Name: Rio Marie G. Suzuki	**Date Performed:** 09/28/2023
Course/Section: CPE232 S6	**Date Submitted:** 09/28/2023
Instructor: Dr. Jonathan Taylar	**Semester and SY:** 1st Sem 2023-2024

Activity 6: Targeting Specific Nodes and Managing Services

1. **Objectives:**

1.1 Individualize hosts
1.2 Apply tags in selecting plays to run
1.3 Managing Services from remote servers using playbooks

2. **Discussion**:

In this activity, we try to individualize hosts. For example, we don't want apache on all our servers, or maybe only one of our servers is a web server, or maybe we have different servers like database or file servers running different things on different categories of servers and that is what we are going to take a look at in this activity.

We also try to manage services that do not automatically run using the automations in playbook. For example, when we install web servers or httpd for CentOS, we notice that the service did not start automatically.

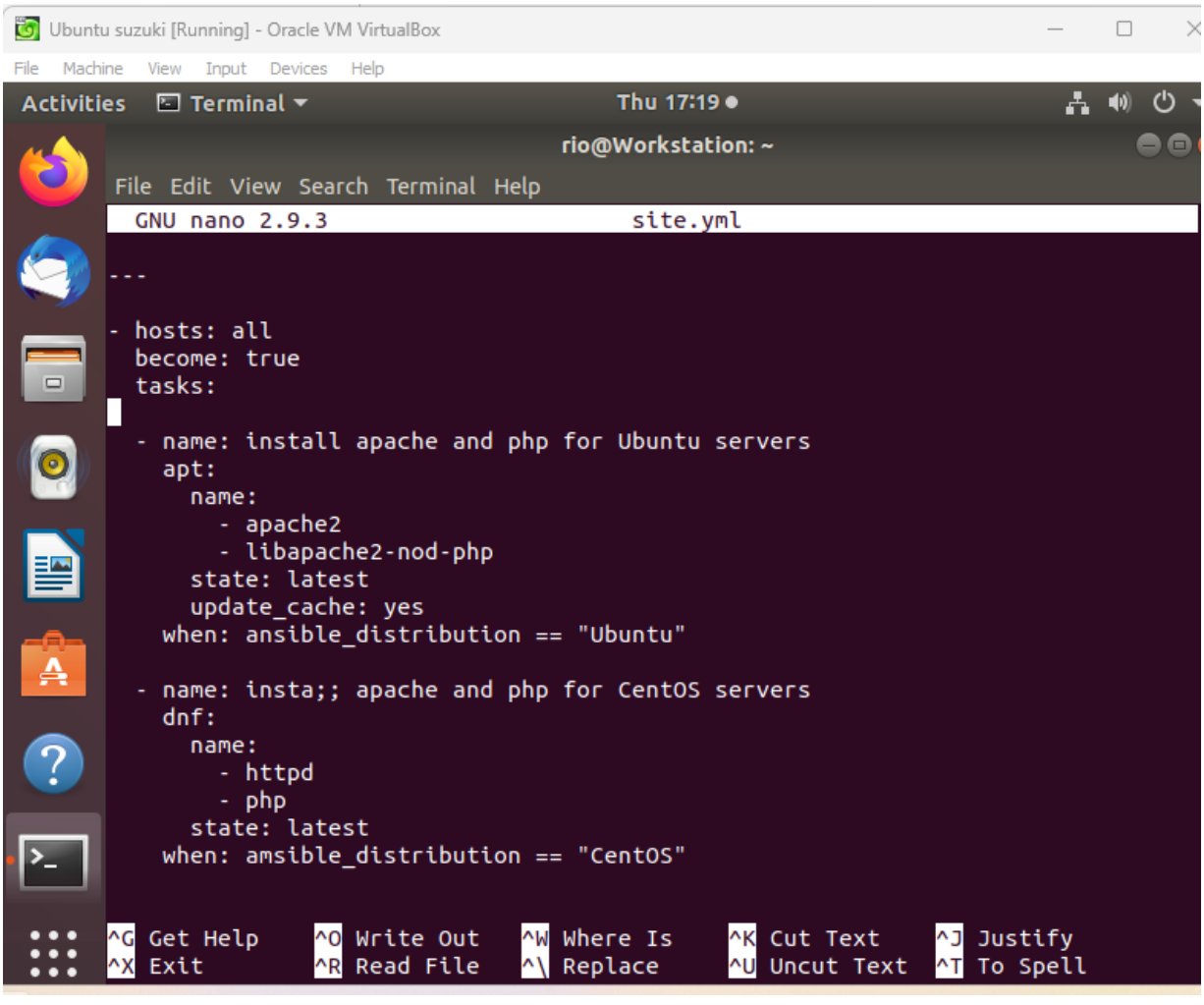
Requirement:

In this activity, you will need to create another Ubuntu VM and name it Server 3. Likewise, you need to activate the second adapter to a host-only adapter after the installations. Take note of the IP address of the Server 3. Make sure to use the command *ssh-copy-id* to copy the public key to Server 3. Verify if you can successfully SSH to Server 3.

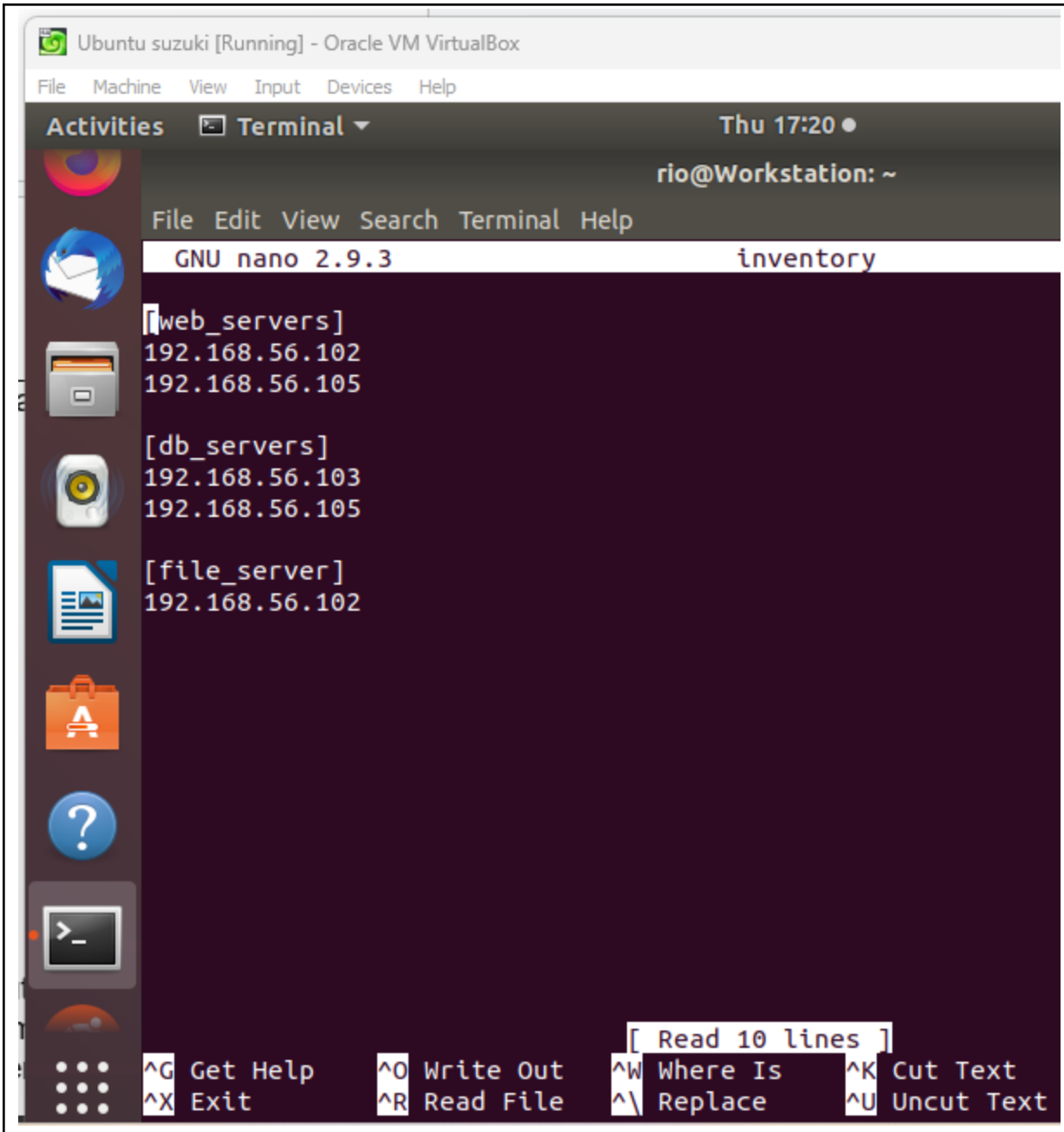
Task 1: Targeting Specific Nodes

1. Create a new playbook and named it site.yml. Follow the commands as shown in the image below. Make sure to save the file and exit.

```
---

- hosts: all
  become: true
  tasks:

  - name: install apache and php for Ubuntu servers
    apt:
      name:
        - apache2
        - libapache2-mod-php
      state: latest
      update_cache: yes
    when: ansible_distribution == "Ubuntu"

  - name: install apache and php for CentOS servers
    dnf:
      name:
        - httpd
        - php
      state: latest
    when: ansible_distribution == "CentOS"
```



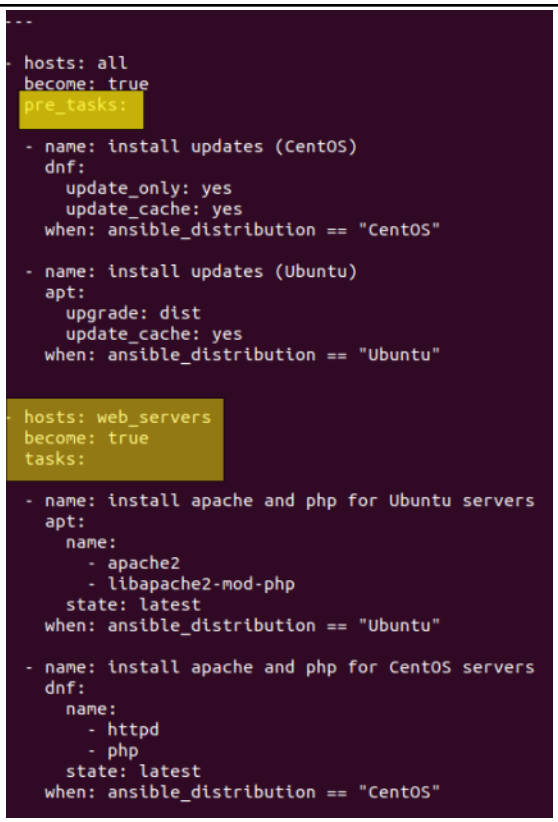
2. Edit the inventory file. Remove the variables we put in our last activity and group according to the image shown below:



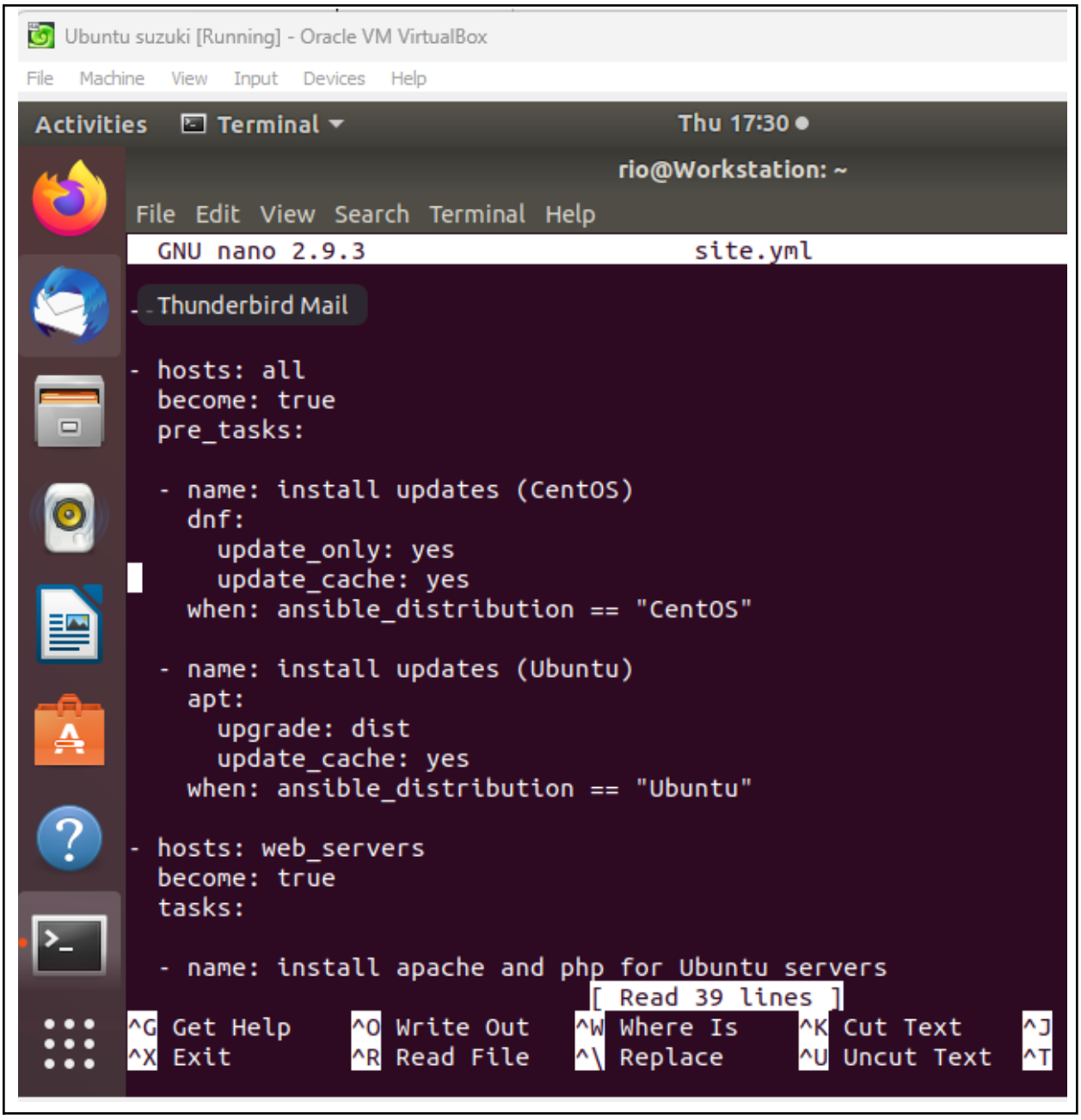
Make sure to save the file and exit.

