2.0-2.1_all.deb ...",
        "Unpacking fonts-lato (2.0-2.1) ...",
```

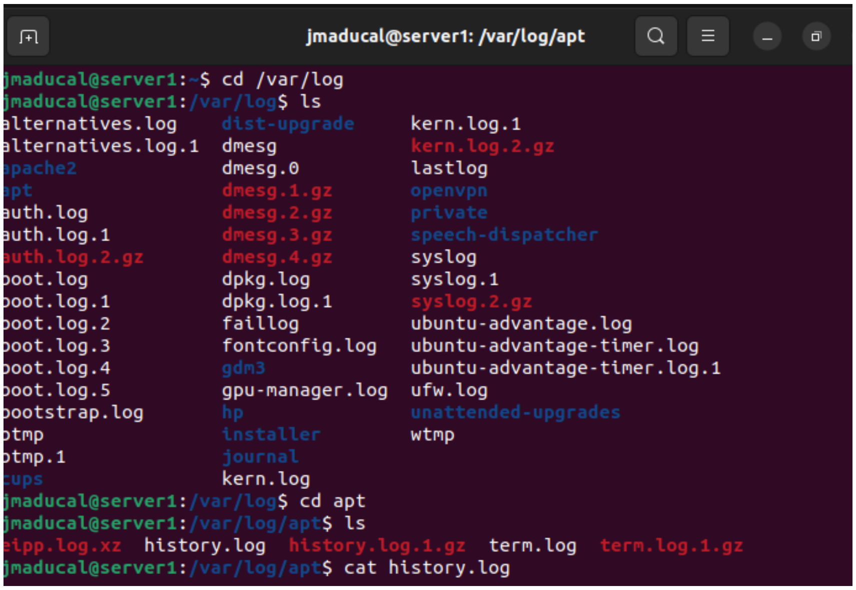
2.1 Verify that you have installed the package in the remote servers. Issue the command *which vim* and the command *apt search vim-nox* respectively. Was the command successful? YES



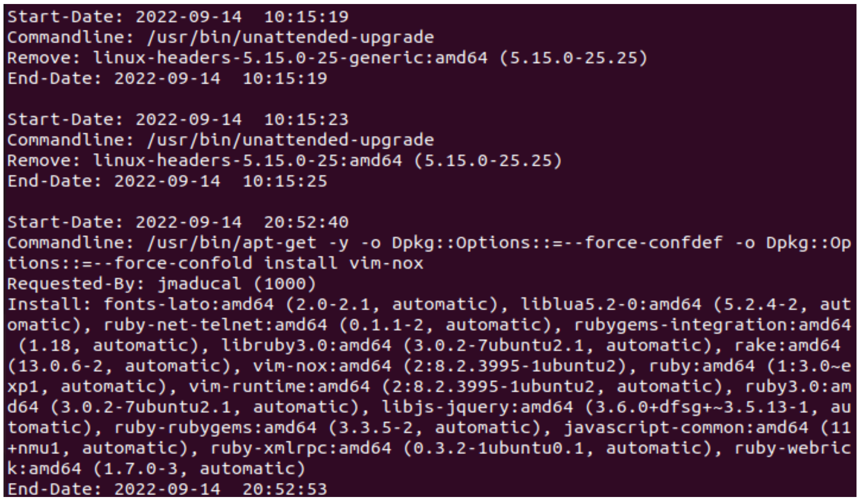
```
jmaducal@workstation:~/HOA4.1_CPE232_ADUCAL$ which vim
/usr/bin/vim
jmaducal@workstation:~/HOA4.1_CPE232_ADUCAL$ apt search vim-nox
Sorting... Done
Full Text Search... Done
vim-nox/jammy,now 2:8.2.3995-1ubuntu2 amd64 [installed]
  Vi IMproved - enhanced vi editor - with scripting languages support

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

jmaducal@workstation:~/HOA4.1_CPE232_ADUCAL$
```

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.

Server 1



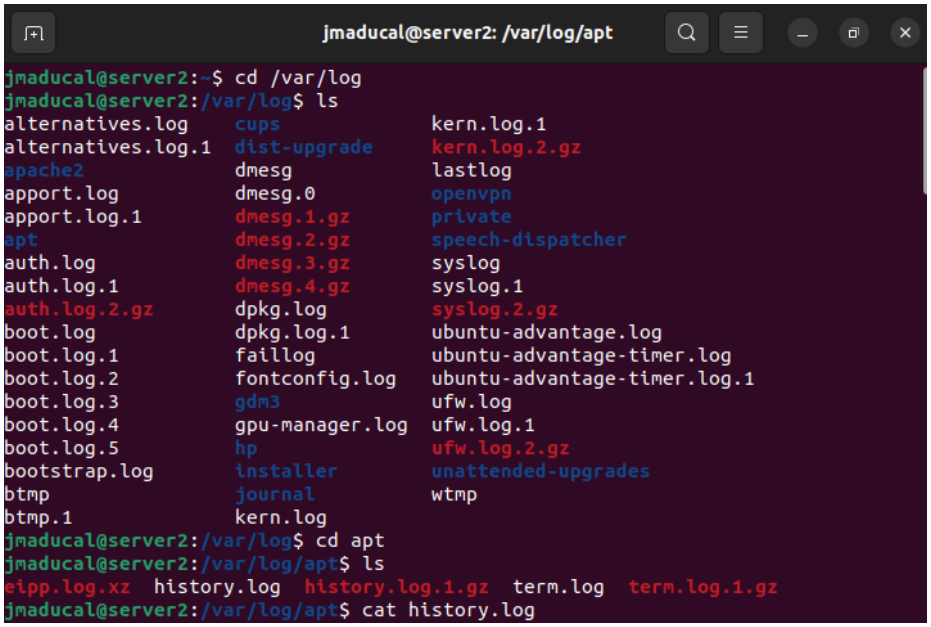
```
jmaducal@server1:~$ cd /var/log
jmaducal@server1:/var/log$ ls
alternatives.log    dist-upgrade     kern.log.1
alternatives.log.1  dmesg            kern.log.2.gz
apache2             dmesg.0          lastlog
apt                 dmesg.1.gz       openvpn
auth.log            dmesg.2.gz       private
auth.log.1          dmesg.3.gz       speech-dispatcher
auth.log.2.gz       dmesg.4.gz       syslog
boot.log            dpkg.log         syslog.1
boot.log.1          dpkg.log.1       syslog.2.gz
boot.log.2          faillog          ubuntu-advantage.log
boot.log.3          fontconfig.log   ubuntu-advantage-timer.log
boot.log.4          gdm3             ubuntu-advantage-timer.log.1
boot.log.5          gpu-manager.log  ufw.log
bootstrap.log       hp               unattended-upgrades
btmp                installer        wtmp
btmp.1              journal
cups                kern.log
jmaducal@server1:/var/log$ cd apt
jmaducal@server1:/var/log/apt$ ls
eipp.log.xz  history.log  history.log.1.gz  term.log  term.log.1.gz
jmaducal@server1:/var/log/apt$ cat history.log
```



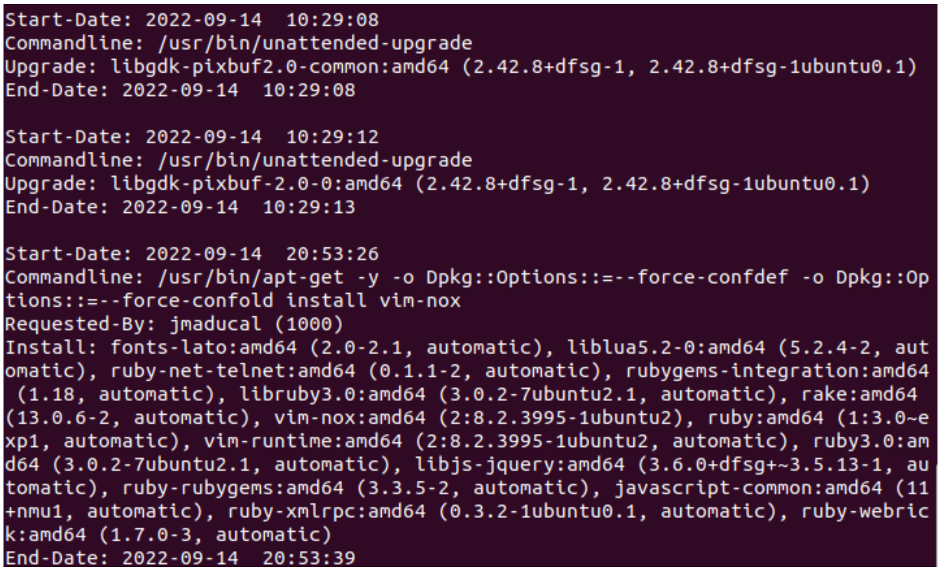
```
Start-Date: 2022-09-14  10:15:19
Commandline: /usr/bin/unattended-upgrade
Remove: linux-headers-5.15.0-25-generic:amd64 (5.15.0-25.25)
End-Date: 2022-09-14  10:15:19

