

<b>Name:</b> Ramirez, Kiel Louis A.	<b>Date Performed:</b> 9-5-25
<b>Course/Section:</b> <a href="#">CPE 212-CPE31S2</a>	<b>Date Submitted:</b> 9-5-25
<b>Instructor:</b> Sir Robin	<b>Semester and SY:</b> 1st 2025-2026
<b>Activity 4: Running Elevated Ad hoc Commands</b>	
<b>1. Objectives:</b> 1.1 Use commands that makes changes to remote machines 1.2 Use playbook in automating ansible commands	
<b>2. Discussion:</b>  <i>Provide screenshots for each task.</i>  <b>Elevated Ad hoc commands</b> So far, we have not performed ansible commands that makes changes to the remote servers. We manage to gather facts and connect to the remote machines, but we still did not make changes on those machines. In this activity, we will learn to use commands that would install, update, and upgrade packages in the remote machines. We will also create a playbook that will be used for automations.  <b>Playbooks</b> record and execute <b>Ansible's</b> configuration, deployment, and orchestration functions. They can describe a policy you want your remote systems to enforce, or a set of steps in a general IT process. If Ansible modules are the tools in your workshop, playbooks are your instruction manuals, and your inventory of hosts are your raw material. At a basic level, playbooks can be used to manage configurations of and deployments to remote machines. At a more advanced level, they can sequence multi-tier rollouts involving rolling updates, and can delegate actions to other hosts, interacting with monitoring servers and load balancers along the way. You can check this documentation if you want to learn more about playbooks. <a href="#">Working with playbooks — Ansible Documentation</a>	
<b>Task 1: Run elevated ad hoc commands</b>  1. Locally, we use the command <i>sudo apt update</i> when we want to download package information from all configured resources. The sources often defined in <i>/etc/apt/sources.list</i> file and other files located in <i>/etc/apt/sources.list.d/</i> directory. So, when you run update command, it downloads the package information from the Internet. It is useful to get info on an updated version of packages or their dependencies. We can only run	

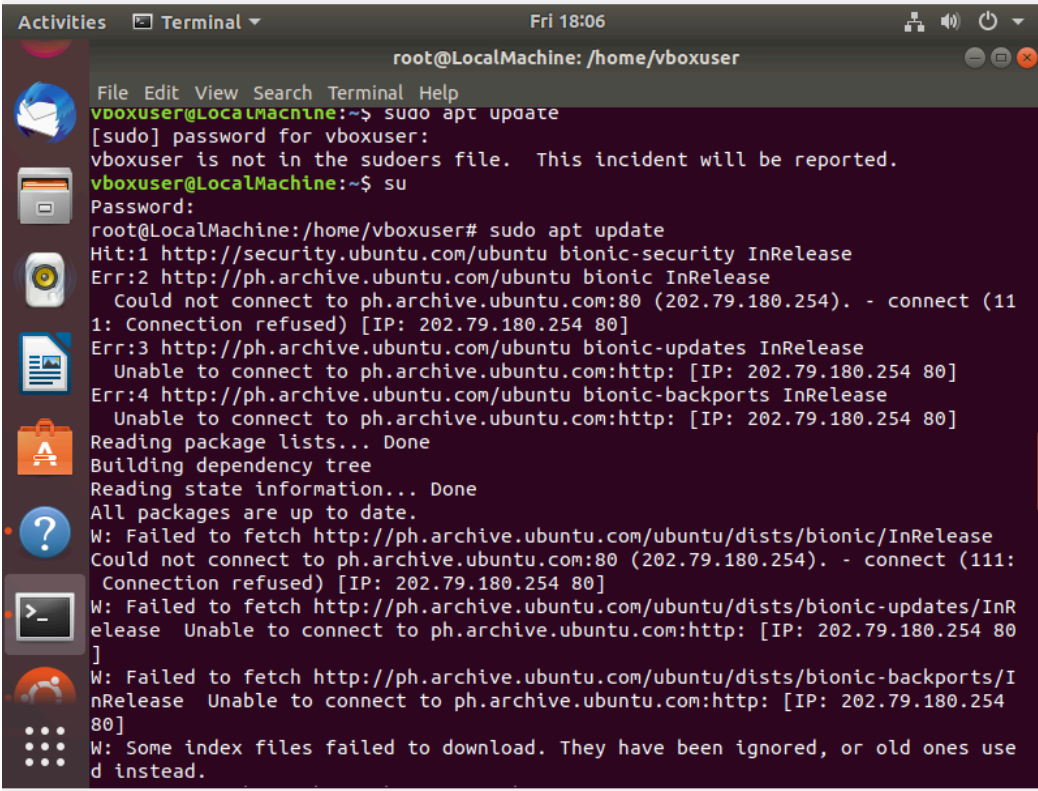
an apt update command in a remote machine. Issue the following command:

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

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

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 Fri 18:06
root@LocalMachine: /home/vboxuser
File Edit View Search Terminal Help
vboxuser@LocalMachine:~$ sudo apt update
[sudo] password for vboxuser:
vboxuser is not in the sudoers file. This incident will be reported.
vboxuser@LocalMachine:~$ su
Password:
root@LocalMachine:/home/vboxuser# sudo apt update
Hit:1 http://security.ubuntu.com/ubuntu bionic-security InRelease
Err:2 http://ph.archive.ubuntu.com/ubuntu bionic InRelease
  Could not connect to ph.archive.ubuntu.com:80 (202.79.180.254). - connect (11
1: Connection refused) [IP: 202.79.180.254 80]
Err:3 http://ph.archive.ubuntu.com/ubuntu bionic-updates InRelease
  Unable to connect to ph.archive.ubuntu.com:http: [IP: 202.79.180.254 80]
Err:4 http://ph.archive.ubuntu.com/ubuntu bionic-backports InRelease
  Unable to connect to ph.archive.ubuntu.com:http: [IP: 202.79.180.254 80]
Reading package lists... Done
Building dependency tree
Reading state information... Done
All packages are up to date.
W: Failed to fetch http://ph.archive.ubuntu.com/ubuntu/dists/bionic/InRelease
  Could not connect to ph.archive.ubuntu.com:80 (202.79.180.254). - connect (111:
  Connection refused) [IP: 202.79.180.254 80]
W: Failed to fetch http://ph.archive.ubuntu.com/ubuntu/dists/bionic-updates/InR
  elease Unable to connect to ph.archive.ubuntu.com:http: [IP: 202.79.180.254
  80]
W: Failed to fetch http://ph.archive.ubuntu.com/ubuntu/dists/bionic-backports/I
  nRelease Unable to connect to ph.archive.ubuntu.com:http: [IP: 202.79.180.254
  80]
W: Some index files failed to download. They have been ignored, or old ones use
  d instead.
```

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.

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

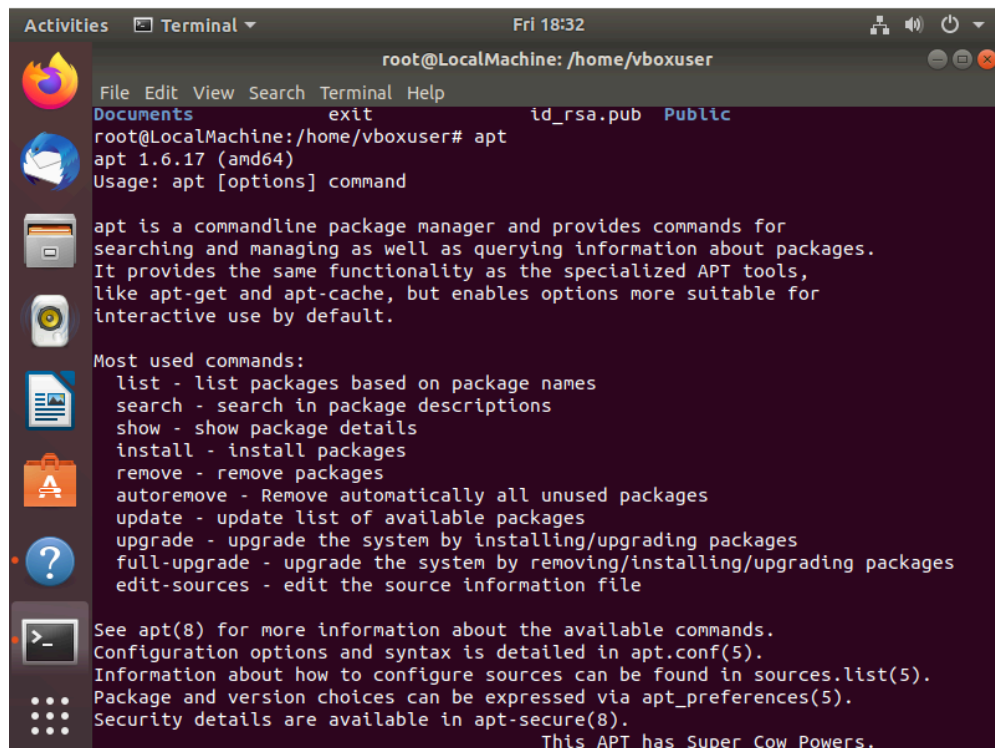
```
root@LocalMachine:/home/vboxuser# which vim
root@LocalMachine:/home/vboxuser# apt search vim-nox
Sorting... Done
Full Text Search... Done
vim-nox/bionic-updates,bionic-security 2:8.0.1453-1ubuntu1.13 amd64
Vi IMproved - enhanced vi editor - with scripting languages support

vim-tiny/bionic-updates,bionic-security,now 2:8.0.1453-1ubuntu1.13 amd64 [insta
lled]
Vi IMproved - enhanced vi editor - compact version

root@LocalMachine:/home/vboxuser#
```

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*.

the version of the apt



```
Activities  Terminal  Fri 18:32
root@LocalMachine: /home/vboxuser

File Edit View Search Terminal Help
Documents  exit  id_rsa.pub Public
root@LocalMachine:/home/vboxuser# apt
apt 1.6.17 (amd64)
Usage: apt [options] command

apt is a commandline package manager and provides commands for
searching and managing as well as querying information about packages.
It provides the same functionality as the specialized APT tools,
like apt-get and apt-cache, but enables options more suitable for
interactive use by default.