Right now, we have created groups in our inventory file and put each server in its own group. In other cases, you can have a server be a member of multiple groups, for example you have a test server that is also a web server.

3. Edit the *site.yml* by following the image below:



```
---

- hosts: all
  become: true
  pre_tasks:

  - name: install updates (CentOS)
    dnf:
      update_only: yes
      update_cache: yes
    when: ansible_distribution == "CentOS"

  - name: install updates (Ubuntu)
    apt:
      upgrade: dist
      update_cache: yes
    when: ansible_distribution == "Ubuntu"


- hosts: web_servers
  become: true
  tasks:

  - name: install apache and php for Ubuntu servers
    apt:
      name:
        - apache2
        - libapache2-mod-php
      state: latest
    when: ansible_distribution == "Ubuntu"

  - name: install apache and php for CentOS servers
    dnf:
      name:
        - httpd
        - php
      state: latest
    when: ansible_distribution == "CentOS"
```

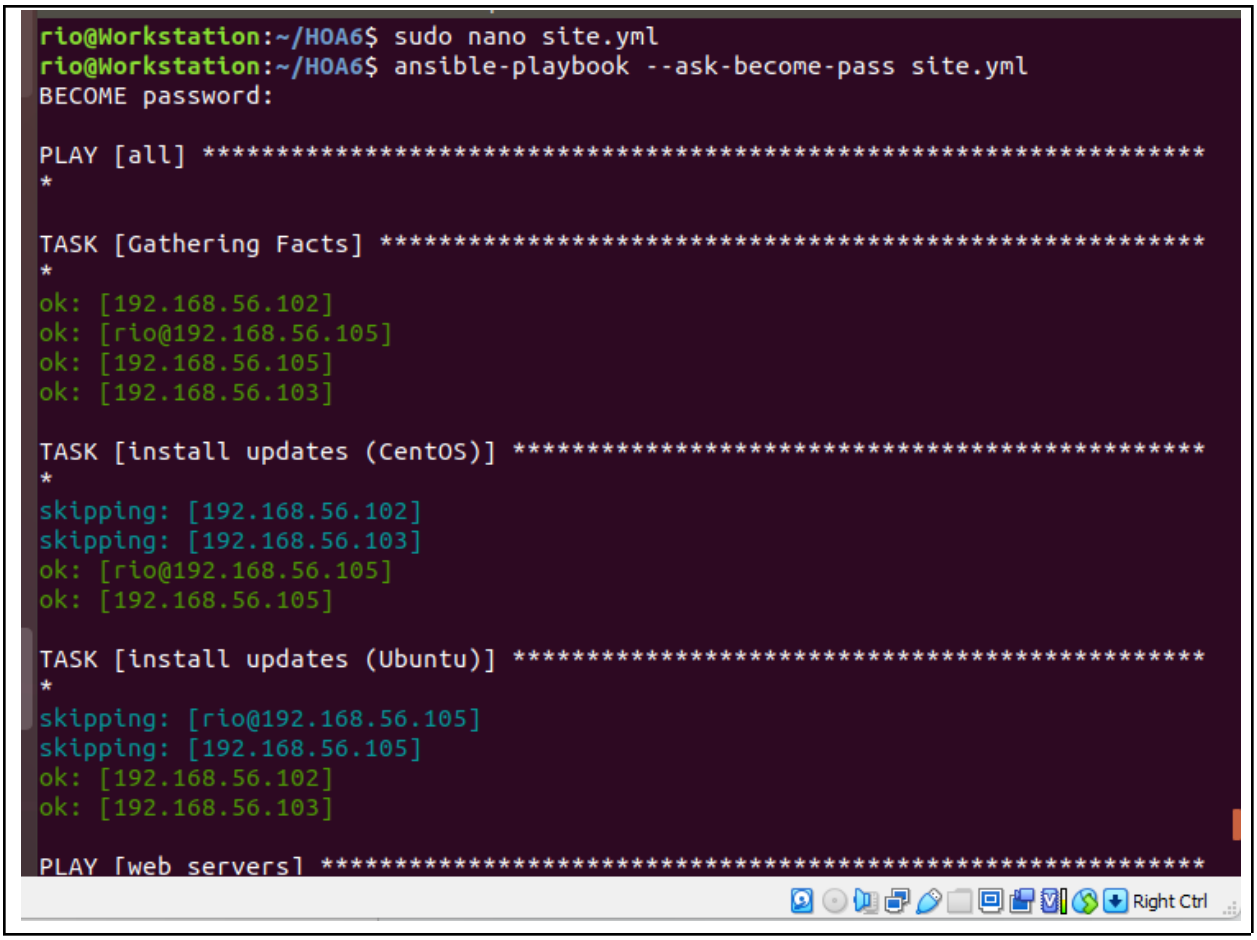

Ubuntu suzuki [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Terminal
Thu 17:30
rio@Workstation: ~
File Edit View Search Terminal Help
GNU nano 2.9.3 site.yml
Thunderbird Mail
- hosts: all
become: true
pre_tasks:
- name: install updates (CentOS)
dnf:
update_only: yes
update_cache: yes
when: ansible_distribution == "CentOS"
- name: install updates (Ubuntu)
apt:
upgrade: dist
update_cache: yes
when: ansible_distribution == "Ubuntu"
- hosts: web_servers
become: true
tasks:
- name: install apache and php for Ubuntu servers
[Read 39 lines]
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T

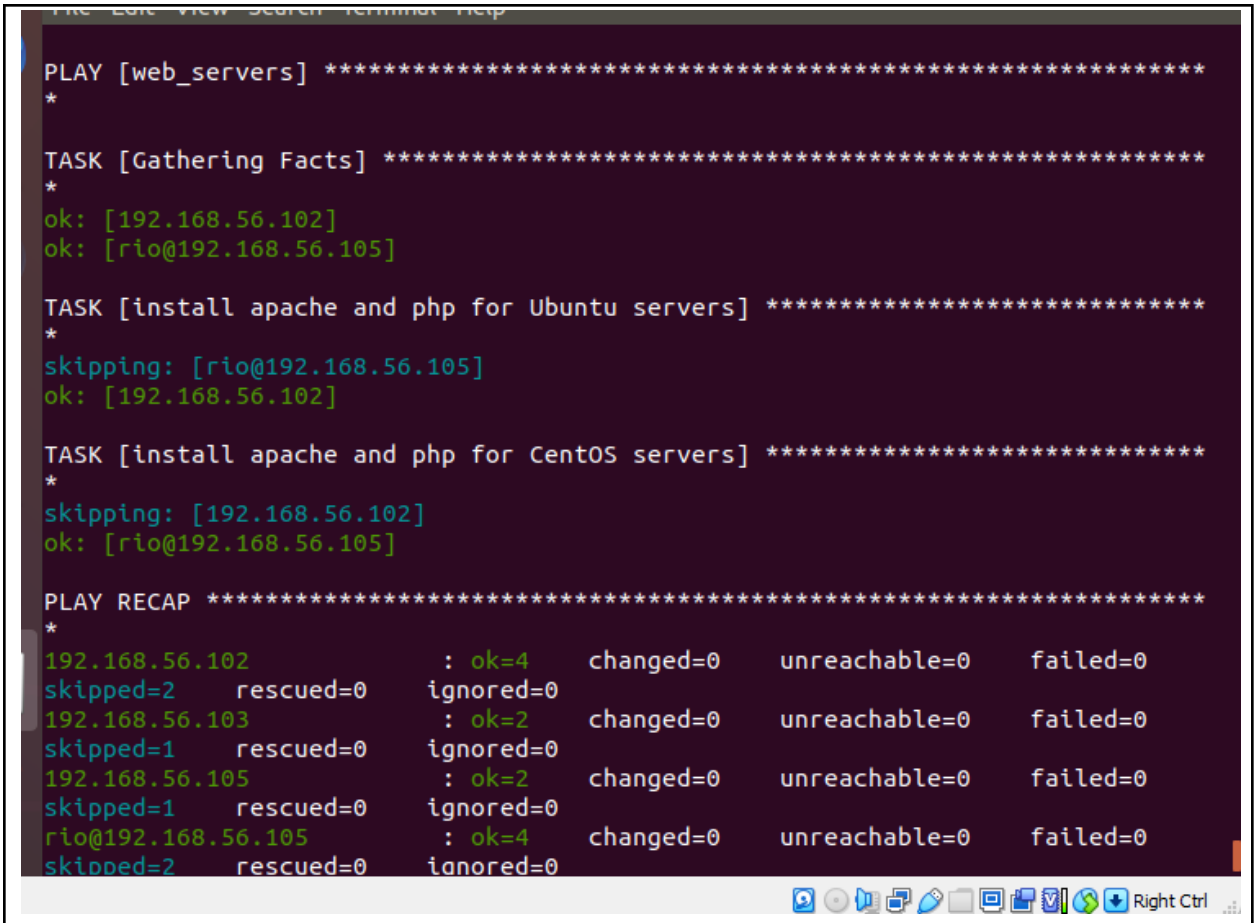
```
  - name: install apache and php for Ubuntu servers
    apt:
      name:
        - apache2
        - libapache2-nod-php
      state: latest
#      update_cache: yes
    when: ansible_distribution == "Ubuntu"

  - name: install apache and php for CentOS servers
    dnf:
      name:
        - httpd

^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify
^X Exit        ^R Read File   ^\ Replace     ^U Uncut Text  ^T To Spell
```

Make sure to save the file and exit.

The *pre-tasks* command tells the ansible to run it before any other thing. In the *pre-tasks*, CentOS will install updates while Ubuntu will upgrade its distribution package. This will run before running the second play, which is targeted at *web_servers*. In the second play, apache and php will be installed on both Ubuntu servers and CentOS servers.

Run the *site.yml* file and describe the result.


rio@Workstation:~/HOA6$ sudo nano site.yml
rio@Workstation:~/HOA6$ ansible-playbook --ask-become-pass site.yml
BECOME password:
PLAY [all] **
*
TASK [Gathering Facts] **
*
ok: [192.168.56.102]
ok: [rio@192.168.56.105]
ok: [192.168.56.105]
ok: [192.168.56.103]
TASK [install updates (CentOS)] ***
*
skipping: [192.168.56.102]
skipping: [192.168.56.103]
ok: [rio@192.168.56.105]
ok: [192.168.56.105]
TASK [install updates (Ubuntu)] ***
*
skipping: [rio@192.168.56.105]
skipping: [192.168.56.105]
ok: [192.168.56.102]
ok: [192.168.56.103]
PLAY [web servers] **
Right Ctrl

