

Name: ROALLOS, Jean Gabriel Vincent G.	Date Performed: 08 / 26 / 2025
Course/Section: CPE212 - CPE31S2	Date Submitted: 09 / 05 / 2025
Instructor: Engr. Robin Valenzuela	Semester & SY: 1st Sem, 2025 - 2026
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: <i>Provide screenshots for each task.</i> 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	

```
ubuntu-gui@Workstation:~/AnsibleFiles$ cat inventory.yaml
[Server1_Brdg]
192.168.88.10

[Server1_NAT]
192.168.245.130

[Server2_Brdg]
192.168.88.16

[Server2_NAT]
192.168.245.131

[IP_Bridged:children]
Server1_Brdg
Server2_Brdg

[IP_NAT:children]
Server1_NAT
Server2_NAT
```

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

```
ubuntu-gui@Workstation:~/AnsibleFiles$ ansible IP_NAT -m apt -a update_cache=true
192.168.245.130 | 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)"
}
192.168.245.131 | 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)"
}
ubuntu-gui@Workstation:~/AnsibleFiles$
```

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

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.

```
-K, --ask-become-pass
ask for privilege escalation password
```

```
ubuntu-gui@Workstation:~/AnsibleFiles$ ansible IP_NAT -m apt -a update_cache=true --become -K
BECOME password:
192.168.245.130 | CHANGED => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "cache_update_time": 1756349525,
  "cache_updated": true,
  "changed": true
}
192.168.245.131 | CHANGED => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "cache_update_time": 1756349532,
  "cache_updated": true,
  "changed": true
}
ubuntu-gui@Workstation:~/AnsibleFiles$
```