Start-Date: 2022-09-14  10:15:23
Commandline: /usr/bin/unattended-upgrade
Remove: linux-headers-5.15.0-25:amd64 (5.15.0-25.25)
End-Date: 2022-09-14  10:15:25

Start-Date: 2022-09-14  20:52:40
Commandline: /usr/bin/apt-get -y -o Dpkg::Options::=--force-confdef -o Dpkg::Op
tions::=--force-confold install vim-nox
Requested-By: jmaducal (1000)
Install: fonts-lato:amd64 (2.0-2.1, automatic), liblua5.2-0:amd64 (5.2.4-2, aut
omatic), ruby-net-telnet:amd64 (0.1.1-2, automatic), rubygems-integration:amd64
 (1.18, automatic), libruby3.0:amd64 (3.0.2-7ubuntu2.1, automatic), rake:amd64
(13.0.6-2, automatic), vim-nox:amd64 (2:8.2.3995-1ubuntu2), ruby:amd64 (1:3.0~e
xp1, automatic), vim-runtime:amd64 (2:8.2.3995-1ubuntu2, automatic), ruby3.0:am
d64 (3.0.2-7ubuntu2.1, automatic), libjs-jquery:amd64 (3.6.0+dfsg+~3.5.13-1, au
tomatic), ruby-rubygems:amd64 (3.3.5-2, automatic), javascript-common:amd64 (11
+nmu1, automatic), ruby-xmlrpc:amd64 (0.3.2-1ubuntu0.1, automatic), ruby-webric
k:amd64 (1.7.0-3, automatic)
End-Date: 2022-09-14  20:52:53
```

In history.log, It shows the updates and upgrades of the packages to the new version of ubuntu system for Server 1.

Server 2



```
jmaducal@server2:~$ cd /var/log
jmaducal@server2:/var/log$ ls
alternatives.log    cups              kern.log.1
alternatives.log.1  dist-upgrade      kern.log.2.gz
apache2             dmesg             lastlog
apport.log          dmesg.0           openvpn
apport.log.1        dmesg.1.gz        private
apt                 dmesg.2.gz        speech-dispatcher
auth.log            dmesg.3.gz        syslog
auth.log.1          dmesg.4.gz        syslog.1
auth.log.2.gz       dpkg.log          syslog.2.gz
boot.log            dpkg.log.1        ubuntu-advantage.log
boot.log.1          faillog           ubuntu-advantage-timer.log
boot.log.2          fontconfig.log    ubuntu-advantage-timer.log.1
boot.log.3          gdm3              ufw.log
boot.log.4          gpu-manager.log   ufw.log.1
boot.log.5          hp                ufw.log.2.gz
bootstrap.log       installer         unattended-upgrades
btmp                journal           wtmp
btmp.1              kern.log
jmaducal@server2:/var/log$ cd apt
jmaducal@server2:/var/log/apt$ ls
eipp.log.xz  history.log  history.log.1.gz  term.log  term.log.1.gz
jmaducal@server2:/var/log/apt$ cat history.log
```



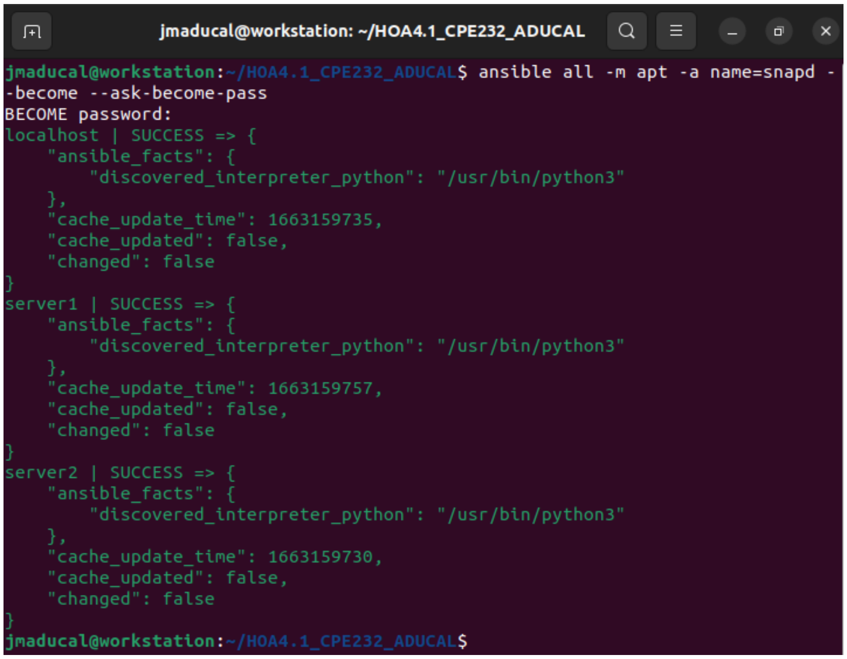
```
Start-Date: 2022-09-14  10:29:08
Commandline: /usr/bin/unattended-upgrade
Upgrade: libgdk-pixbuf2.0-common:amd64 (2.42.8+dfsg-1, 2.42.8+dfsg-1ubuntu0.1)
End-Date: 2022-09-14  10:29:08

Start-Date: 2022-09-14  10:29:12
Commandline: /usr/bin/unattended-upgrade
Upgrade: libgdk-pixbuf-2.0-0:amd64 (2.42.8+dfsg-1, 2.42.8+dfsg-1ubuntu0.1)
End-Date: 2022-09-14  10:29:13

Start-Date: 2022-09-14  20:53:26
Commandline: /usr/bin/apt-get -y -o Dpkg::Options::=--force-confdef -o Dpkg::Op
tions::=--force-confold install vim-nox
Requested-By: jmaducal (1000)
Install: fonts-lato:amd64 (2.0-2.1, automatic), liblua5.2-0:amd64 (5.2.4-2, aut
omatic), ruby-net-telnet:amd64 (0.1.1-2, automatic), rubygems-integration:amd64
 (1.18, automatic), libruby3.0:amd64 (3.0.2-7ubuntu2.1, automatic), rake:amd64
(13.0.6-2, automatic), vim-nox:amd64 (2:8.2.3995-1ubuntu2), ruby:amd64 (1:3.0~e
xp1, automatic), vim-runtime:amd64 (2:8.2.3995-1ubuntu2, automatic), ruby3.0:am
d64 (3.0.2-7ubuntu2.1, automatic), libjs-jquery:amd64 (3.6.0+dfsg+~3.5.13-1, au
tomatic), ruby-rubygems:amd64 (3.3.5-2, automatic), javascript-common:amd64 (11
+nmu1, automatic), ruby-xmlrpc:amd64 (0.3.2-1ubuntu0.1, automatic), ruby-webric
k:amd64 (1.7.0-3, automatic)
End-Date: 2022-09-14  20:53:39
```

In history.log, It shows the updates and upgrades of the packages to the new version of ubuntu system for Server 2.

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? It shows success, the command does not change anything in servers. I understand that the goal of the command is only to check for the installation package.

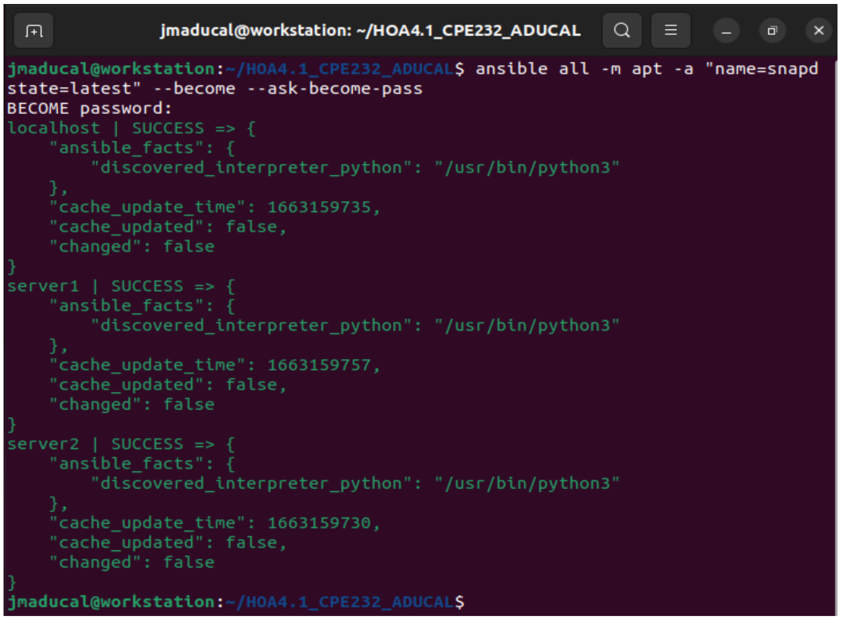