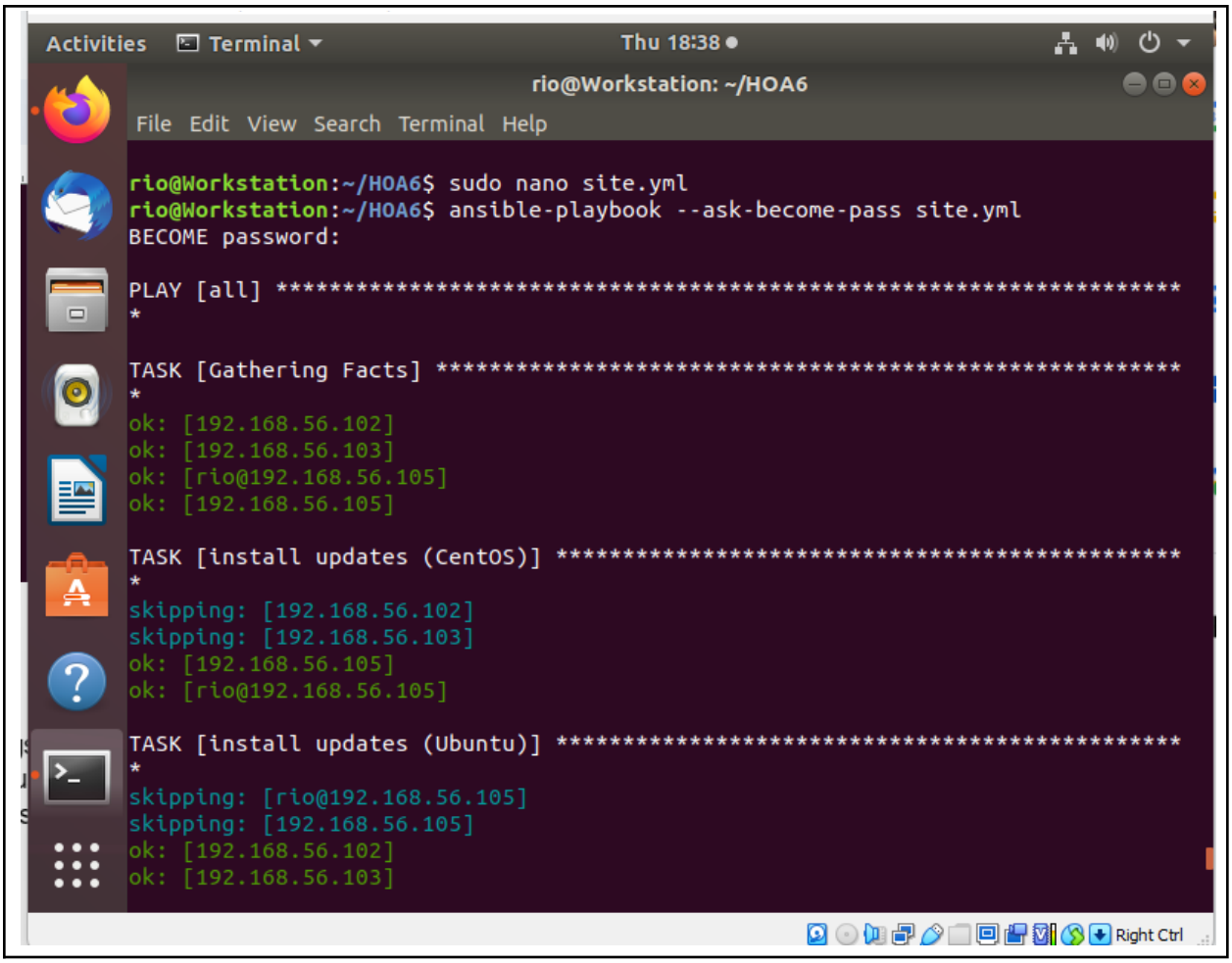


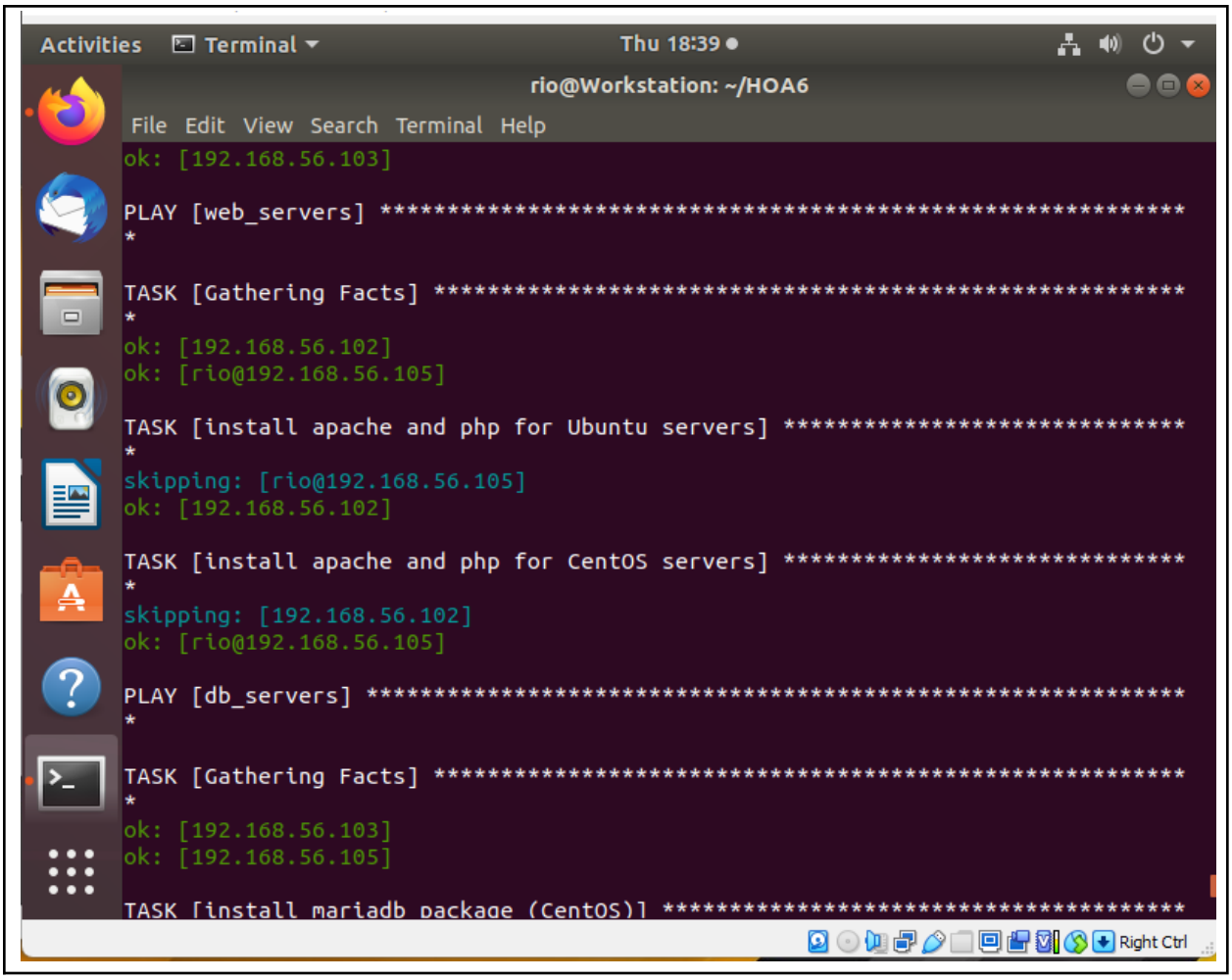
4. Let's try to edit again the *site.yml* file. This time, we are going to add plays targeting the other servers. This time we target the *db_servers* by adding it on the current *site.yml*. Below is an example: (Note add this at the end of the playbooks from task 1.3.

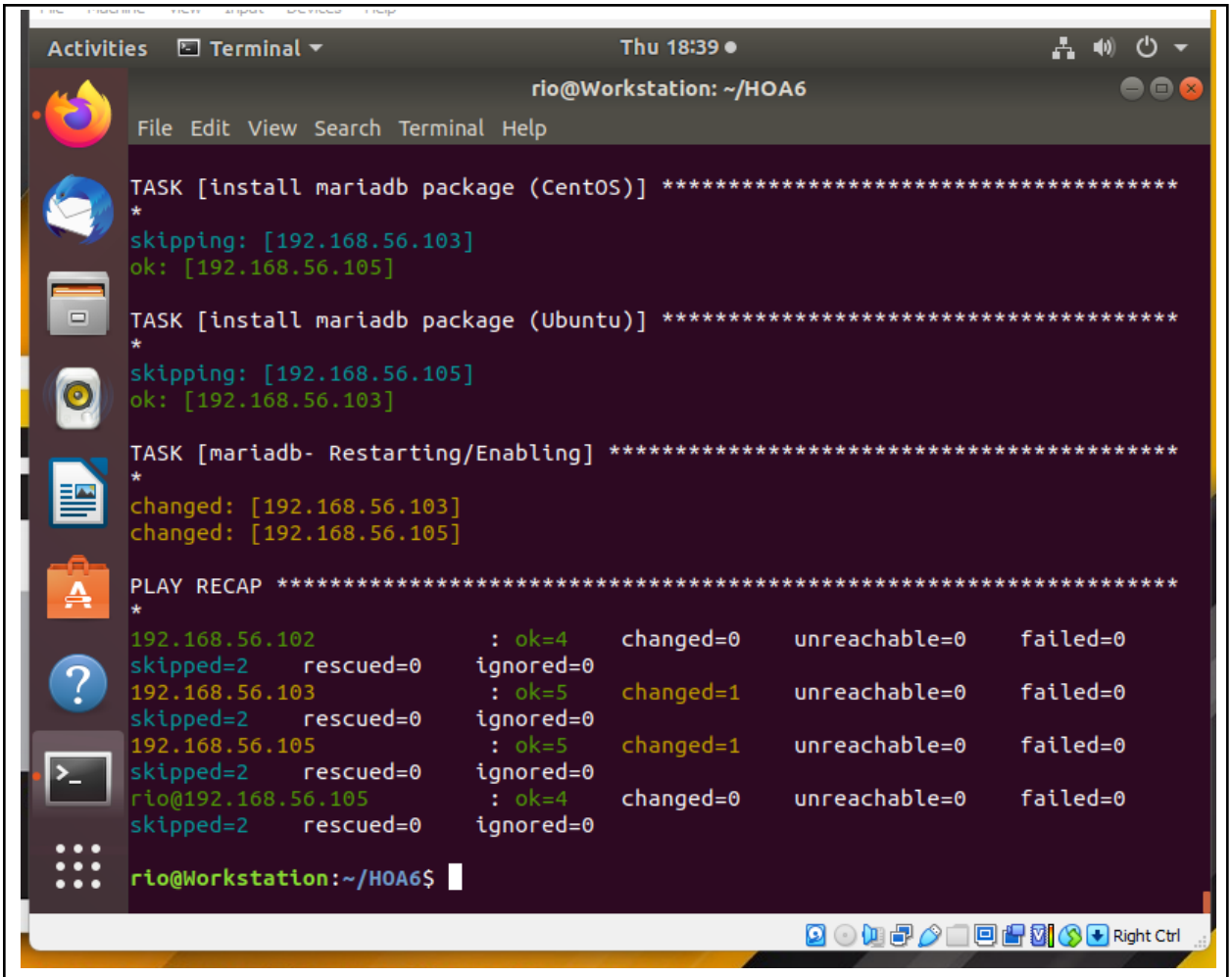
```
- hosts: db_servers
  become: true
  tasks:

  - name: install mariadb package (CentOS)
    yum:
      name: mariadb-server
      state: latest
    when: ansible_distribution == "CentOS"

  - name: "Mariadb- Restarting/Enabling"
    service:
      name: mariadb
      state: restarted
      enabled: true

  - name: install mariadb packege (Ubuntu)
    apt:
      name: mariadb-server
      state: latest
    when: ansible_distribution == "Ubuntu"
```

Make sure to save the file and exit.

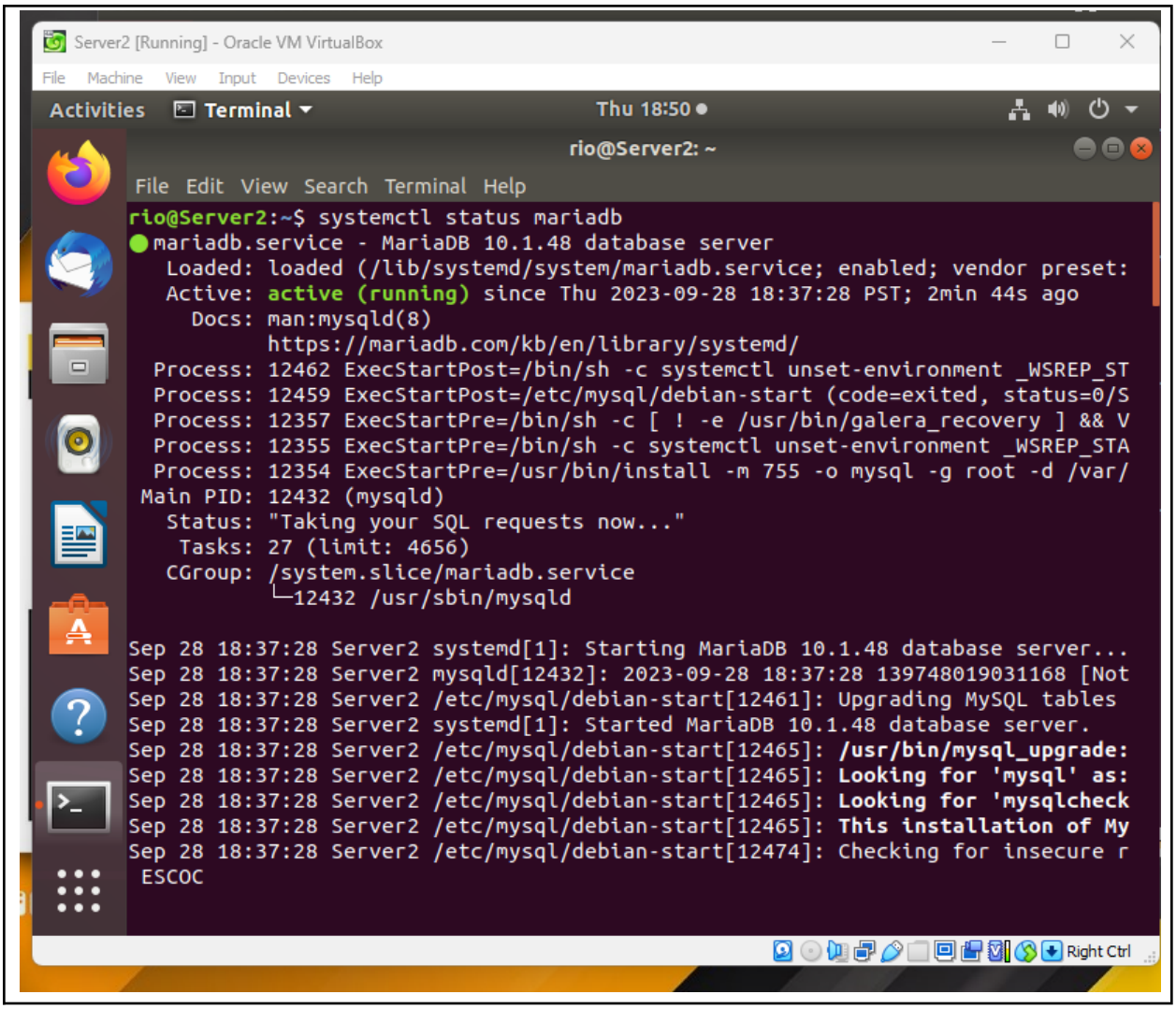
Run the *site.yml* file and describe the result.