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.

```

ubuntu-gui@Workstation:~/AnsibleFile$ ansible IP_NAT -m apt -a name=vim-nox --become -K
BECOME password:
192.168.245.130 | CHANGED => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "cache_update_time": 1756349525,
  "cache_updated": false,
  "changed": true,
  "stderr": "",
  "stderr_lines": [],
  "stdout": "Reading package lists...\nBuilding dependency tree...\nReading state information...\nThe following additional packages will be installed:\n fonts-lato javascript-common libjs-jquery liblua5.1-0 libruby libruby3.2 rake ruby ruby-net-telnet ruby-rubygems ruby-sdbm ruby-webrick ruby-xmlrpc\n ruby3.2 rubygems-integration vim-runtime\nSuggested packages:\n apache2 | lighttpd | httpd ri ruby-dev bundler cscope vim-doc\nThe following NEW packages will be installed:\n fonts-lato javascript-common libjs-jquery liblua5.1-0 libruby libruby3.2 rake ruby ruby-net-telnet ruby-rubygems ruby-sdbm ruby-webrick ruby-xmlrpc\n ruby3.2 rubygems-integration vim-nox vim-runtime\n0 upgraded, 17 newly installed, 0 to remove and 25 not upgraded.\nNeed to get 18.4 MB of archives.\nAfter this operation, 83.8 MB of additional disk space will be used.\nGet:1 http://ph.archive.ubuntu.com/ubuntu noble/main amd64 fonts-lato all 2.015-1 [2781 kB]\nGet:2 http://ph.archive.ubuntu.com/ubuntu noble/main amd64 javascript-common all 11+nmu1 [5936 B]\nGet:3 http://ph.archive.ubuntu.com/ubuntu noble/main amd64 libjs-jquery all 3.6.1+dfsg-3.5.14-1 [328 kB]\nGet:4 http://ph.archive.ubuntu.com/ubuntu noble/universe amd64 liblua5.1-0 amd64 5.1.5-9build2 [120 kB]\nGet:5 http://ph.archive.ubuntu.com/ubuntu noble/main amd64 rubygems-integration all 1.18 [5336 B]\nGet:6 http://ph.archive.ubuntu.com/ubuntu noble-updates/main amd64 ruby3.2 amd64 3.2.3-1ubuntu0.24.04.5 [50.7 kB]\nGet:7 http://ph.archive.ubuntu.com/ubuntu noble-updates/main amd64 ruby-rubygems all 3.4.20-1 [238 kB]\nGet:8 http://ph.archive.ubuntu.com/ubuntu noble/main amd64 rake all 13.0.6-3 [61.6 kB]\nGet:9 http://ph.archive.ubuntu.com/ubuntu noble/main amd64 rake all 13.0.6-3 [61.6 kB]\nGet:10 http://ph.archive.ubuntu.com/ubuntu noble/main amd64 ruby-net-telnet all 0.2.0-1 [13.3 kB]\nGet:11 http://ph.archive.ubuntu.com/ubuntu noble-updates/main amd64 ruby-webrick all 1.8.1-1ubuntu0.2 [53.0 kB]\nGet:12 http://ph.archive.ubuntu.com/ubuntu noble-updates/main amd64 ruby-xmlrpc all 0.3.2-2 [24.8 kB]\nGet:13 http://ph.archive.ubuntu.com/ubuntu noble/main amd64 ruby-sdbm amd64 1.0.0-5build4 [16.2 kB]\nGet:14 http://ph.archive.ubuntu.com/ubuntu noble-updates/main amd64 libruby3.2 amd64 3.2.3-1ubuntu0.24.04.5 [5341 kB]\nGet:15 http://ph.archive.ubuntu.com/ubuntu noble/main amd64 libruby amd64 1:3.2-ubuntu1 [4694 B]\nGet:16 http://ph.archive.ubuntu.com/ubuntu noble-updates/main amd64 vim-runtime all 2:9.1.0016-1ubuntu7.8 [7281 kB]\nGet:17 http://ph.archive.ubuntu.com/ubuntu noble-updates/universe amd64 vim-nox amd64 2:9.1.0016-1ubuntu7.8 [2098 kB]\nFetched 18.4 MB in 5s (3358 kB/s)\nSelecting previously unselected package fonts-lato.\nReading database ... \n(Reading database ... 5%\n(Reading database ... 10%\n(Reading database ... 15%\n(Reading database ... 20%\n(Reading database ... 25%\n(Reading database ... 30%\n(Reading database ... 35%\n(Reading database ... 40%\n(Reading database ... 45%\n(Reading database ... 50%\n(Reading database ... 55%\n(Reading database ... 60%\n(Reading database ... 65%\n(Reading database ... 70%\n(Reading database ... 75%\n(Reading database ... 80%\n(Reading database ... 85%\n(Reading database ... 90%\n(Reading database ... 95%\n(Reading database ... 100%\n)