```
jmaducal@workstation:~/HOA4.1_CPE232_ADUCAL$ ansible all -m apt -a name=snapd -
-become --ask-become-pass
BECOME password:
localhost | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "cache_update_time": 1663159735,
    "cache_updated": false,
    "changed": false
}
server1 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "cache_update_time": 1663159757,
    "cache_updated": false,
    "changed": false
}
server2 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "cache_update_time": 1663159730,
    "cache_updated": false,
    "changed": false
}
jmaducal@workstation:~/HOA4.1_CPE232_ADUCAL$
```

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

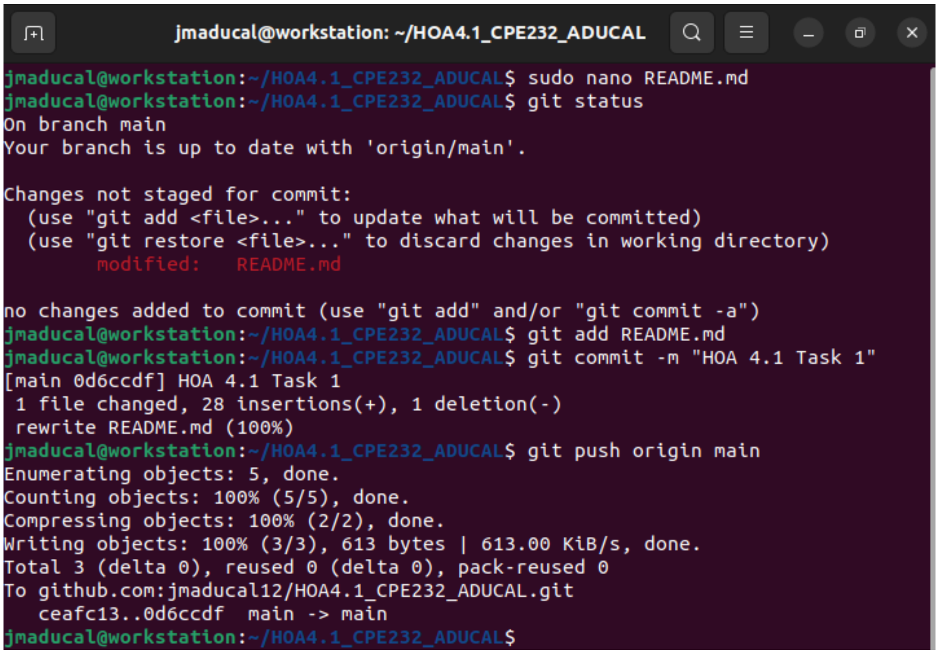
Describe the output of this command. Notice how we added the command *state=latest* and placed them in double quotations.

By adding the command state=latest it will check for the latest installation package.

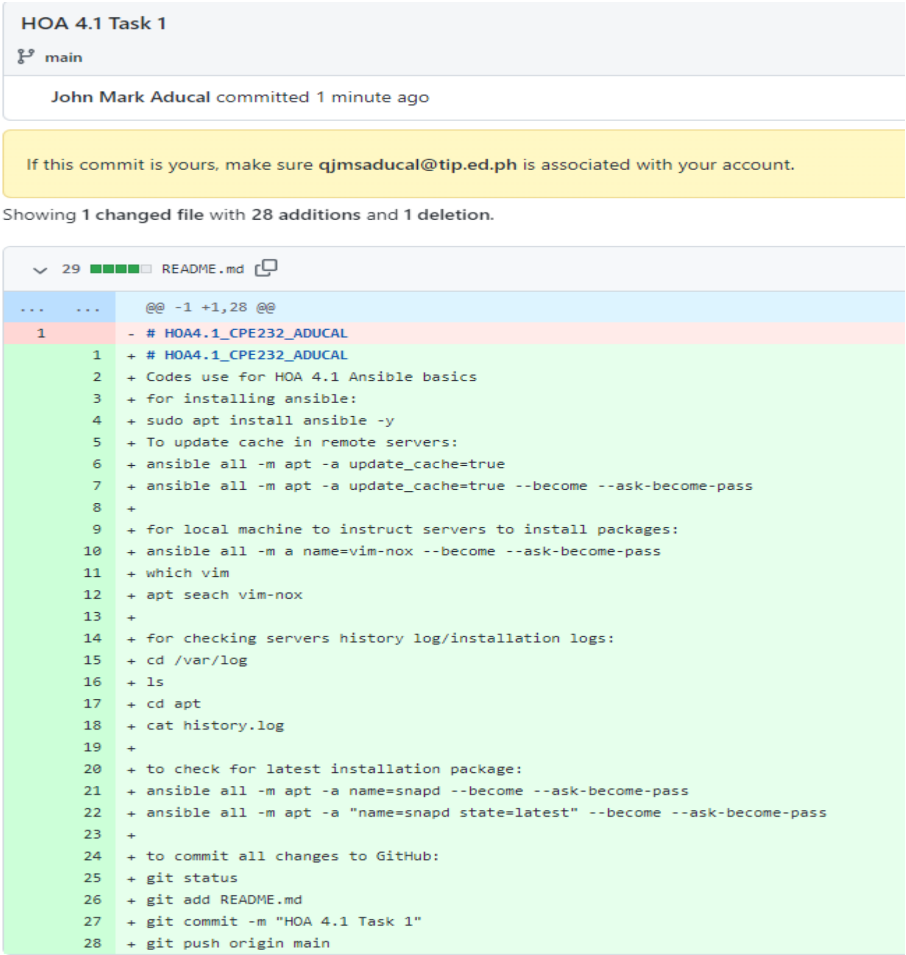


```
jmaducal@workstation:~/HOA4.1_CPE232_ADUCAL$ ansible all -m apt -a "name=snapd
state=latest" --become --ask-become-pass
BECOME password:
localhost | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "cache_update_time": 1663159735,
    "cache_updated": false,
    "changed": false
}
server1 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "cache_update_time": 1663159757,
    "cache_updated": false,
    "changed": false
}
server2 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "cache_update_time": 1663159730,
    "cache_updated": false,
    "changed": false
}
jmaducal@workstation:~/HOA4.1_CPE232_ADUCAL$
```

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