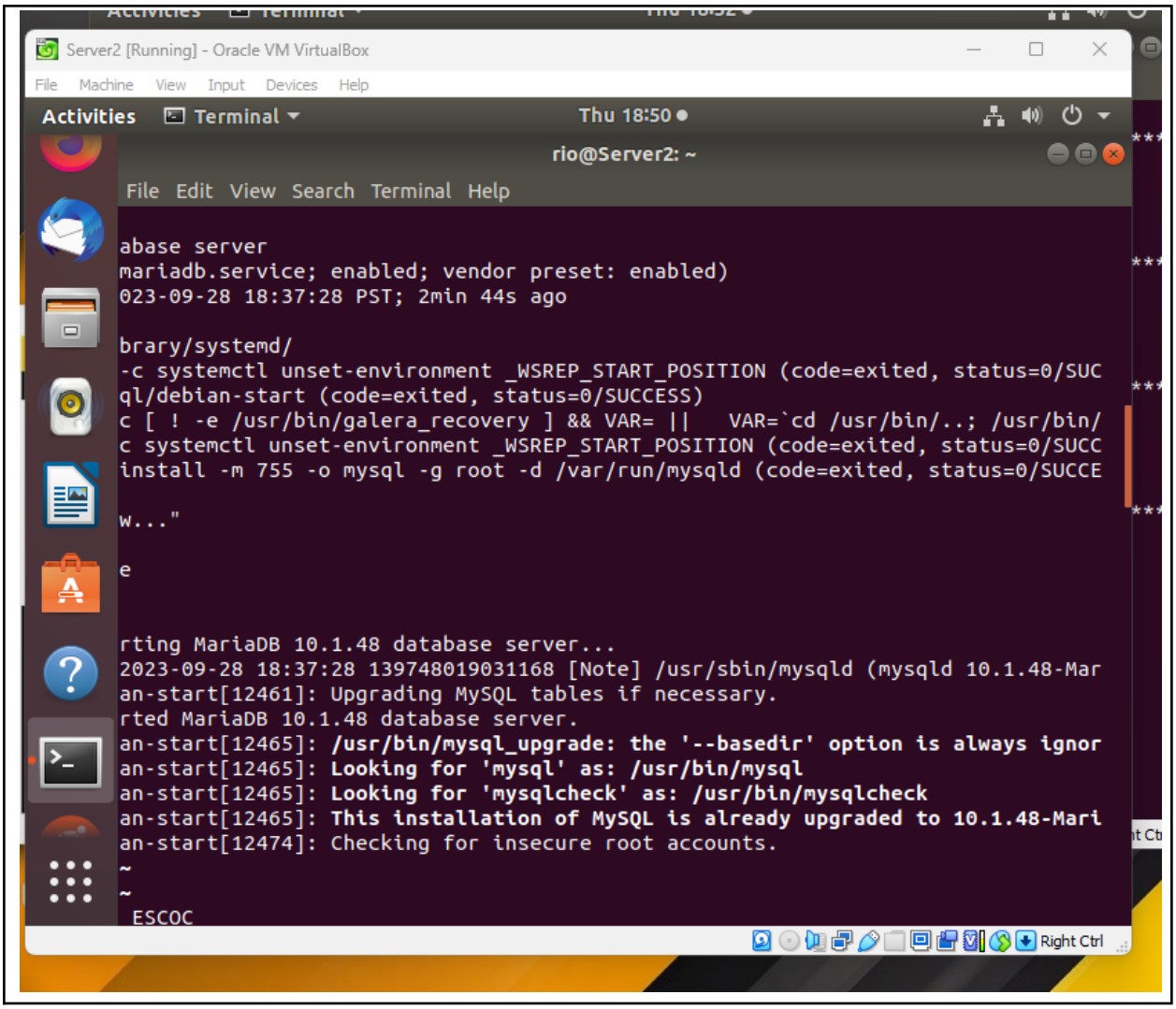

Activities
Terminal
Thu 18:38
rio@Workstation: ~/HOA6
File Edit View Search Terminal Help
rio@Workstation:~/HOA6$ sudo nano site.yml
rio@Workstation:~/HOA6$ ansible-playbook --ask-become-pass site.yml
BECOME password:
PLAY [all] ***
*
TASK [Gathering Facts] ***
*
ok: [192.168.56.102]
ok: [192.168.56.103]
ok: [rio@192.168.56.105]
ok: [192.168.56.105]
TASK [install updates (CentOS)] **
*
skipping: [192.168.56.102]
skipping: [192.168.56.103]
ok: [192.168.56.105]
ok: [rio@192.168.56.105]
TASK [install updates (Ubuntu)] **
*
skipping: [rio@192.168.56.105]
skipping: [192.168.56.105]
ok: [192.168.56.102]
ok: [192.168.56.103]
Right Ctrl

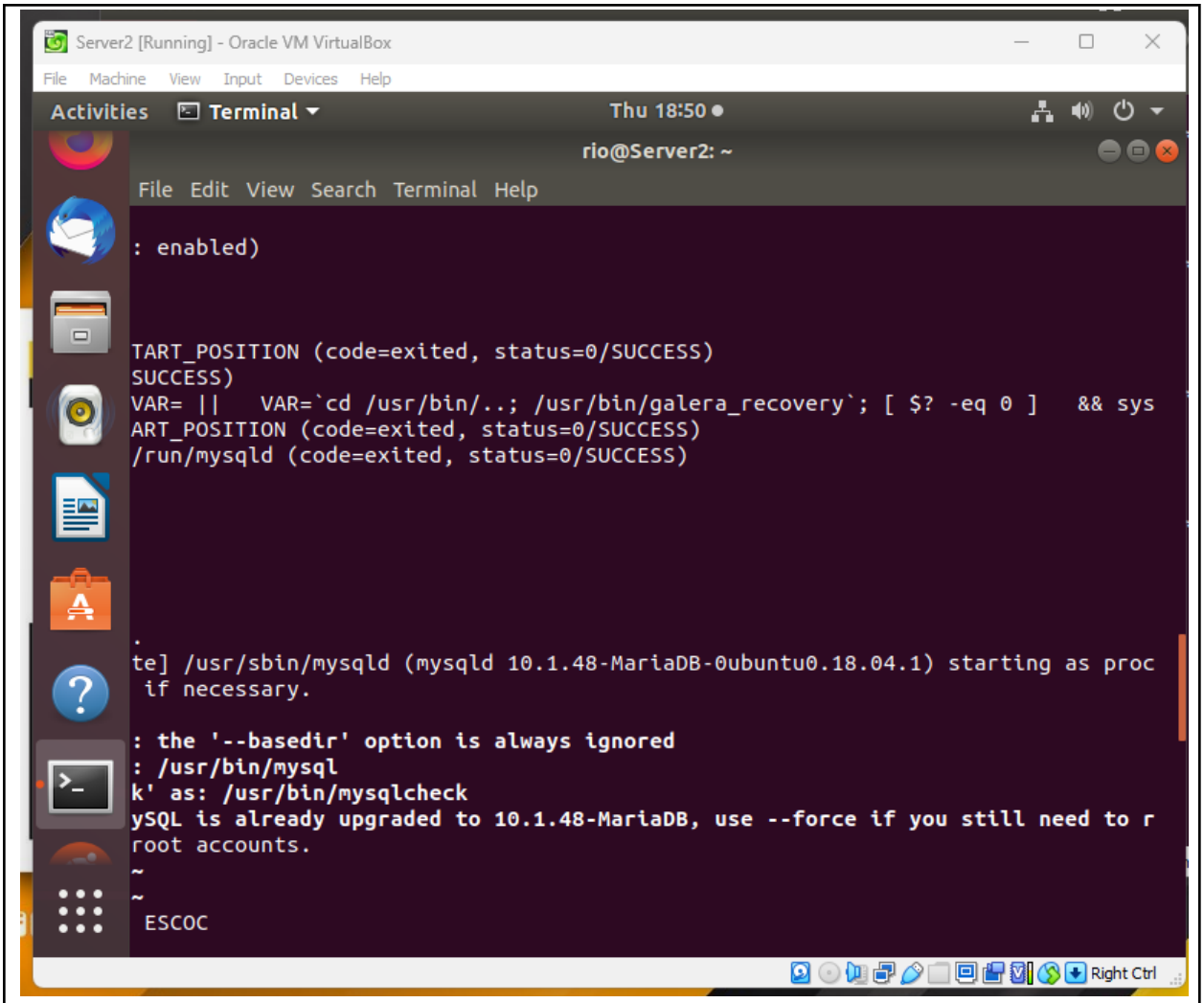

Activities
Terminal
Thu 18:39
rio@Workstation: ~/HOA6
File Edit View Search Terminal Help
ok: [192.168.56.103]
PLAY [web_servers] **
*
TASK [Gathering Facts] **
*
ok: [192.168.56.102]
ok: [rio@192.168.56.105]
TASK [install apache and php for Ubuntu servers] **********************************
*
skipping: [rio@192.168.56.105]
ok: [192.168.56.102]
TASK [install apache and php for CentOS servers] **********************************
*
skipping: [192.168.56.102]
ok: [rio@192.168.56.105]
PLAY [db_servers] ***
*
TASK [Gathering Facts] **
*
ok: [192.168.56.103]
ok: [192.168.56.105]
TASK [install mariadb package (CentOS)] ***
Right Ctrl



5. Go to the remote server (Ubuntu) terminal that belongs to the db_servers group and check the status for mariadb installation using the command: *systemctl status mariadb.* Do this on the CentOS server also.


Server2 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Terminal
Thu 18:50
rio@Server2: ~
File Edit View Search Terminal Help
rio@Server2:~$ systemctl status mariadb
mariadb.service - MariaDB 10.1.48 database server
Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor preset:
Active: active (running) since Thu 2023-09-28 18:37:28 PST; 2min 44s ago
Docs: man:mysqld(8)
https://mariadb.com/kb/en/library/systemd/
Process: 12462 ExecStartPost=/bin/sh -c systemctl unset-environment _WSREP_ST
Process: 12459 ExecStartPost=/etc/mysql/debian-start (code=exited, status=0/S
Process: 12357 ExecStartPre=/bin/sh -c [! -e /usr/bin/galera_recovery] && V
Process: 12355 ExecStartPre=/bin/sh -c systemctl unset-environment _WSREP_STA
Process: 12354 ExecStartPre=/usr/bin/install -m 755 -o mysql -g root -d /var/
Main PID: 12432 (mysqld)
Status: "Taking your SQL requests now..."
Tasks: 27 (limit: 4656)
CGroup: /system.slice/mariadb.service
└─12432 /usr/sbin/mysqld
Sep 28 18:37:28 Server2 systemd[1]: Starting MariaDB 10.1.48 database server...
Sep 28 18:37:28 Server2 mysqld[12432]: 2023-09-28 18:37:28 139748019031168 [Not
Sep 28 18:37:28 Server2 /etc/mysql/debian-start[12461]: Upgrading MySQL tables
Sep 28 18:37:28 Server2 systemd[1]: Started MariaDB 10.1.48 database server.
Sep 28 18:37:28 Server2 /etc/mysql/debian-start[12465]: /usr/bin/mysql_upgrade:
Sep 28 18:37:28 Server2 /etc/mysql/debian-start[12465]: Looking for 'mysql' as:
Sep 28 18:37:28 Server2 /etc/mysql/debian-start[12465]: Looking for 'mysqlcheck
Sep 28 18:37:28 Server2 /etc/mysql/debian-start[12465]: This installation of My
Sep 28 18:37:28 Server2 /etc/mysql/debian-start[12474]: Checking for insecure r
ESCOC
Right Ctrl


Server2 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Thu 18:50
rio@Server2: ~
File Edit View Search Terminal Help
abase server
mariadb.service; enabled; vendor preset: enabled)
023-09-28 18:37:28 PST; 2min 44s ago
brary/systemd/
-c systemctl unset-environment _WSREP_START_POSITION (code=exited, status=0/SUC
ql/debian-start (code=exited, status=0/SUCCESS)
c [! -e /usr/bin/galera_recovery] && VAR= || VAR=`cd /usr/bin/..; /usr/bin/
c systemctl unset-environment _WSREP_START_POSITION (code=exited, status=0/SUCC
install -m 755 -o mysql -g root -d /var/run/mysqld (code=exited, status=0/SUCCE
w..."
e
rting MariaDB 10.1.48 database server...
2023-09-28 18:37:28 139748019031168 [Note] /usr/sbin/mysqld (mysqld 10.1.48-Mar
an-start[12461]: Upgrading MySQL tables if necessary.
rted MariaDB 10.1.48 database server.
an-start[12465]: /usr/bin/mysql_upgrade: the '--basedir' option is always ignor
an-start[12465]: Looking for 'mysql' as: /usr/bin/mysql
an-start[12465]: Looking for 'mysqlcheck' as: /usr/bin/mysqlcheck
an-start[12465]: This installation of MySQL is already upgraded to 10.1.48-Mari
an-start[12474]: Checking for insecure root accounts.
~
~
ESCOC
Right Ctrl