```

```

192.168.245.131 | CHANGED => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "cache_update_time": 1756349532,
  "cache_updated": false,
  "changed": true,
  "stderr": "",
  "stderr_lines": [],
  "stdout": "Reading package lists...\nBuilding dependency tree...\nReading state information...\nThe following additional packages will be installed:\n fonts-lato javascript-common libjs-jquery liblua5.1-0 libruby libruby3.2 rake ruby ruby-net-telnet ruby-rubygems ruby-sdbm ruby-webrick ruby-xmlrpc\n ruby3.2 rubygems-integration vim-runtime\nSuggested packages:\n apache2 | lighttpd | httpd ri ruby-dev bundler cscope vim-doc\nThe following NEW packages will be installed:\n fonts-lato javascript-common libjs-jquery liblua5.1-0 libruby libruby3.2 rake ruby ruby-net-telnet ruby-rubygems ruby-sdbm ruby-webrick ruby-xmlrpc\n ruby3.2 rubygems-integration vim-nox vim-runtime\n0 upgraded, 17 newly installed, 0 to remove and 25 not upgraded.\nNeed to get 18.4 MB of archives.\nAfter this operation, 83.8 MB of additional disk space will be used.\nGet:1 http://ph.archive.ubuntu.com/ubuntu noble/main amd64 fonts-lato all 2.015-1 [2781 kB]\nGet:2 http://ph.archive.ubuntu.com/ubuntu noble/main amd64 javascript-common all 11+nmu1 [5936 B]\nGet:3 http://ph.archive.ubuntu.com/ubuntu noble/main amd64 libjs-jquery all 3.6.1+dfsg-3.5.14-1 [328 kB]\nGet:4 http://ph.archive.ubuntu.com/ubuntu noble/universe amd64 liblua5.1-0 amd64 5.1.5-9build2 [120 kB]\nGet:5 http://ph.archive.ubuntu.com/ubuntu noble/main amd64 rubygems-integration all 1.18 [5336 B]\nGet:6 http://ph.archive.ubuntu.com/ubuntu noble-updates/main amd64 ruby3.2 amd64 3.2.3-1ubuntu0.24.04.5 [50.7 kB]\nGet:7 http://ph.archive.ubuntu.com/ubuntu noble-updates/main amd64 ruby-rubygems all 3.4.20-1 [238 kB]\nGet:8 http://ph.archive.ubuntu.com/ubuntu noble/main amd64 rake all 13.0.6-3 [61.6 kB]\nGet:9 http://ph.archive.ubuntu.com/ubuntu noble/main amd64 rake all 13.0.6-3 [61.6 kB]\nGet:10 http://ph.archive.ubuntu.com/ubuntu noble/main amd64 ruby-net-telnet all 0.2.0-1 [13.3 kB]\nGet:11 http://ph.archive.ubuntu.com/ubuntu noble-updates/main amd64 ruby-webrick all 1.8.1-1ubuntu0.2 [53.0 kB]\nGet:12 http://ph.archive.ubuntu.com/ubuntu noble-updates/main amd64 ruby-xmlrpc all 0.3.2-2 [24.8 kB]\nGet:13 http://ph.archive.ubuntu.com/ubuntu noble/main amd64 ruby-sdbm amd64 1.0.0-5build4 [16.2 kB]\nGet:14 http://ph.archive.ubuntu.com/ubuntu noble-updates/main amd64 libruby3.2 amd64 3.2.3-1ubuntu0.24.04.5 [5341 kB]\nGet:15 http://ph.archive.ubuntu.com/ubuntu noble/main amd64 libruby amd64 1:3.2-ubuntu1 [4694 B]\nGet:16 http://ph.archive.ubuntu.com/ubuntu noble-updates/main amd64 vim-runtime all 2:9.1.0016-1ubuntu7.8 [7281 kB]\nGet:17 http://ph.archive.ubuntu.com/ubuntu noble-updates/universe amd64 vim-nox amd64 2:9.1.0016-1ubuntu7.8 [2098 kB]\nFetched 18.4 MB in 13s (1396 kB/s)\nSelecting previously unselected package fonts-lato.\nReading database ... \n(Reading database ... 5%\n(Reading database ... 10%\n(Reading database ... 15%\n(Reading database ... 20%\n(Reading database ... 25%\n(Reading database ... 30%\n(Reading database ... 35%\n(Reading database ... 40%\n(Reading database ... 45%\n(Reading database ... 50%\n(Reading database ... 55%\n(Reading database ... 60%\n(Reading database ... 65%\n(Reading database ... 70%\n(Reading database ... 75%\n(Reading database ... 80%\n(Reading database ... 85%\n(Reading database ... 90%\n(Reading database ... 95%\n(Reading database ... 100%\n)

```

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?

```
ubuntu-gui@Server1:~$ which vim
/usr/bin/vim
ubuntu-gui@Server1:~$ sudo apt search vim-nox
[sudo] password for ubuntu-gui:
Sorting... Done
Full Text Search... Done
vim-nox/noble-updates,noble-security,now 2:9.1.0016-1ubuntu7.8 amd64 [installed]
  Vi IMproved - enhanced vi editor - with scripting languages support

vim-tiny/noble-updates,noble-security,now 2:9.1.0016-1ubuntu7.8 amd64 [installed,automatic]
  Vi IMproved - enhanced vi editor - compact version

ubuntu-gui@Server1:~$
```

```
ubuntu-gui@Server2:~$ which vim
/usr/bin/vim
ubuntu-gui@Server2:~$ sudo apt search vim-nox
[sudo] password for ubuntu-gui:
Sorting... Done
Full Text Search... Done
vim-nox/noble-updates,noble-security,now 2:9.1.0016-1ubuntu7.8 amd64 [installed]
  Vi IMproved - enhanced vi editor - with scripting languages support

vim-tiny/noble-updates,noble-security,now 2:9.1.0016-1ubuntu7.8 amd64 [installed,automatic]
  Vi IMproved - enhanced vi editor - compact version

ubuntu-gui@Server2:~$
```