```
jmaducal@workstation:~/HOA4.1_CPE232_ADUCAL$ sudo nano README.md
jmaducal@workstation:~/HOA4.1_CPE232_ADUCAL$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")
jmaducal@workstation:~/HOA4.1_CPE232_ADUCAL$ git add README.md
jmaducal@workstation:~/HOA4.1_CPE232_ADUCAL$ git commit -m "HOA 4.1 Task 1"
[main 0d6ccdf] HOA 4.1 Task 1
 1 file changed, 28 insertions(+), 1 deletion(-)
 rewrite README.md (100%)
jmaducal@workstation:~/HOA4.1_CPE232_ADUCAL$ git push origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 613 bytes | 613.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:jmaducal12/HOA4.1_CPE232_ADUCAL.git
   ceafc13..0d6ccdf  main -> main
jmaducal@workstation:~/HOA4.1_CPE232_ADUCAL$
```



HOA 4.1 Task 1

main

John Mark Aducal committed 1 minute ago

If this commit is yours, make sure **qjmsaducal@tip.ed.ph** is associated with your account.

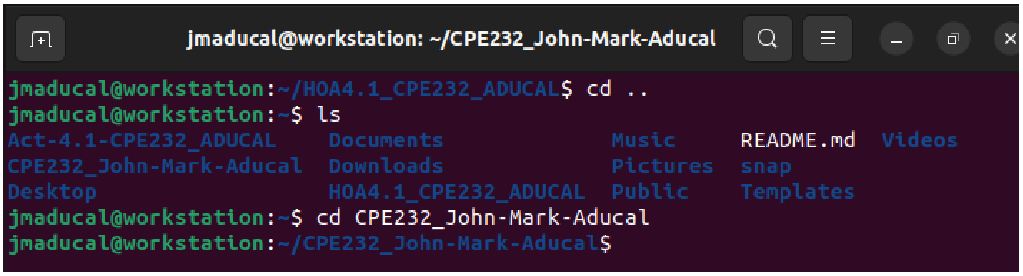
Showing **1 changed file** with **28 additions** and **1 deletion**.

29 README.md

```
@@ -1 +1,28 @@
- # HOA4.1_CPE232_ADUCAL
+ # HOA4.1_CPE232_ADUCAL
+ Codes use for HOA 4.1 Ansible basics
+ for installing ansible:
+ sudo apt install ansible -y
+ To update cache in remote servers:
+ ansible all -m apt -a update_cache=true
+ ansible all -m apt -a update_cache=true --become --ask-become-pass
+
+ for local machine to instruct servers to install packages:
+ ansible all -m a name=vim-nox --become --ask-become-pass
+ which vim
+ apt seach vim-nox
+
+ for checking servers history log/installation logs:
+ cd /var/log
+ ls
+ cd apt
+ cat history.log
+
+ to check for latest installation package:
+ ansible all -m apt -a name=snapd --become --ask-become-pass
+ ansible all -m apt -a "name=snapd state=latest" --become --ask-become-pass
+
+ to commit all changes to GitHub:
+ git status
+ git add README.md
+ git commit -m "HOA 4.1 Task 1"
+ git push origin main
```

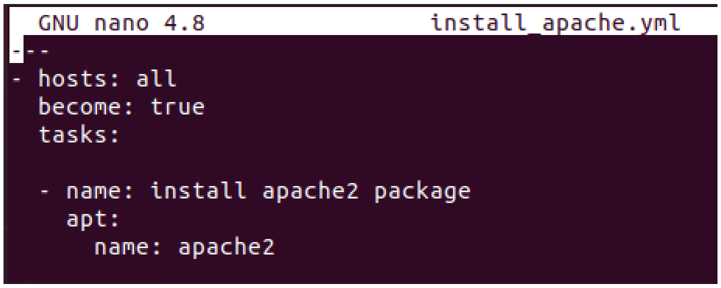
Task 2: Writing our First Playbook

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.

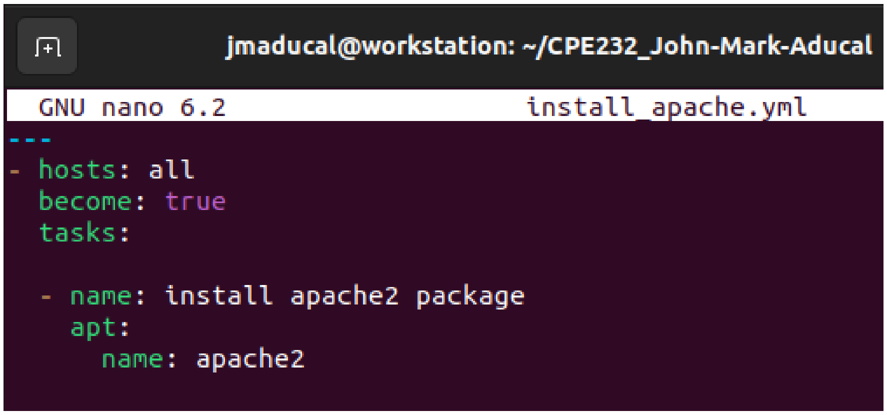


```
jmaducal@workstation:~/HOA4.1_CPE232_ADUCAL$ cd ..
jmaducal@workstation:~$ ls
Act-4.1-CPE232_ADUCAL    Documents              Music     README.md  Videos
CPE232_John-Mark-Aducal  Downloads              Pictures  snap
Desktop                  HOA4.1_CPE232_ADUCAL   Public    Templates
jmaducal@workstation:~$ cd CPE232_John-Mark-Aducal
jmaducal@workstation:~/CPE232_John-Mark-Aducal$
```

When the editor appears, type the following:



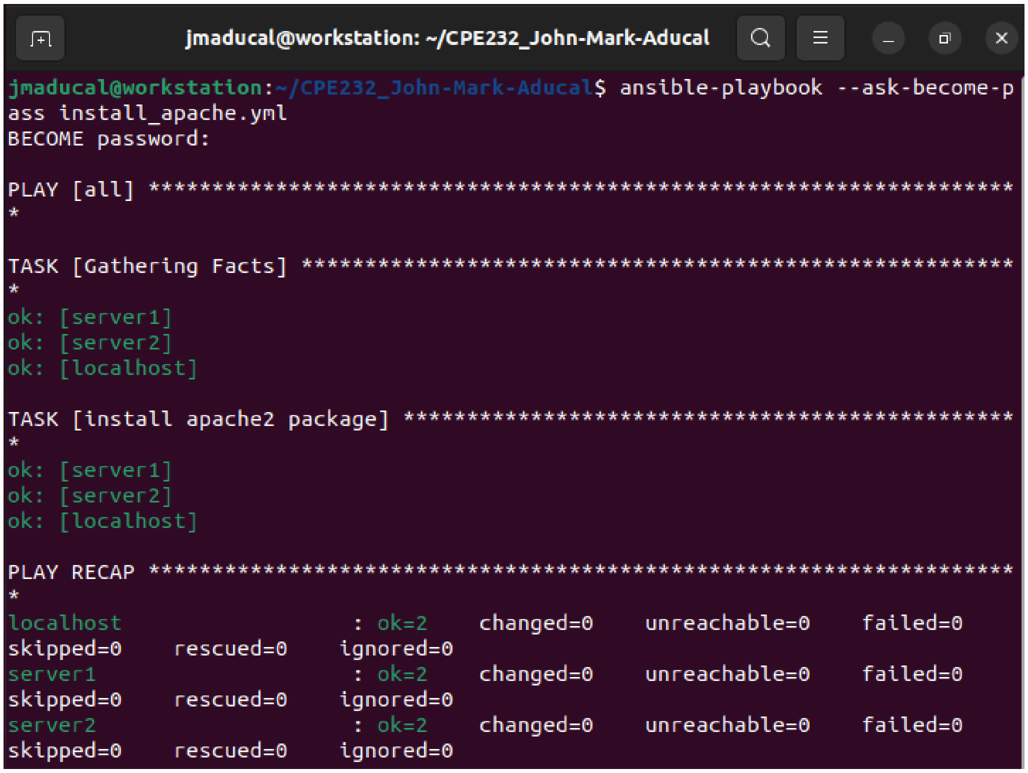
```
  GNU nano 4.8                      install_apache.yml
---
- hosts: all
  become: true
  tasks:

  - 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
```

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

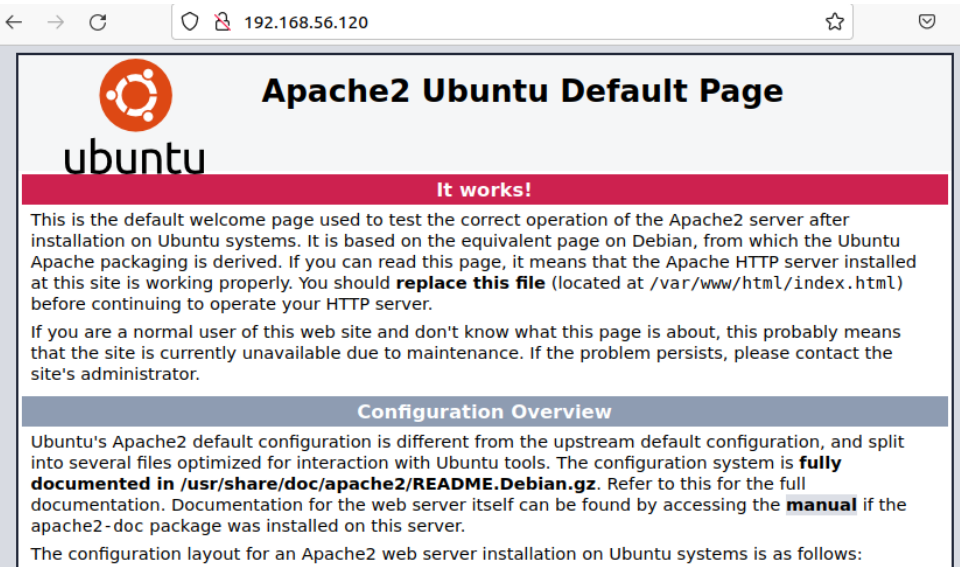
2. Run the yml file using the command: *ansible-playbook --ask-become-pass install_apache.yml.* Describe the result of this command.



```
jmaducal@workstation:~/CPE232_John-Mark-Aducal$ ansible-playbook --ask-become-p
ass install_apache.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************
*
ok: [server1]
ok: [server2]
ok: [localhost]

TASK [install apache2 package] *************************************************
*
ok: [server1]
ok: [server2]
ok: [localhost]

PLAY RECAP *********************************************************************
*
localhost                  : ok=2    changed=0    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
server1                    : ok=2    changed=0    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
server2                    : ok=2    changed=0    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
```

I think the command is used to install apache2 package to local host and remote servers using the ansible playbook.

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.

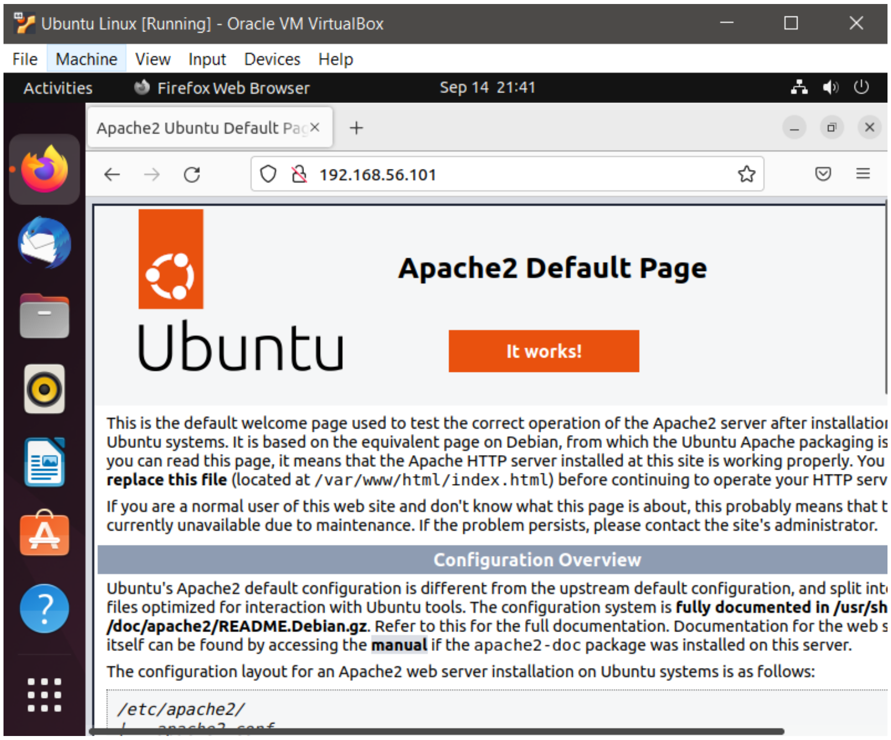
If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

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:

Localhost / Workstation


Ubuntu Linux [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Firefox Web Browser
Sep 14 21:41
Apache2 Ubuntu Default Pa
192.168.56.101
Ubuntu

Apache2 Default Page

It works!

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

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

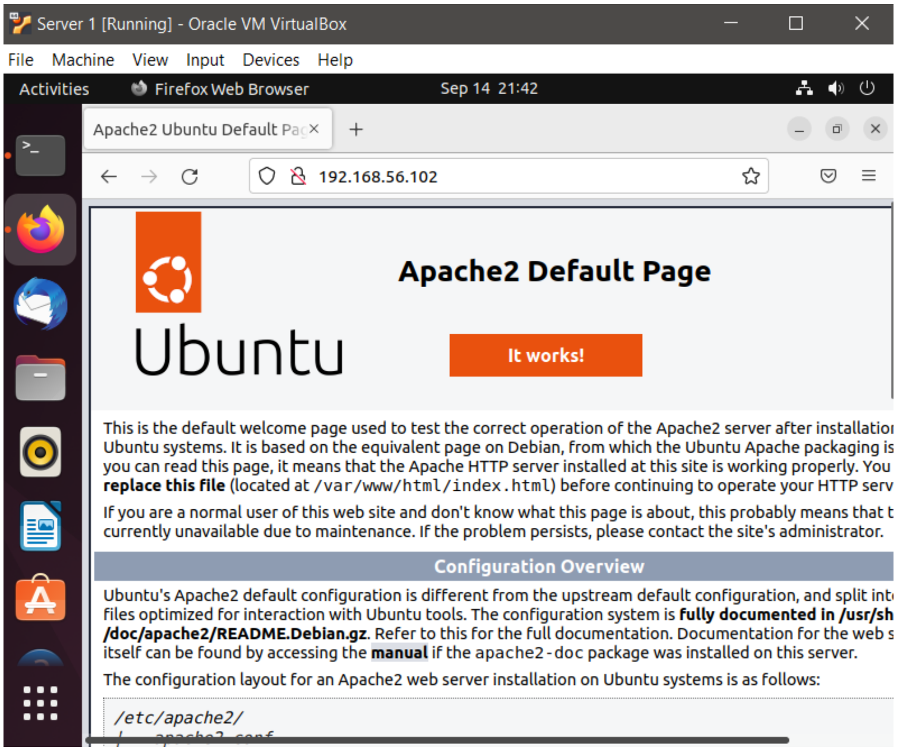
Configuration Overview

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

Server 1


Server 1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Firefox Web Browser
Sep 14 21:42
Apache2 Ubuntu Default Pa
192.168.56.102
Ubuntu

Apache2 Default Page

It works!

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

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

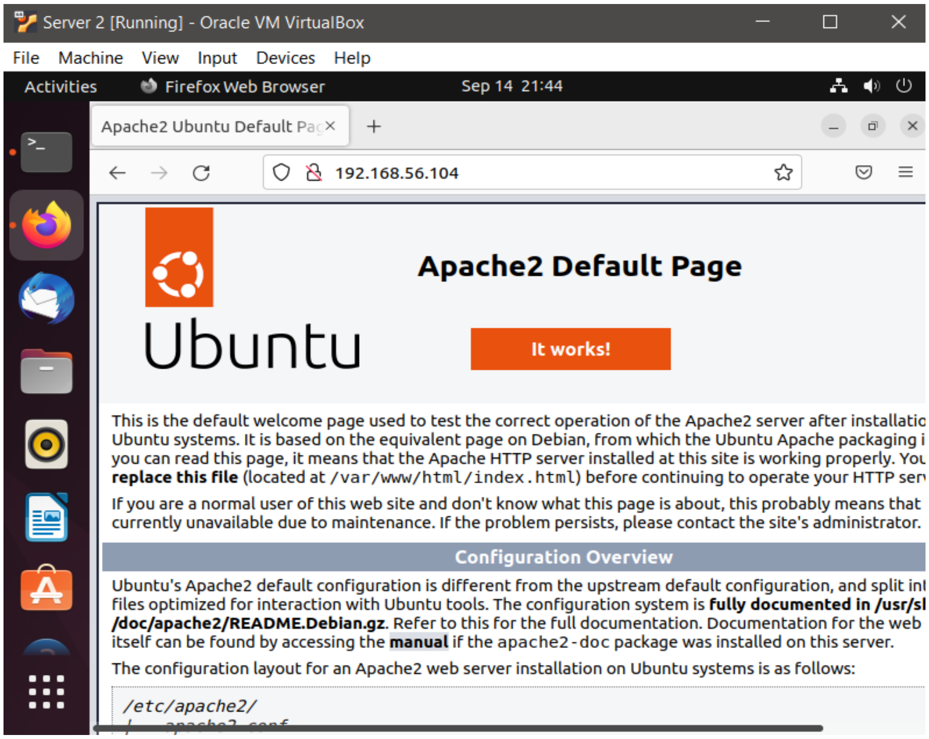
Configuration Overview

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

Server 2


Server 2 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Firefox Web Browser
Sep 14 21:44
Apache2 Ubuntu Default Pa
192.168.56.104

Apache2 Default Page

Ubuntu

It works!

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

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

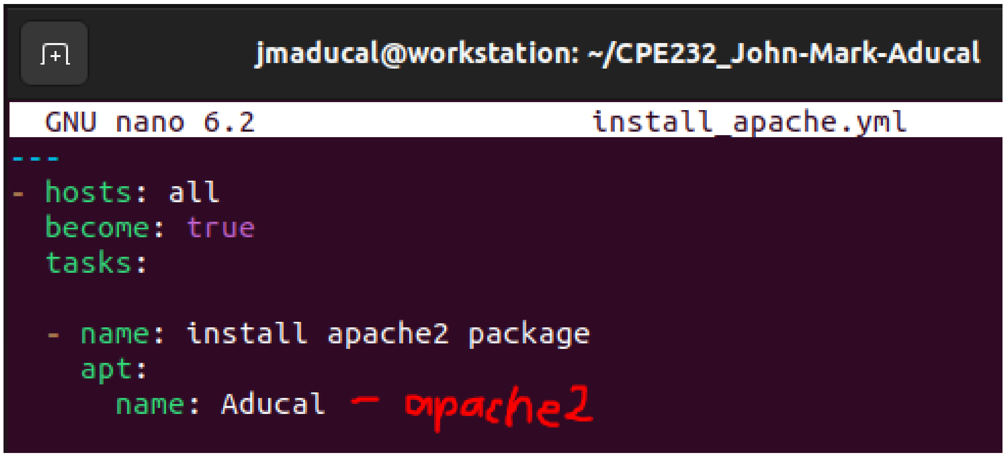
Configuration Overview

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

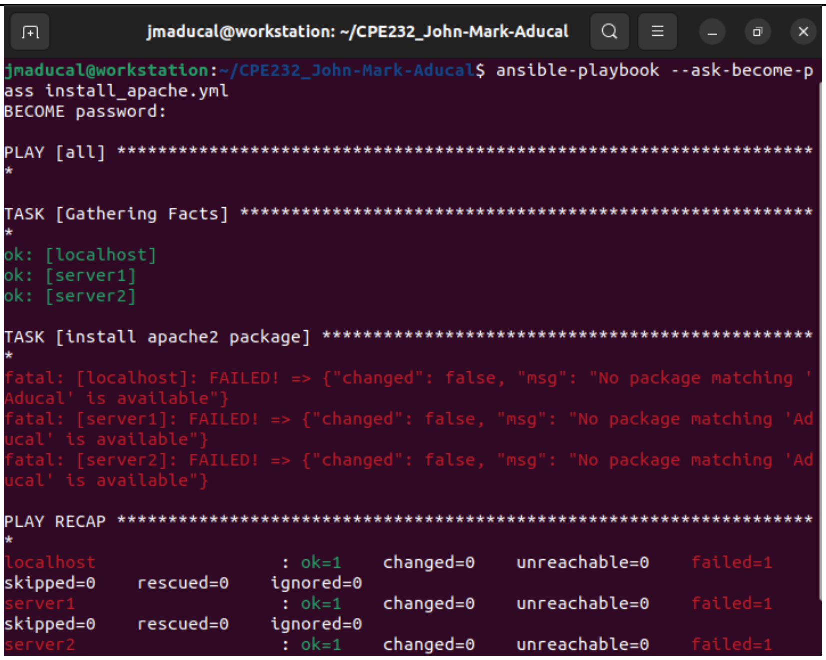
```
/etc/apache2/
```

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?
It shows an error no package matching Aducal is available. that's why it's FAILED.



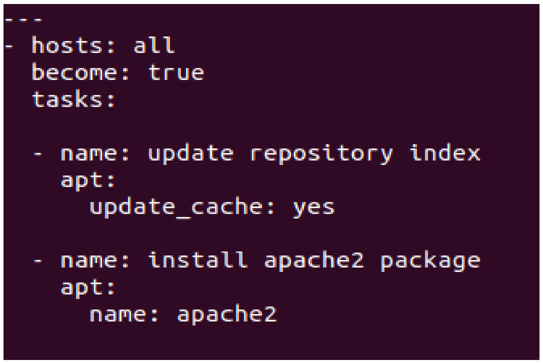
jmaducal@workstation: ~/CPE232_John-Mark-Aducal

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

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

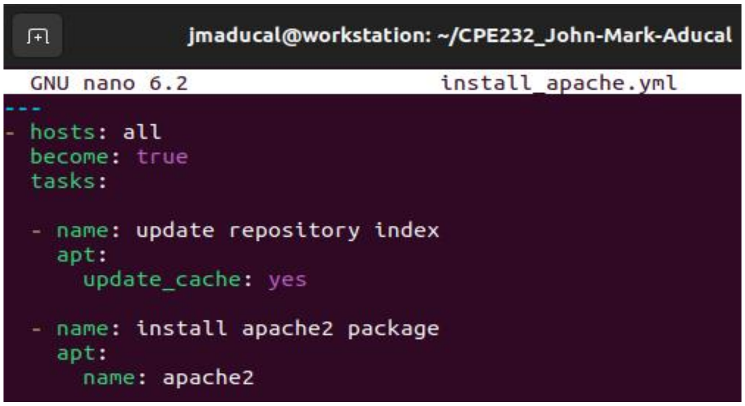