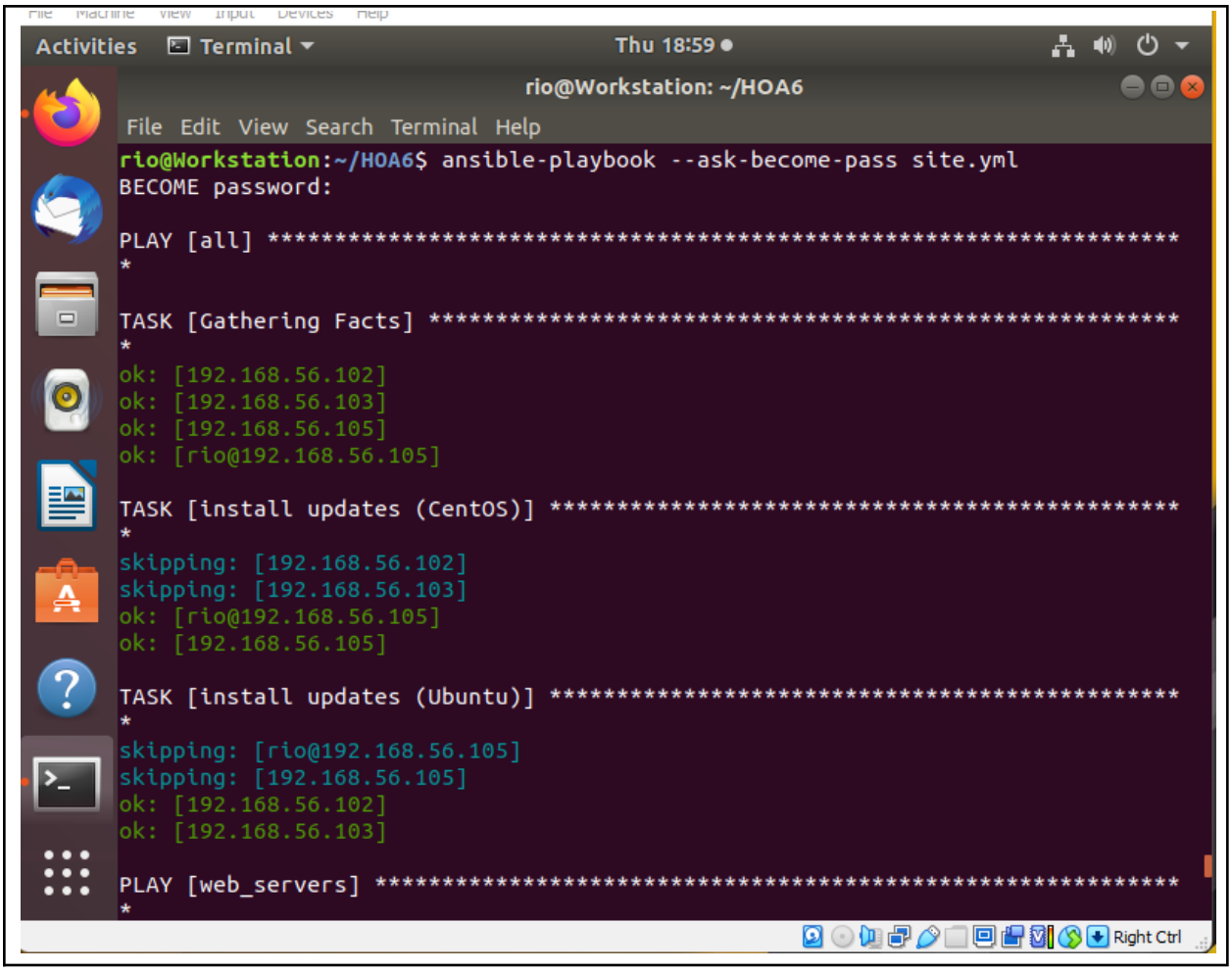
Describe the output.

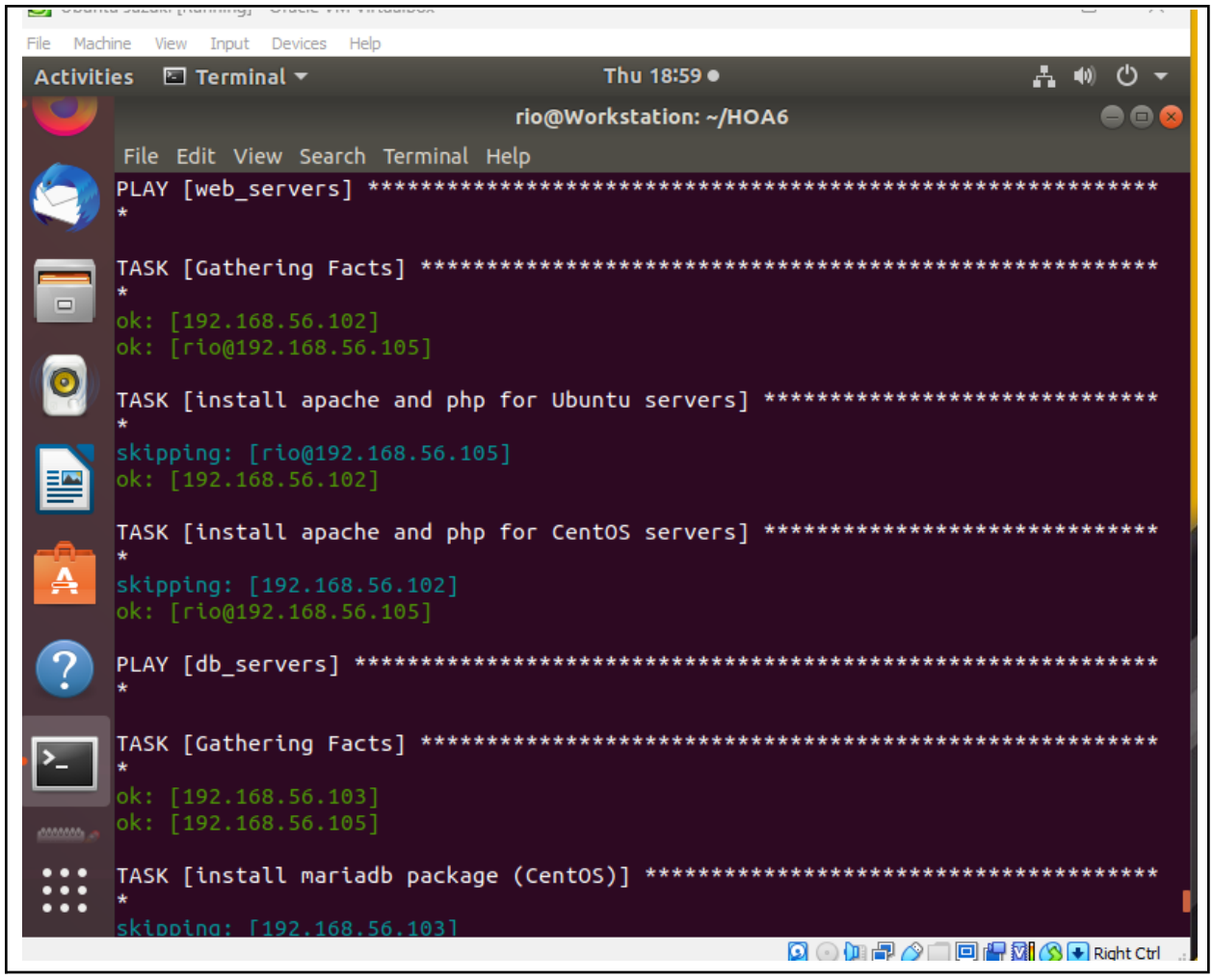
6. Edit the *site.yml* again. This time we will append the code to configure installation on the *file_servers* group. We can add the following on our file.

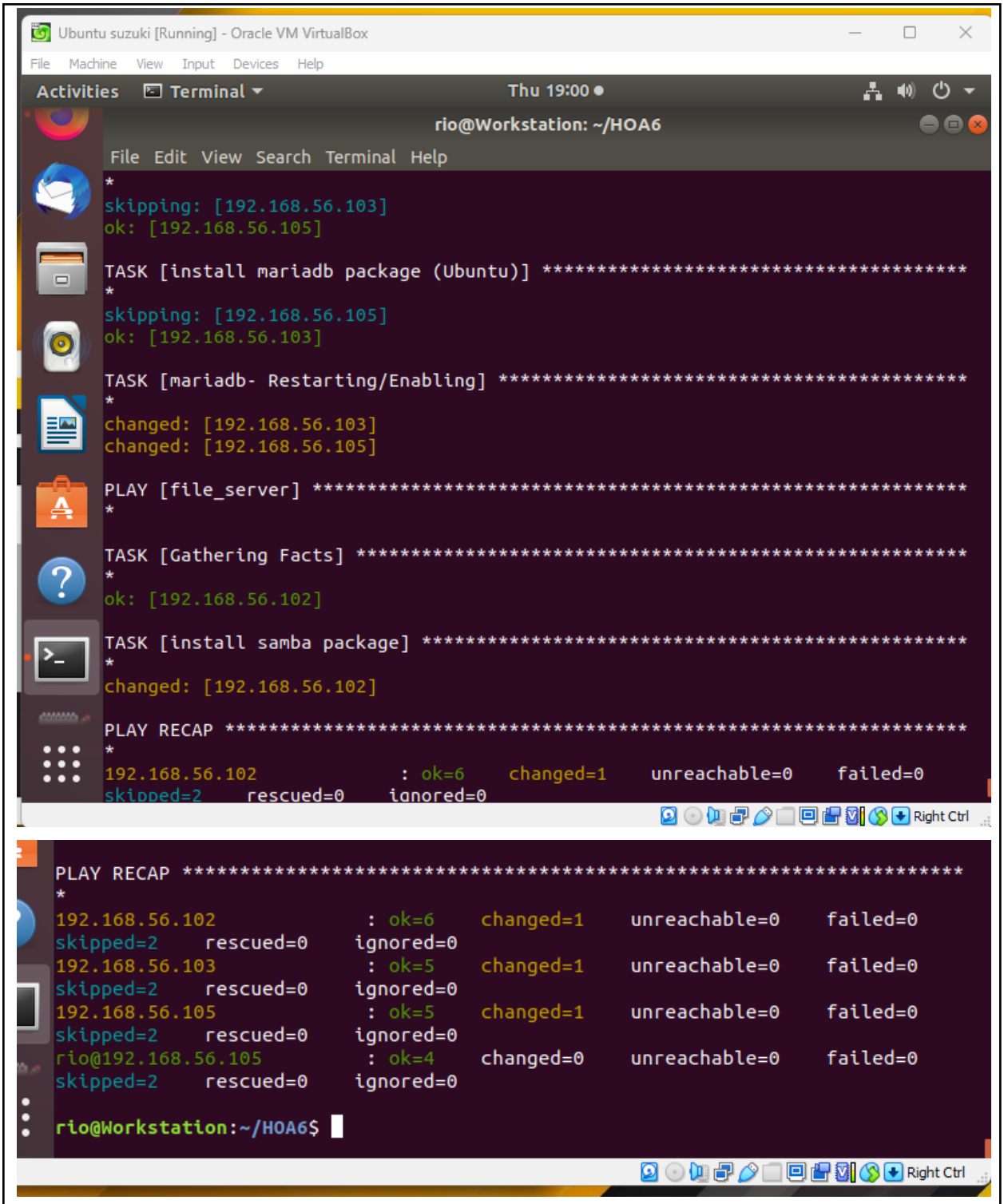
```
- hosts: file_servers
  become: true
  tasks:

  - name: install samba package
    package:
      name: samba
      state: latest
```

Make sure to save the file and exit.

Run the *site.yml* file and describe the result.


Activities
Terminal
Thu 18:59
rio@Workstation: ~/HOA6
File Edit View Search Terminal Help
rio@Workstation:~/HOA6$ ansible-playbook --ask-become-pass site.yml
BECOME password:
PLAY [all] ***
*
TASK [Gathering Facts] ***
*
ok: [192.168.56.102]
ok: [192.168.56.103]
ok: [192.168.56.105]
ok: [rio@192.168.56.105]
TASK [install updates (CentOS)] **
*
skipping: [192.168.56.102]
skipping: [192.168.56.103]
ok: [rio@192.168.56.105]
ok: [192.168.56.105]
TASK [install updates (Ubuntu)] **
*
skipping: [rio@192.168.56.105]
skipping: [192.168.56.105]
ok: [192.168.56.102]
ok: [192.168.56.103]
PLAY [web_servers] ***
*
Right Ctrl


File Machine View Input Devices Help
Activities
Terminal
Thu 18:59
rio@Workstation: ~/HOA6
File Edit View Search Terminal Help
PLAY [web_servers] **
*
TASK [Gathering Facts] **
*
ok: [192.168.56.102]
ok: [rio@192.168.56.105]
TASK [install apache and php for Ubuntu servers] **********************************
*
skipping: [rio@192.168.56.105]
ok: [192.168.56.102]
TASK [install apache and php for CentOS servers] **********************************
*
skipping: [192.168.56.102]
ok: [rio@192.168.56.105]
PLAY [db_servers] ***
*
TASK [Gathering Facts] **
*
ok: [192.168.56.103]
ok: [192.168.56.105]
TASK [install mariadb package (CentOS)] ***
*
skipping: [192.168.56.103]
Right Ctrl

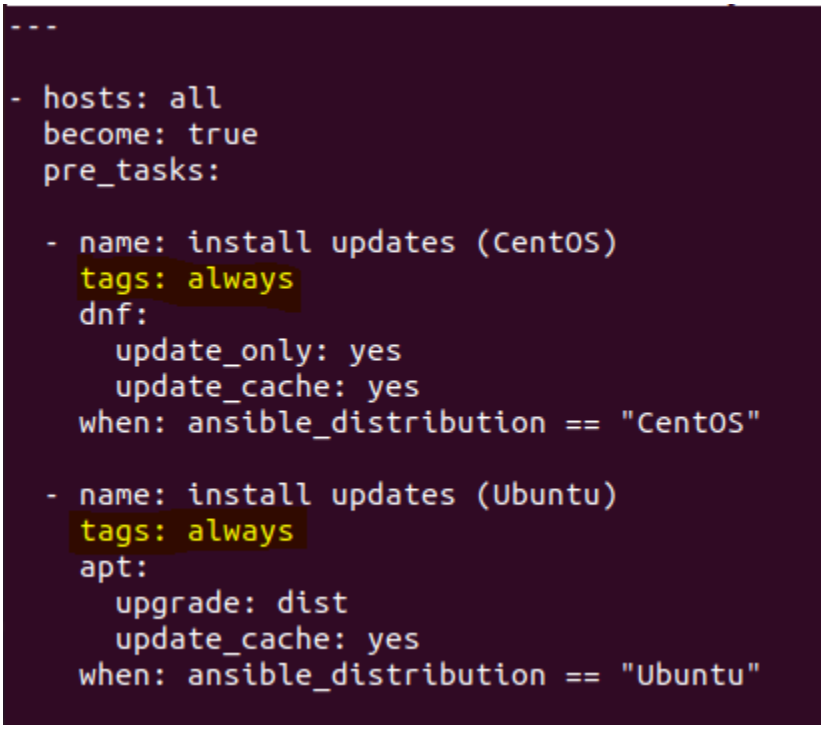


The testing of the *file_servers* is beyond the scope of this activity, and as well as our topics and objectives. However, in this activity we were able to show that we can target hosts or servers using grouping in ansible playbooks.