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*.

```
Start-Date: 2025-08-28 10:53:55
Commandline: /usr/bin/apt-get -y -o Dpkg::Options::=--force-confdef -o Dpkg::Options::=--force-confold install vim-nox=2:9.1.0016-1ubuntu7.8
Requested-By: ubuntu-gui (1000)
Install: ruby-sdbm:amd64 (1.0.0-5build4, automatic), liblua5.1-0:amd64 (5.1.5-9build2, automatic), fonts-lato:amd64 (2.015-1, automatic)
End-Date: 2025-08-28 10:54:07
```

```
Start-Date: 2025-08-28 10:54:13
Commandline: /usr/bin/apt-get -y -o Dpkg::Options::=--force-confdef -o Dpkg::Options::=--force-confold install vim-nox=2:9.1.0016-1ubuntu7.8
Requested-By: ubuntu-gui (1000)
Install: ruby-sdbm:amd64 (1.0.0-5build4, automatic), liblua5.1-0:amd64 (5.1.5-9build2, automatic), fonts-lato:amd64 (2.015-1, automatic)
End-Date: 2025-08-28 10:54:25
```

3. This time, we will install a package called *snapt*. *Snapt* 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=snapt --become --ask-become-pass*

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

```
ubuntu-gui@Workstation:~/AnsibleFiles$ ansible IP_NAT -m apt -a name=snapd --become -K
BECOME password:
192.168.245.130 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "cache_update_time": 1756349525,
  "cache_updated": false,
  "changed": false
}
192.168.245.131 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "cache_update_time": 1756349532,
  "cache_updated": false,
  "changed": false
}
ubuntu-gui@Workstation:~/AnsibleFiles$
```

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.

```
ubuntu-gui@Workstation:~/AnsibleFiles$ ansible IP_NAT -m apt -a "name=snapd state=latest" --become -K
BECOME password:
192.168.245.130 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "cache_update_time": 1756349525,
  "cache_updated": false,
  "changed": false
}
192.168.245.131 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "cache_update_time": 1756349532,
  "cache_updated": false,
  "changed": false
}
ubuntu-gui@Workstation:~/AnsibleFiles$
```

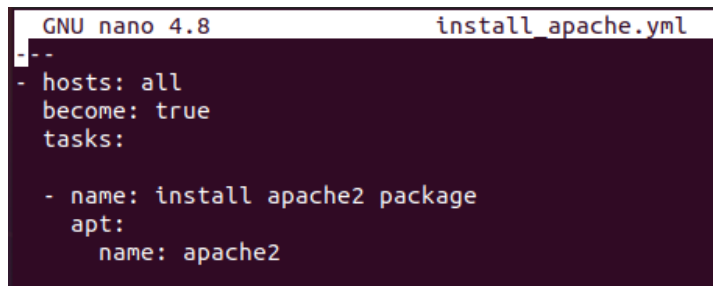
4. At this point, make sure to commit all changes to GitHub.

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.