```
jmaducal@workstation:~/CPE232_John-Mark-Aducal$ ansible-playbook --ask-become-p
ass install_apache.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************
*
ok: [localhost]
ok: [server1]
ok: [server2]

TASK [install apache2 package] *************************************************
*
fatal: [localhost]: FAILED! => {"changed": false, "msg": "No package matching '
Aducal' is available"}
fatal: [server1]: FAILED! => {"changed": false, "msg": "No package matching 'Ad
ucal' is available"}
fatal: [server2]: FAILED! => {"changed": false, "msg": "No package matching 'Ad
ucal' is available"}

PLAY RECAP *********************************************************************
*
localhost                  : ok=1    changed=0    unreachable=0    failed=1
skipped=0    rescued=0    ignored=0
server1                    : ok=1    changed=0    unreachable=0    failed=1
skipped=0    rescued=0    ignored=0
server2                    : ok=1    changed=0    unreachable=0    failed=1
```

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.



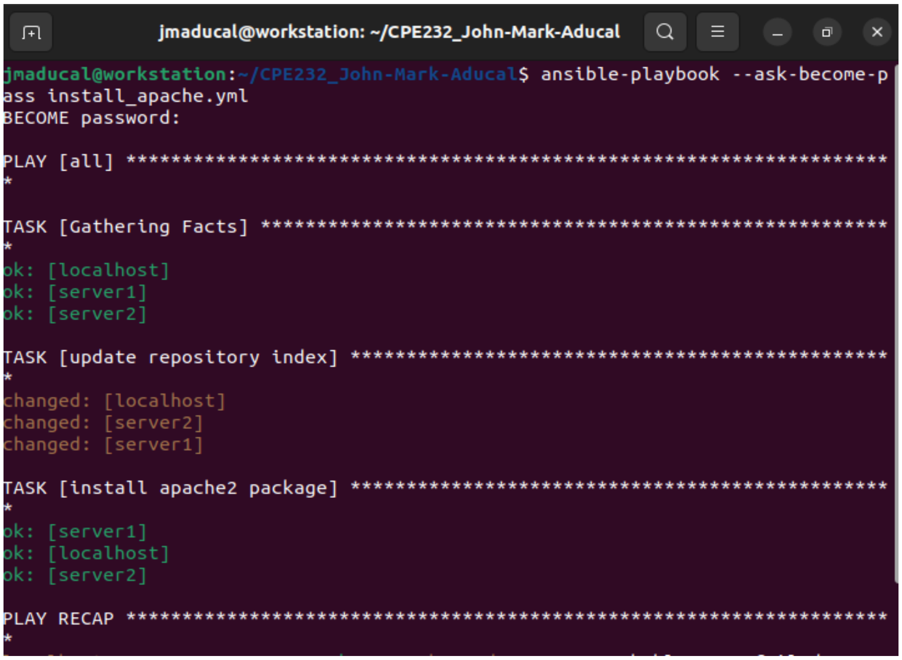
```
---
- hosts: all
  become: true
  tasks:

  - name: update repository index
    apt:
      update_cache: yes

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



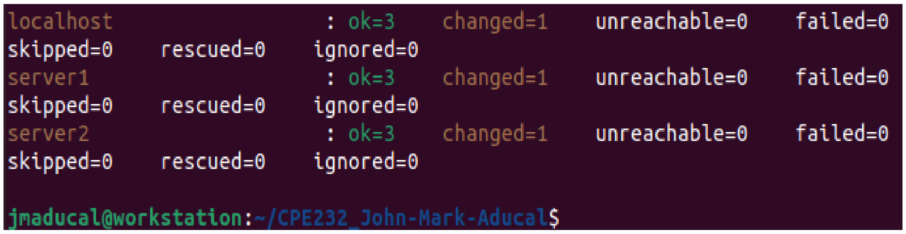
```
  GNU nano 6.2                        install_apache.yml
---
- hosts: all
  become: true
  tasks:

  - name: update repository index
    apt:
      update_cache: yes

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

Save the changes to this file and exit.

6. Run the playbook and describe the output. Did the new command change anything on the remote servers? Yes, the new command will update the repository index.

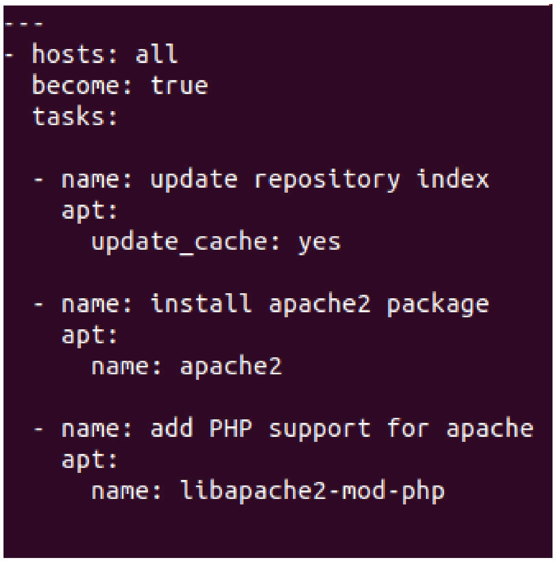


```
jmaducal@workstation:~/CPE232_John-Mark-Aducal$ ansible-playbook --ask-become-p
ass install_apache.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************
*
ok: [localhost]
ok: [server1]
ok: [server2]

TASK [update repository index] *************************************************
*
changed: [localhost]
changed: [server2]
changed: [server1]

TASK [install apache2 package] *************************************************
*
ok: [server1]
ok: [localhost]
ok: [server2]

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

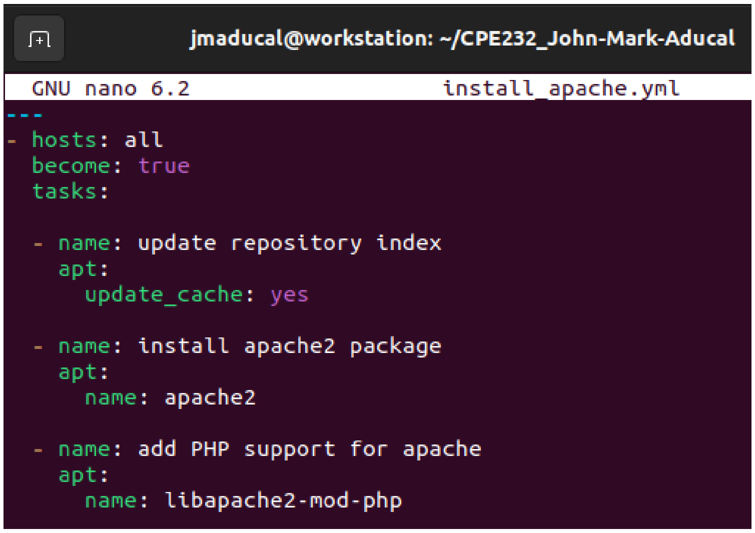


```
localhost                  : ok=3    changed=1    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
server1                    : ok=3    changed=1    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
server2                    : ok=3    changed=1    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0

jmaducal@workstation:~/CPE232_John-Mark-Aducal$
```

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



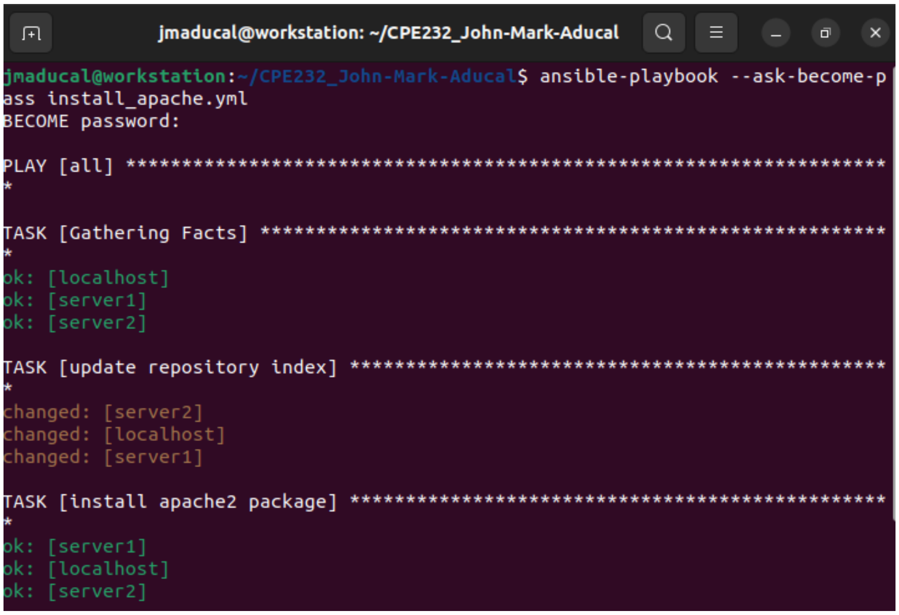
```
---
- hosts: all
  become: true
  tasks:

  - 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
```



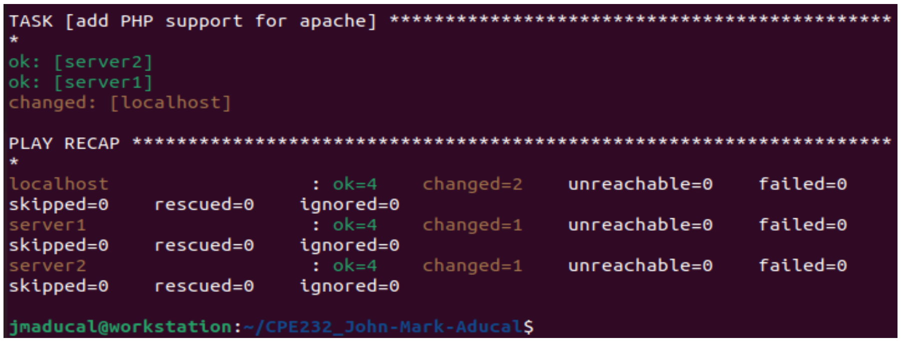
```
GNU nano 6.2                          install_apache.yml
---
- hosts: all
  become: true
  tasks:

  - 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
```

Save the changes to this file and exit.