Task 2: Using Tags in running playbooks

In this task, our goal is to add metadata to our plays so that we can only run the plays that we want to run, and not all the plays in our playbook.

1. Edit the *site.yml* file. Add tags to the playbook. After the name, we can place the tags: *name_of_tag*. This is an arbitrary command, which means you can use any name for a tag.

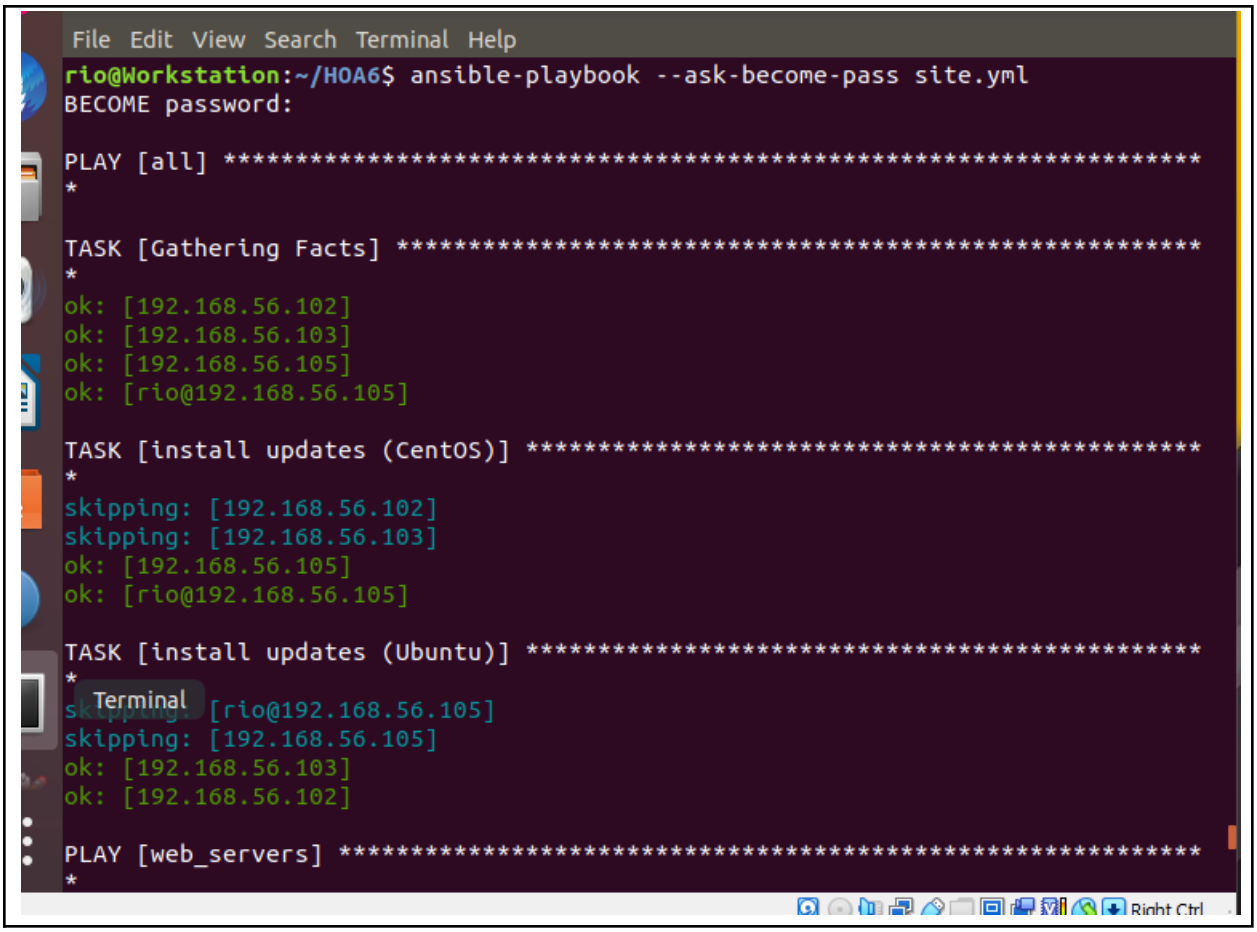


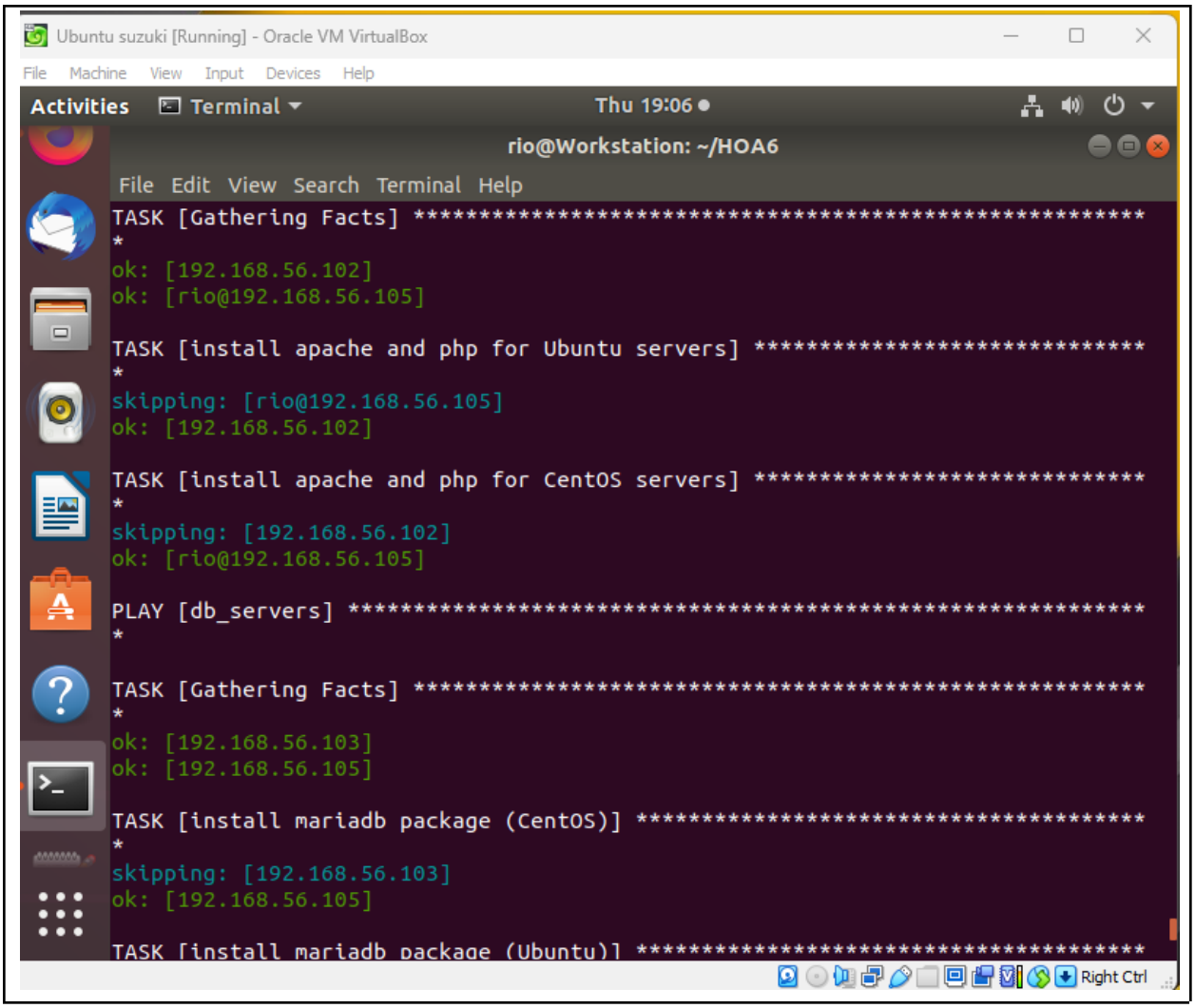
```
---

- hosts: all
  become: true
  pre_tasks:

  - name: install updates (CentOS)
    tags: always
    dnf:
      update_only: yes
      update_cache: yes
    when: ansible_distribution == "CentOS"

  - name: install updates (Ubuntu)
    tags: always
    apt:
      upgrade: dist
      update_cache: yes
    when: ansible_distribution == "Ubuntu"
```

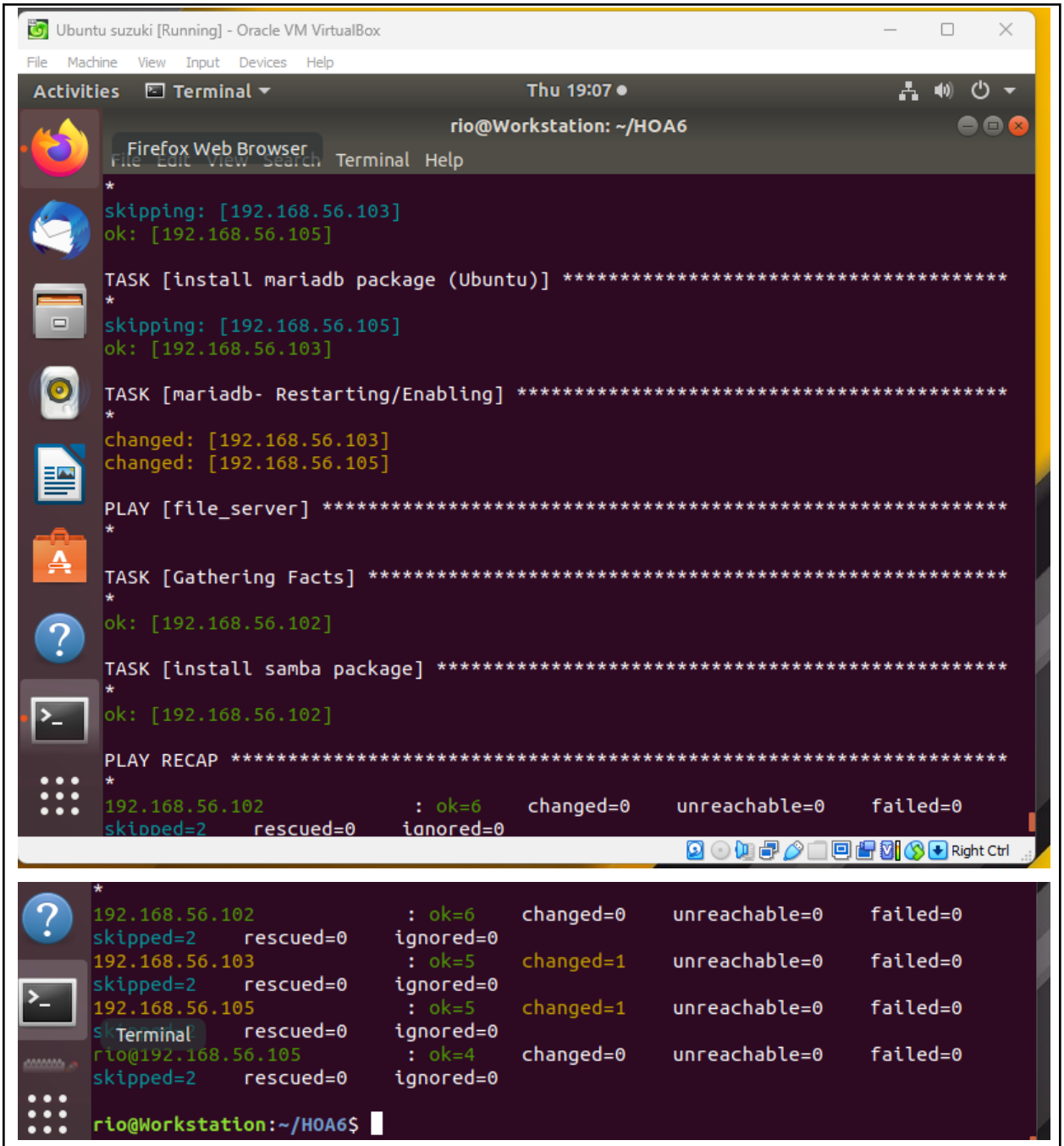
```
- hosts: web_servers
  become: true
  tasks:

  - name: install apache and php for Ubuntu servers
    tags: apache,apache2,ubuntu
    apt:
      name:
        - apache2
        - libapache2-mod-php
      state: latest
    when: ansible_distribution == "Ubuntu"

  - name: install apache and php for CentOS servers
    tags: apache,centos,httpd
    dnf:
      name:
        - httpd
        - php
      state: latest
    when: ansible_distribution == "CentOS"
```

```
- hosts: db_servers
  become: true
  tasks:

  - name: install mariadb package (CentOS)
    tags: centos, db,mariadb
    dnf:
      name: mariadb-server
      state: latest
    when: ansible_distribution == "CentOS"

  - name: "Mariadb- Restarting/Enabling"
    service:
      name: mariadb
      state: restarted
      enabled: true

  - name: install mariadb packege (Ubuntu)
    tags: db, mariadb,ubuntu
    apt:
      name: mariadb-server
      state: latest
    when: ansible_distribution == "Ubuntu"

- hosts: file_servers
  become: true
  tasks:

  - name: install samba package
    tags: samba
    package:
      name: samba
      state: latest
```

Make sure to save the file and exit.
Run the *site.yml* file and describe the result.