Most used commands:
list - list packages based on package names
search - search in package descriptions
show - show package details
install - install packages
remove - remove packages
autoremove - Remove automatically all unused packages
update - update list of available packages
upgrade - upgrade the system by installing/upgrading packages
full-upgrade - upgrade the system by removing/installing/upgrading packages
edit-sources - edit the source information file

See apt(8) for more information about the available commands.
Configuration options and syntax is detailed in apt.conf(5).
Information about how to configure sources can be found in sources.list(5).
Package and version choices can be expressed via apt_preferences(5).
Security details are available in apt-secure(8).

This APT has Super Cow Powers.
```

3. This time, we will install a package called *snapt*. 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=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?

yes because it worked, but i dont have a host list

```
root@LocalMachine:/home/vboxuser# ansible all -m apt -a name=snapt --become --ask-become-pass
SUDO password:
[WARNING]: provided hosts list is empty, only localhost is available. Note that the implicit localhost does not match 'all'

root@LocalMachine:/home/vboxuser#
```

3.2 Now, try to issue this command: *ansible all -m apt -a "name=snapt 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.

```
root@LocalMachine:/home/vboxuser# ansible all -m apt -a "name=snapt state=latest" --become --ask-become-pass
SUDO password:
[WARNING]: provided hosts list is empty, only localhost is available. Note that the implicit localhost does not match 'all'

Show Applications
root@LocalMachine:/home/vboxuser#
```

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.

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

```

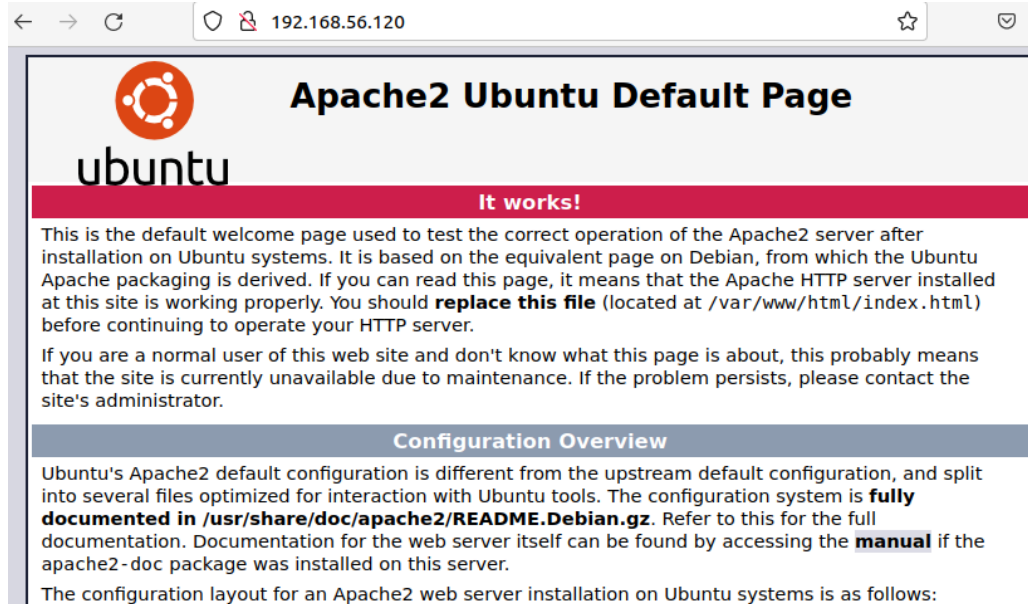
root@LocalMachine:/home/vboxuser# ansible-playbook --ask-become-pass install_ap
ache.yml
SUDO password:
[WARNING]: provided hosts list is empty, only localhost is available. Note
that the implicit localhost does not match 'all'

PLAY [all] *****
*
skipping: no hosts matched

PLAY RECAP *****
*
Show Applications
root@LocalMachine:/home/vboxuser#

```

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



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

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

```

root@LocalMachine:/home/vboxuser# ansible-playbook --ask-become-pass install_ap
ache.yml
SUDO password:
[WARNING]: provided hosts list is empty, only localhost is available. Note
that the implicit localhost does not match 'all'

PLAY [all] *****
*
skipping: no hosts matched

PLAY RECAP *****
*

root@LocalMachine:/home/vboxuser#

```

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.

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

```

root@LocalMachine:/home/vboxuser# ansible-playbook --ask-become-pass install_ap
ache.yml
SUDO password:
[WARNING]: provided hosts list is empty, only localhost is available. Note
that the implicit localhost does not match 'all'

PLAY [all] *****
*
skipping: no hosts matched

PLAY RECAP *****
*

root@LocalMachine:/home/vboxuser#

```

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

**Reflections:**

Answer the following:

1. What is the importance of using a playbook?  
the playbook can save all your list of files that would be just like your google drive
2. Summarize what we have done on this activity.

what i did was creating a playbook in an ansible, this would teach students to make there playbook in an ubuntu using ansible