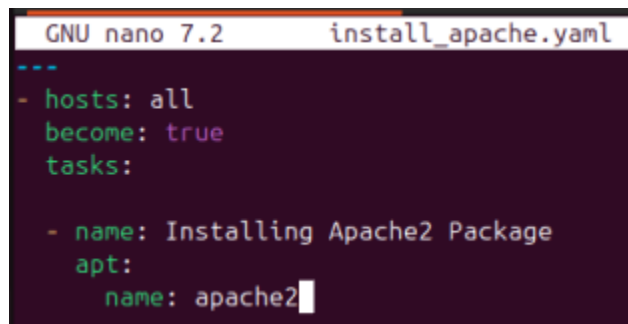
When the editor appears, type the following:



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

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

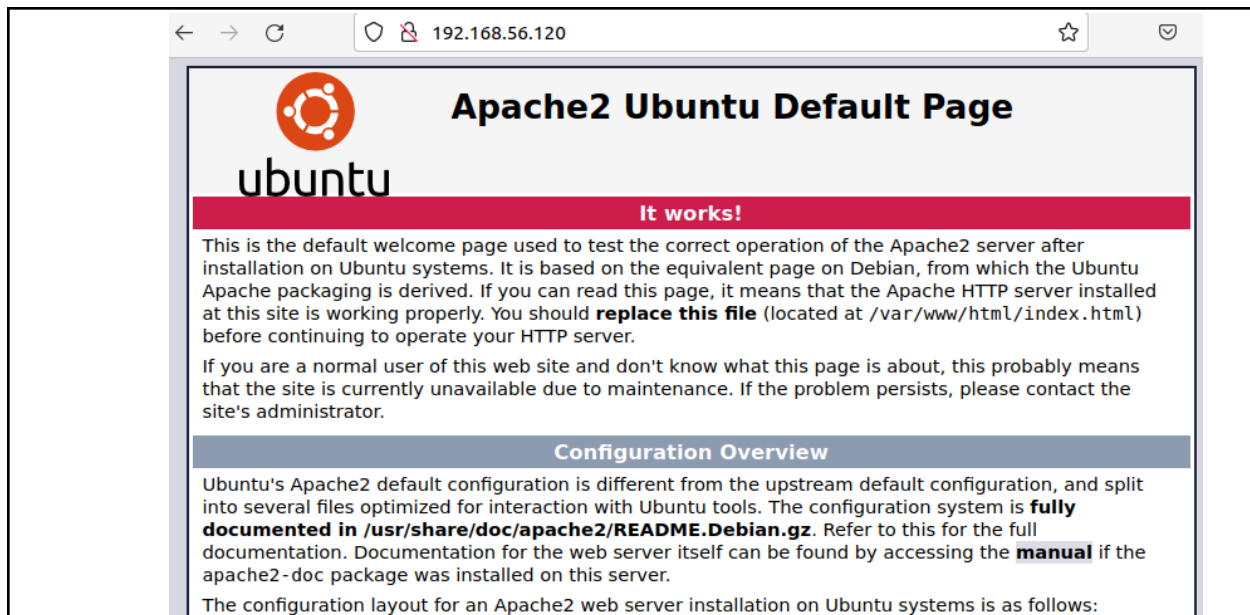
Make sure to save the file. Take note also of the alignments of the texts.



```
GNU nano 7.2      install_apache.yml
---
- hosts: all
  become: true
  tasks:

  - name: Installing Apache2 Package
    apt:
      name: apache2
```