File Edit View Search Terminal Help
rio@Workstation:~/HOA6$ ansible-playbook --ask-become-pass site.yml
BECOME password:
PLAY [all] ***
*
TASK [Gathering Facts] ***
*
ok: [192.168.56.102]
ok: [192.168.56.103]
ok: [192.168.56.105]
ok: [rio@192.168.56.105]
TASK [install updates (CentOS)] **
*
skipping: [192.168.56.102]
skipping: [192.168.56.103]
ok: [192.168.56.105]
ok: [rio@192.168.56.105]
TASK [install updates (Ubuntu)] **
*
Terminal
skipping: [rio@192.168.56.105]
skipping: [192.168.56.105]
ok: [192.168.56.103]
ok: [192.168.56.102]
PLAY [web_servers] ***
*
Right Ctrl


Ubuntu suzuki [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Terminal
Thu 19:06
rio@Workstation: ~/HOA6
File Edit View Search Terminal Help
TASK [Gathering Facts] *********
*
ok: [192.168.56.102]
ok: [rio@192.168.56.105]
TASK [install apache and php for Ubuntu servers] *********
*
skipping: [rio@192.168.56.105]
ok: [192.168.56.102]
TASK [install apache and php for CentOS servers] *********
*
skipping: [192.168.56.102]
ok: [rio@192.168.56.105]
PLAY [db_servers] *********
*
TASK [Gathering Facts] *********
*
ok: [192.168.56.103]
ok: [192.168.56.105]
TASK [install mariadb package (CentOS)] *********
*
skipping: [192.168.56.103]
ok: [192.168.56.105]
TASK [install mariadb package (Ubuntu)] *********
Right Ctrl



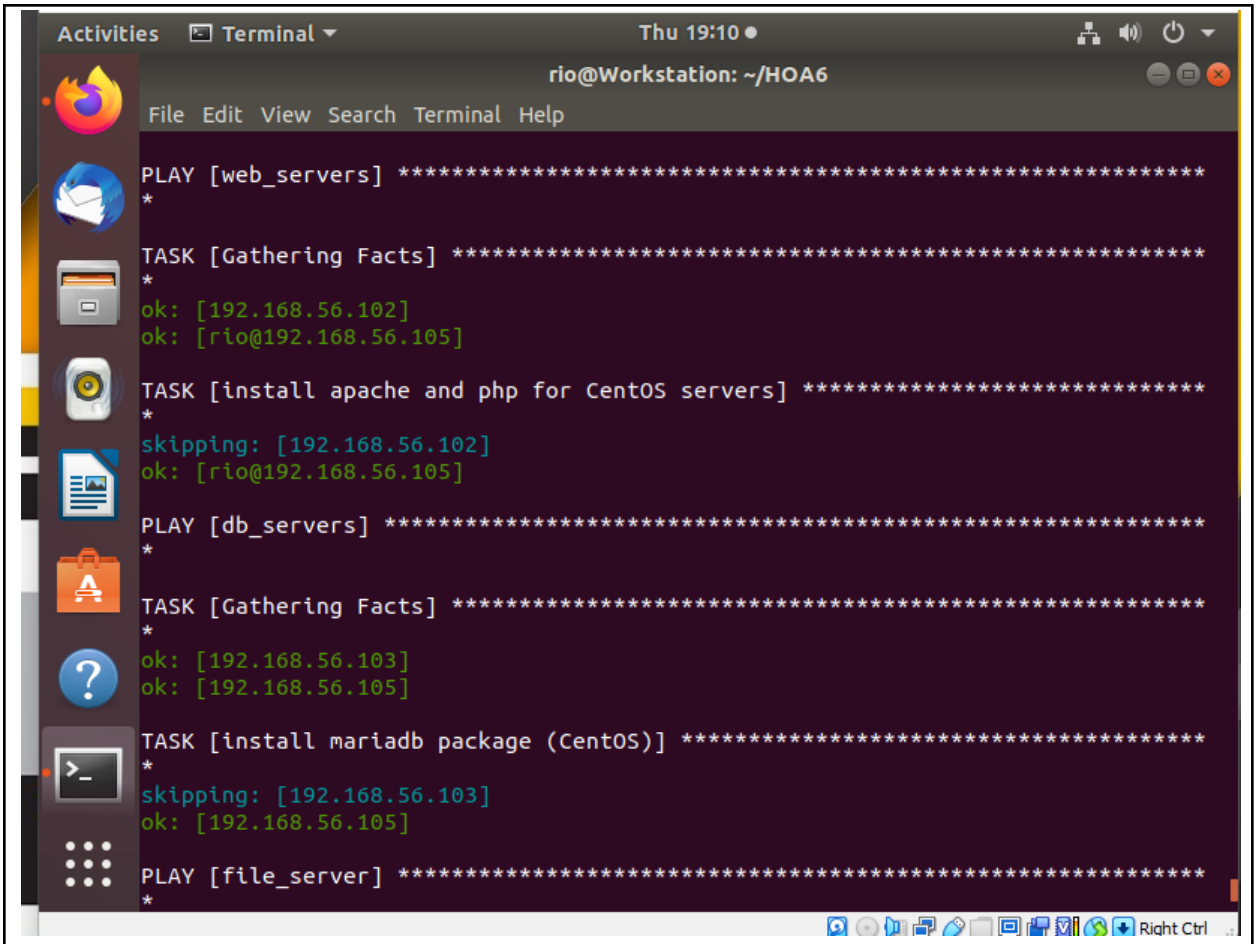
2. On the local machine, try to issue the following commands and describe each result:
   2.1 *ansible-playbook --list-tags site.yml*

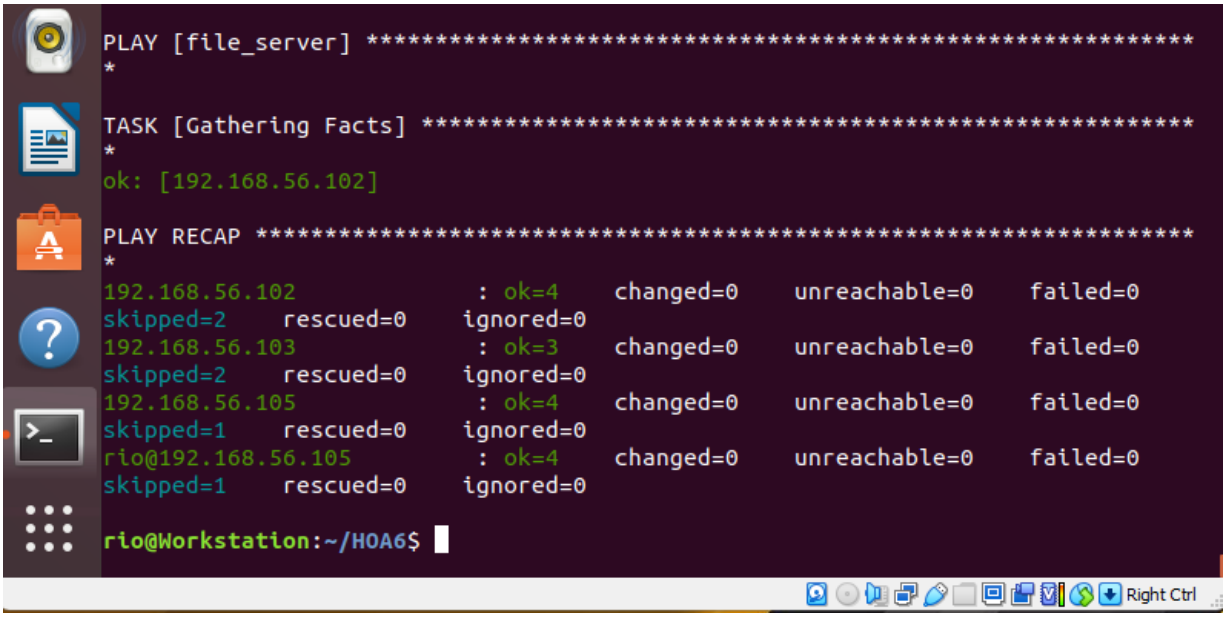
```
rio@Workstation:~/HOA6$ ansible-playbook --list-tags site.yml

playbook: site.yml

  play #1 (all): all    TAGS: []
      TASK TAGS: [always]

  play #2 (web_servers): web_servers    TAGS: []
      TASK TAGS: [apache, apache2, centos, httpd, ubuntu]

  play #3 (db_servers): db_servers      TAGS: []
      TASK TAGS: [centos, db, mariadb, ubuntu]

  play #4 (file_server): file_server    TAGS: []
      TASK TAGS: [samba]
rio@Workstation:~/HOA6$
```

2.2 *ansible-playbook --tags centos --ask-become-pass site.yml*

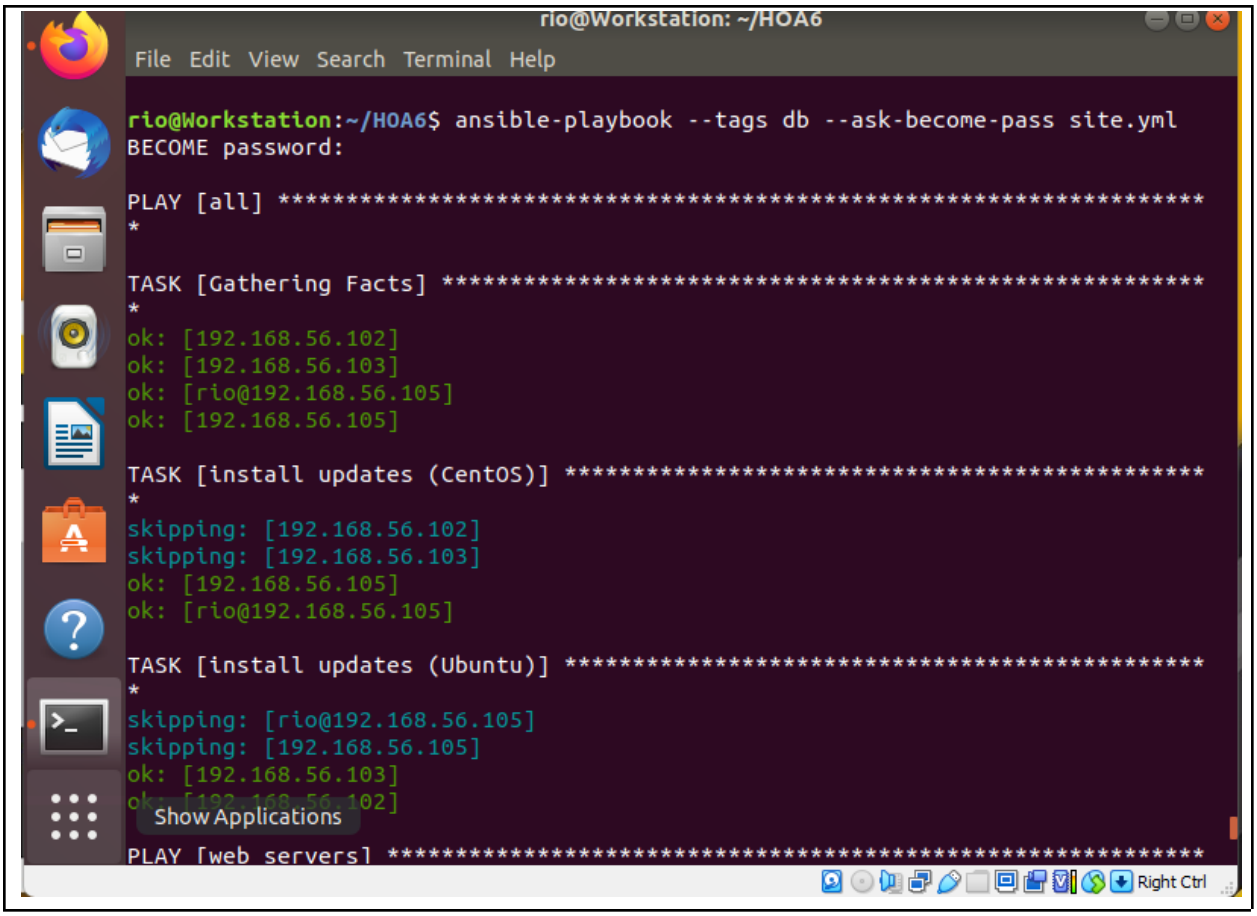
```
File  Edit  View  Search  Terminal  Help
rio@Workstation:~/HOA6$ ansible-playbook --tags centos --ask-become-pass site.y
ml
BECOME password:

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

TASK [Gathering Facts] *********************************************************
*
ok: [192.168.56.102]
ok: [192.168.56.103]
ok: [192.168.56.105]
ok: [rio@192.168.56.105]

TASK [install updates (CentOS)] ************************************************
*
skipping: [192.168.56.102]
skipping: [192.168.56.103]
ok: [rio@192.168.56.105]
ok: [192.168.56.105]

TASK [install updates (Ubuntu)] ************************************************
*
skipping: [rio@192.168.56.105]
skipping: [192.168.56.105]
ok: [192.168.56.102]
ok: [192.168.56.103]

PLAY [web_servers] *************************************************************
```