8. Run the playbook and describe the output. Did the new command change anything on the remote servers? Yes, the new command will add PHP support for apache.

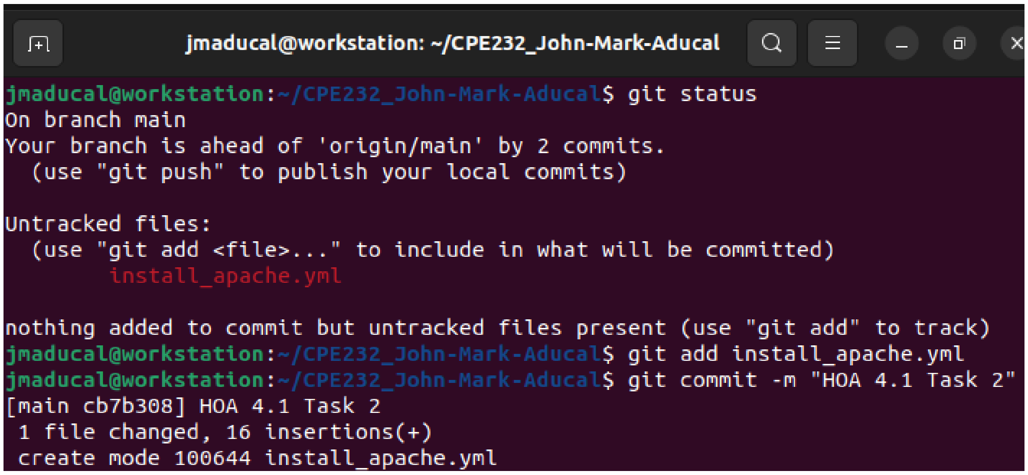


```
jmaducal@workstation:~/CPE232_John-Mark-Aducal$ ansible-playbook --ask-become-p
ass install_apache.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************
*
ok: [localhost]
ok: [server1]
ok: [server2]

TASK [update repository index] *************************************************
*
changed: [server2]
changed: [localhost]
changed: [server1]

TASK [install apache2 package] *************************************************
*
ok: [server1]
ok: [localhost]
ok: [server2]
```

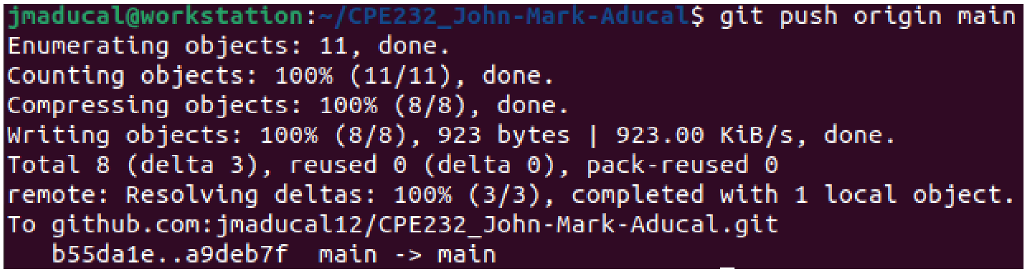


```
TASK [add PHP support for apache] **********************************************
*
ok: [server2]
ok: [server1]
changed: [localhost]

PLAY RECAP *********************************************************************
*
localhost                  : ok=4    changed=2    unreachable=0    failed=0    
skipped=0    rescued=0    ignored=0
server1                    : ok=4    changed=1    unreachable=0    failed=0    
skipped=0    rescued=0    ignored=0
server2                    : ok=4    changed=1    unreachable=0    failed=0    
skipped=0    rescued=0    ignored=0

jmaducal@workstation:~/CPE232_John-Mark-Aducal$
```

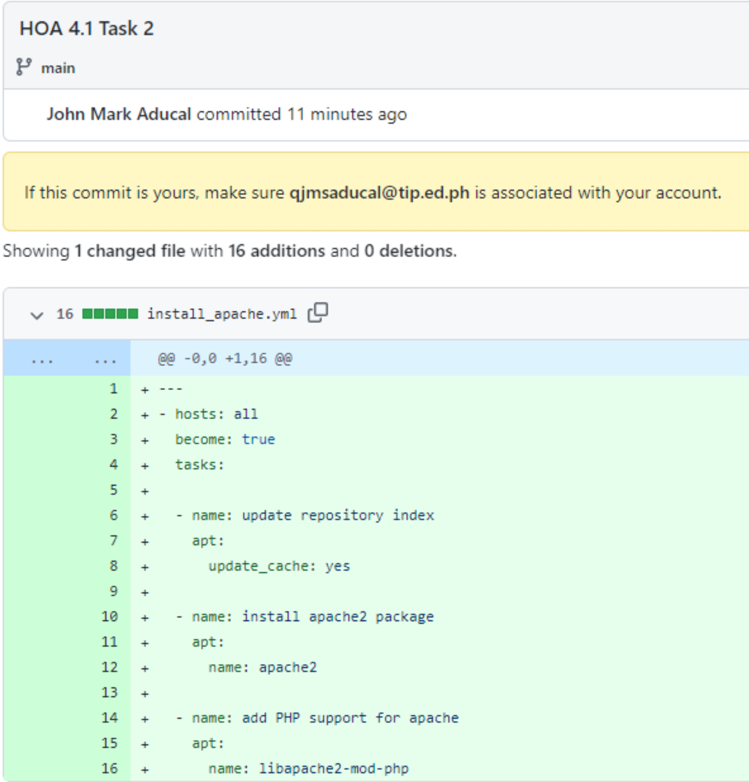
9. Finally, make sure that we are in sync with GitHub. Provide the link of your GitHub repository.



```
jmaducal@workstation:~/CPE232_John-Mark-Aducal$ git status
On branch main
Your branch is ahead of 'origin/main' by 2 commits.
  (use "git push" to publish your local commits)

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)
jmaducal@workstation:~/CPE232_John-Mark-Aducal$ git add install_apache.yml
jmaducal@workstation:~/CPE232_John-Mark-Aducal$ git commit -m "HOA 4.1 Task 2"
[main cb7b308] HOA 4.1 Task 2
 1 file changed, 16 insertions(+)
 create mode 100644 install_apache.yml
```



```
jmaducal@workstation:~/CPE232_John-Mark-Aducal$ git push origin main
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Compressing objects: 100% (8/8), done.
Writing objects: 100% (8/8), 923 bytes | 923.00 KiB/s, done.
Total 8 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), completed with 1 local object.
To github.com:jmaducal12/CPE232_John-Mark-Aducal.git
   b55da1e..a9deb7f  main -> main
```



HOA 4.1 Task 2

John Mark Aducal committed 11 minutes ago

If this commit is yours, make sure qjmsaducal@tip.ed.ph is associated with your account.

Showing 1 changed file with 16 additions and 0 deletions.

```
@@ -0,0 +1,16 @@
+ ---
+ - 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
```

GitHub repository Link:
https://github.com/jmaducal12/CPE232_John-Mark-Aducal.git

Reflections:

Answer the following:

1. **What is the importance of using a playbook?** Ansible Playbooks provides powerful automation and yet It have a simple configuration management for deployments, installations and updates for remote servers and workstations. It is useful for student system administrator just like me, IT Professionals and DevSecOps.

2. **Summarize what we have done on this activity.**
   In Task 1, I have learned how to execute ansible commands that would make update of cache in a remote servers and local host. To be able to execute the ansible command successfuly. It needs a privilege and –ask-become-pass ask for become password. Using the ansible command I can able to change something on the remote server and local host. Like for example installing packages and updating cache on remote servers using the local machine. Verifying the installed packages in remote servers and using the command to check for the latest installation packages. In Task 2, I have learned how to use ansible playbook first by creating a playbook (.yml) for playbook extension file. Writing some commands inside the playbook file for installing apache2 package in remote servers and localhost. And verifying if the apache2 package was successfully installed in remote servers by typing the IP Address of remote machines itself and It should display Apache2 Default page. To add another task in playbook by editing the playbook file. After executing the file with ansible command it will execute the new command and to update the remote servers. And lastly to commit all the changes and sync to GitHub.

Honor Pledge:

I affirm that I will not give or receive any unauthorized help on this activity and that all work will be my own.

John Mark S. Aducal