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



```
ubuntu-gui@Workstation:~/AnsibleFiles$ ansible-playbook install_apache.yaml -K
BECOME password:

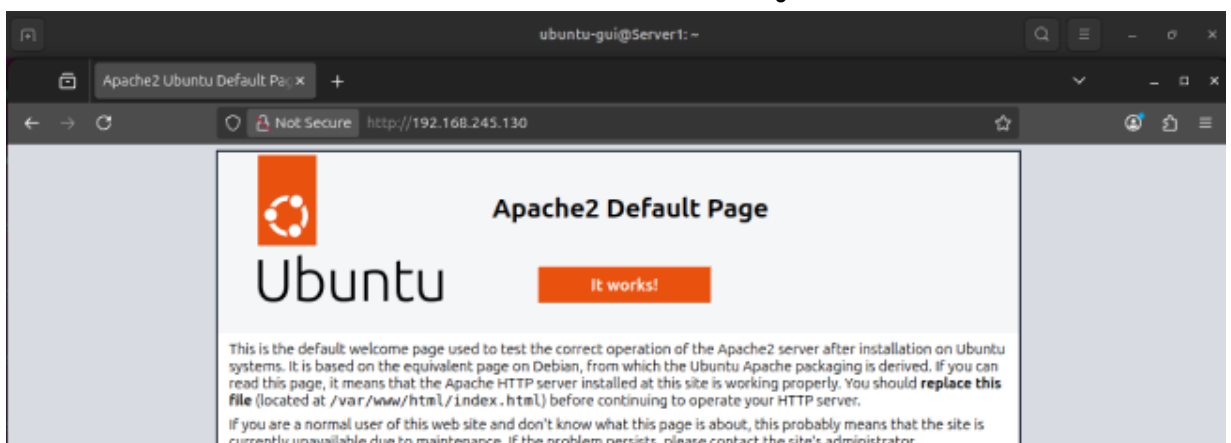
PLAY [all] *****

TASK [Gathering Facts] *****
fatal: [192.168.88.16]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: ssh: connect t
o host 192.168.88.16 port 22: No route to host", "unreachable": true}
fatal: [192.168.88.10]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: ssh: connect t
o host 192.168.88.10 port 22: No route to host", "unreachable": true}
ok: [192.168.245.130]
ok: [192.168.245.131]

TASK [Installing Apache2 Package] *****
changed: [192.168.245.131]
changed: [192.168.245.130]

PLAY RECAP *****
192.168.245.130      : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
192.168.245.131      : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
192.168.88.10       : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0
192.168.88.16       : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0
```

Have failures since 2 IP addresses are meant for Bridged connections in VM





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?

```
GNU nano 7.2  install_apache.yml
--
hosts: all
become: true
tasks:
- name: Installing Apache2 Package
  apt:
    name: mariadb # should be mariadb-server
```

```
ubuntu-gui@Workstation:~/AnsibleFiles$ ansible-playbook install_apache.yml -K
BECOME password:

PLAY [all] *****

TASK [Gathering Facts] *****
fatal: [192.168.88.10]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: ssh: connect t
o host 192.168.88.10 port 22: No route to host", "unreachable": true}
fatal: [192.168.88.16]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: ssh: connect t
o host 192.168.88.16 port 22: No route to host", "unreachable": true}
ok: [192.168.245.130]
ok: [192.168.245.131]

TASK [Installing Apache2 Package] *****
fatal: [192.168.245.131]: FAILED! => {"changed": false, "msg": "No package matching 'mariadb' is available"}
fatal: [192.168.245.130]: FAILED! => {"changed": false, "msg": "No package matching 'mariadb' is available"}

PLAY RECAP *****
192.168.245.130      : ok=1    changed=0    unreachable=0    failed=1    skipped=0    rescued=0    ignored=0
192.168.245.131      : ok=1    changed=0    unreachable=0    failed=1    skipped=0    rescued=0    ignored=0
192.168.88.10       : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0
192.168.88.16       : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0

ubuntu-gui@Workstation:~/AnsibleFiles$
```

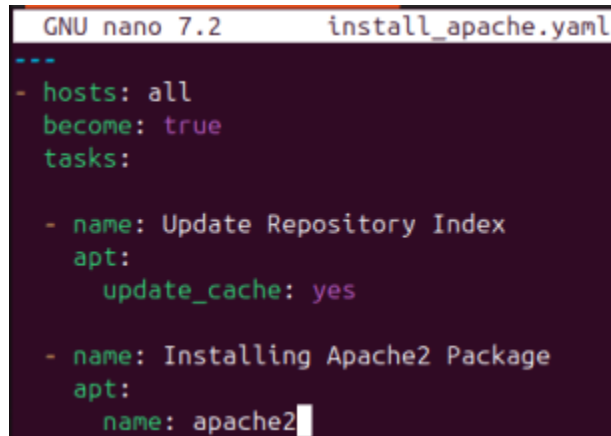
5. This time, we are going to put additional task to our playbook. Edit the *install_apache.yml*. As you can see, we are now adding an additional command, which is the *update_cache*. This command updates existing package-indexes on a supporting distro but not upgrading installed-packages (utilities) 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 7.2  install_apache.yml
---
- hosts: all
  become: true
  tasks:

    - name: Update Repository Index
      apt:
        update_cache: yes

    - name: Installing Apache2 Package
      apt:
        name: apache2
```

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

```

ubuntu-gui@Workstation:~/AnsibleFiles$ ansible-playbook install_apache.yml -K
BECOME password:

PLAY [all] *****

TASK [Gathering Facts] *****
fatal: [192.168.88.10]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: ssh: connect t
o host 192.168.88.10 port 22: No route to host", "unreachable": true}
fatal: [192.168.88.16]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: ssh: connect t
o host 192.168.88.16 port 22: No route to host", "unreachable": true}
ok: [192.168.245.130]
ok: [192.168.245.131]

TASK [Update Repository Index] *****
changed: [192.168.245.130]
changed: [192.168.245.131]

TASK [Installing Apache2 Package] *****
ok: [192.168.245.130]
ok: [192.168.245.131]

PLAY RECAP *****
192.168.245.130      : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
192.168.245.131      : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
192.168.88.10       : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0
192.168.88.16       : ok=0    changed=0    unreachable=1    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 7.2      install_apache.yaml
---
- hosts: all
  become: true
  tasks:

    - name: Update Repository Index
      apt:
        update_cache: yes

    - name: Installing Apache2 Package
      apt:
        name: apache2

    - name: Add PHP Support for Apache2
      apt:
        name: libapache2-mod-php

```

8. Run the playbook and describe the output. Did the new command change anything on the remote servers?
9. Finally, make sure that we are in sync with GitHub. Provide the link of your GitHub repository.

```

ubuntu-gui@Workstation:~/AnsibleFiles$ cp -v ansible.cfg inventory.yaml install_apache.yaml ~/CPE212_JeanGabrielVincentRoallos/
'ansible.cfg' -> '/home/ubuntu-gui/CPE212_JeanGabrielVincentRoallos/ansible.cfg'
'inventory.yaml' -> '/home/ubuntu-gui/CPE212_JeanGabrielVincentRoallos/inventory.yaml'
'install_apache.yaml' -> '/home/ubuntu-gui/CPE212_JeanGabrielVincentRoallos/install_apache.yaml'
ubuntu-gui@Workstation:~/AnsibleFiles$ cd ..; cd CPE212_JeanGabrielVincentRoallos
ubuntu-gui@Workstation:~/CPE212_JeanGabrielVincentRoallos$ ls
ansible.cfg  install_apache.yaml  inventory.yaml  README.md
ubuntu-gui@Workstation:~/CPE212_JeanGabrielVincentRoallos$ git add ansible.cfg; git commit -m "Config file for Ansible"
[main 6b3a1b2] Config file for Ansible
1 file changed, 3 insertions(+)
create mode 100644 ansible.cfg
ubuntu-gui@Workstation:~/CPE212_JeanGabrielVincentRoallos$ git add inventory.yaml; git commit -m "Inventory file for Ansible"
[main 3784175] Inventory file for Ansible
1 file changed, 19 insertions(+)
create mode 100644 inventory.yaml
ubuntu-gui@Workstation:~/CPE212_JeanGabrielVincentRoallos$ git add install_apache.yaml; git commit -m "Sample Ansible playbook used in HOA #4"
[main 31d0f9b] Sample Ansible playbook used in HOA #4
1 file changed, 16 insertions(+)
create mode 100644 install_apache.yaml

```

https://github.com/jgvroallos/CPE212_JeanGabrielVincentRoallos

Reflections:

Answer the following:

1. What is the importance of using a playbook?

This centralizes the execution of commands from one source (control node running Ansible playbook) to multiple nodes, instead of manually visiting each node to enter the same lines of commands to each.

2. Summarize what we have done on this activity.

This laboratory activity mostly showed the accepted syntaxes for an Ansible playbook for installing applications/services, making sure we obtain the latest versions of said applications/services.