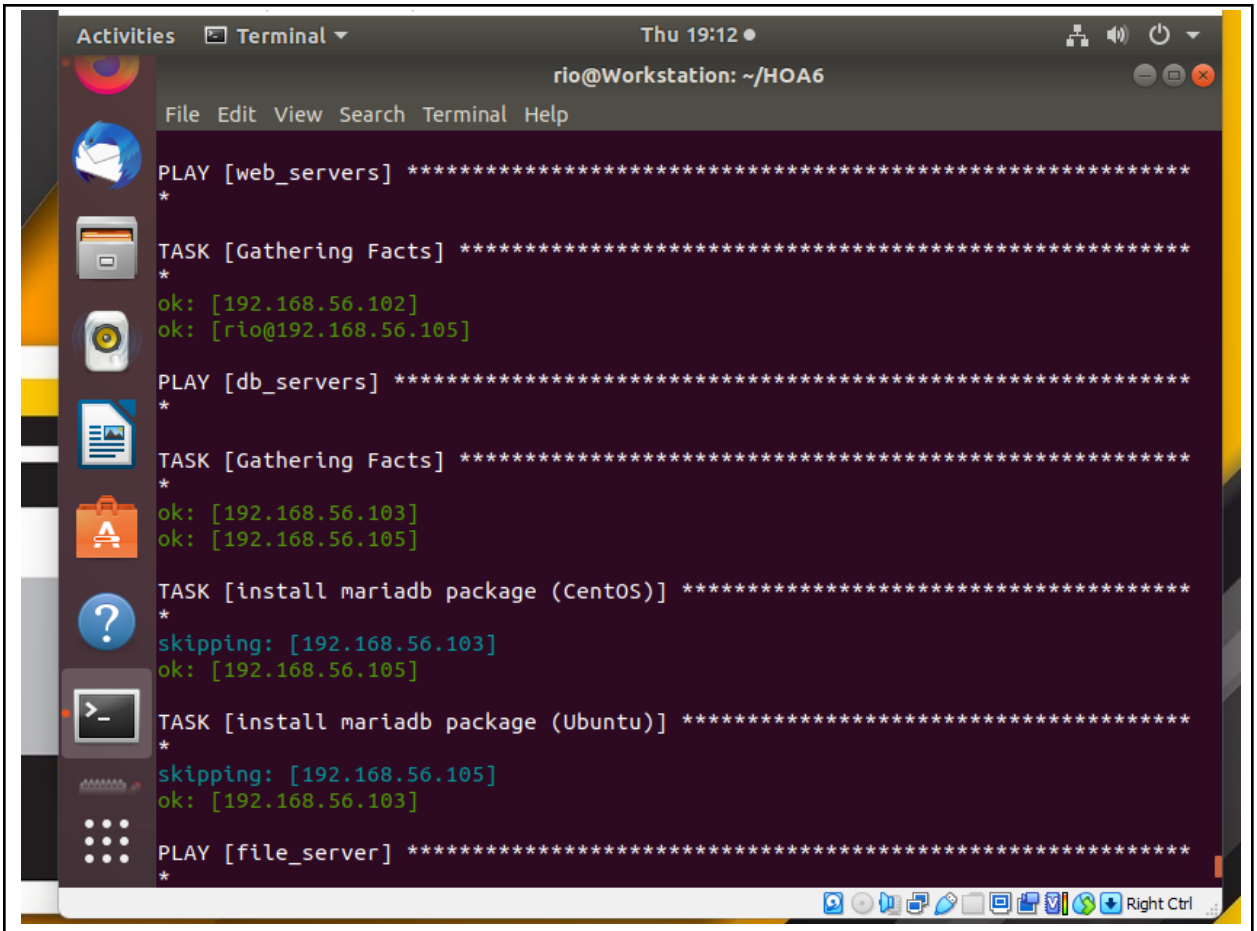
Right Ctrl

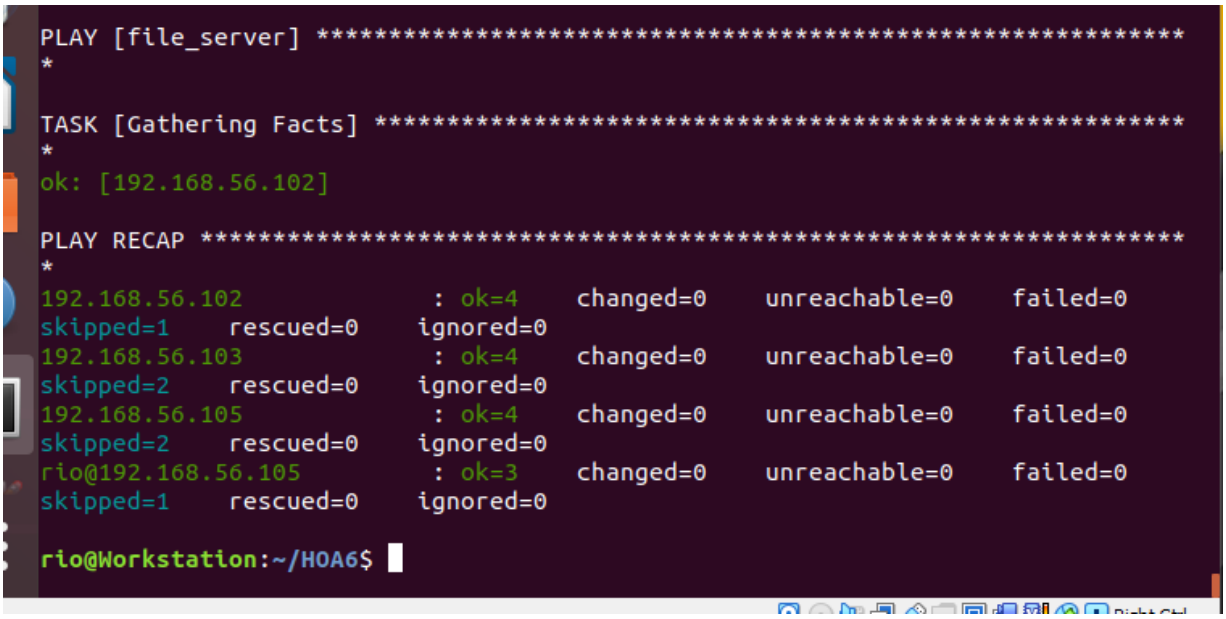




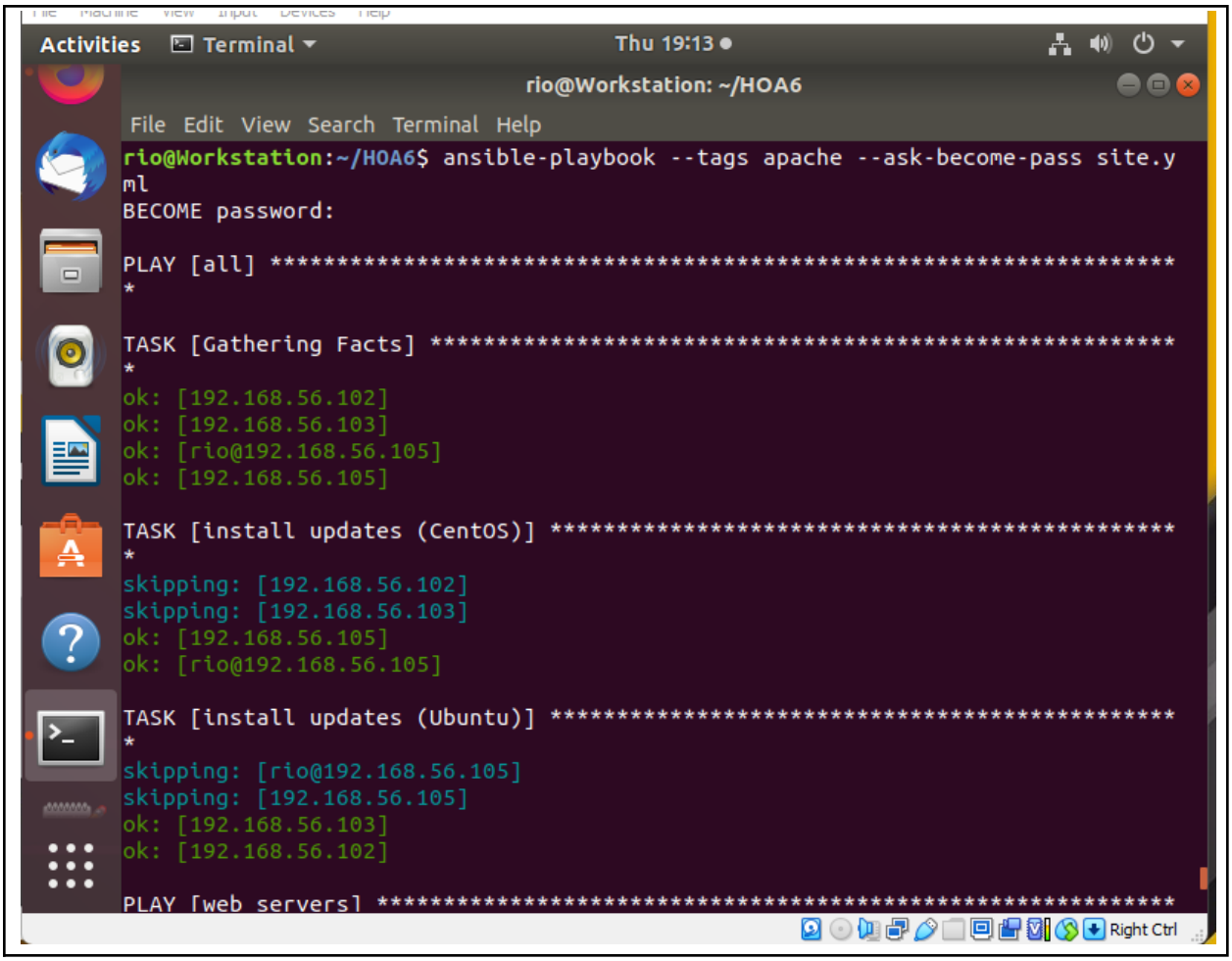
2.3 *ansible-playbook --tags db --ask-become-pass site.yml*

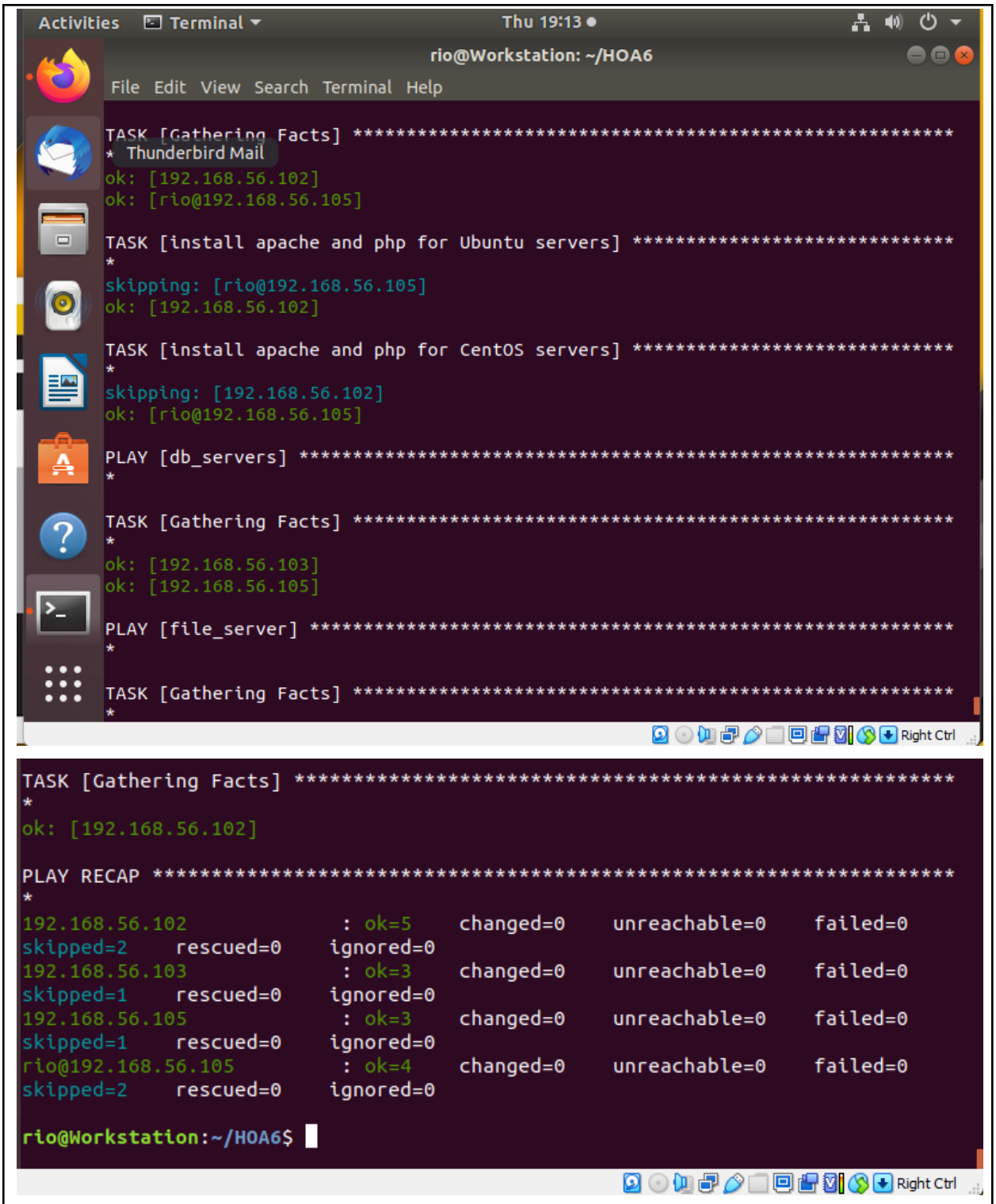

rio@Workstation: ~/HOA6
File Edit View Search Terminal Help
rio@Workstation:~/HOA6$ ansible-playbook --tags db --ask-become-pass site.yml
BECOME password:
PLAY [all] ***
*
TASK [Gathering Facts] ***
*
ok: [192.168.56.102]
ok: [192.168.56.103]
ok: [rio@192.168.56.105]
ok: [192.168.56.105]
TASK [install updates (CentOS)] **
*
skipping: [192.168.56.102]
skipping: [192.168.56.103]
ok: [192.168.56.105]
ok: [rio@192.168.56.105]
TASK [install updates (Ubuntu)] **
*
skipping: [rio@192.168.56.105]
skipping: [192.168.56.105]
ok: [192.168.56.103]
Show Applications
PLAY [web servers] ***
Right Ctrl



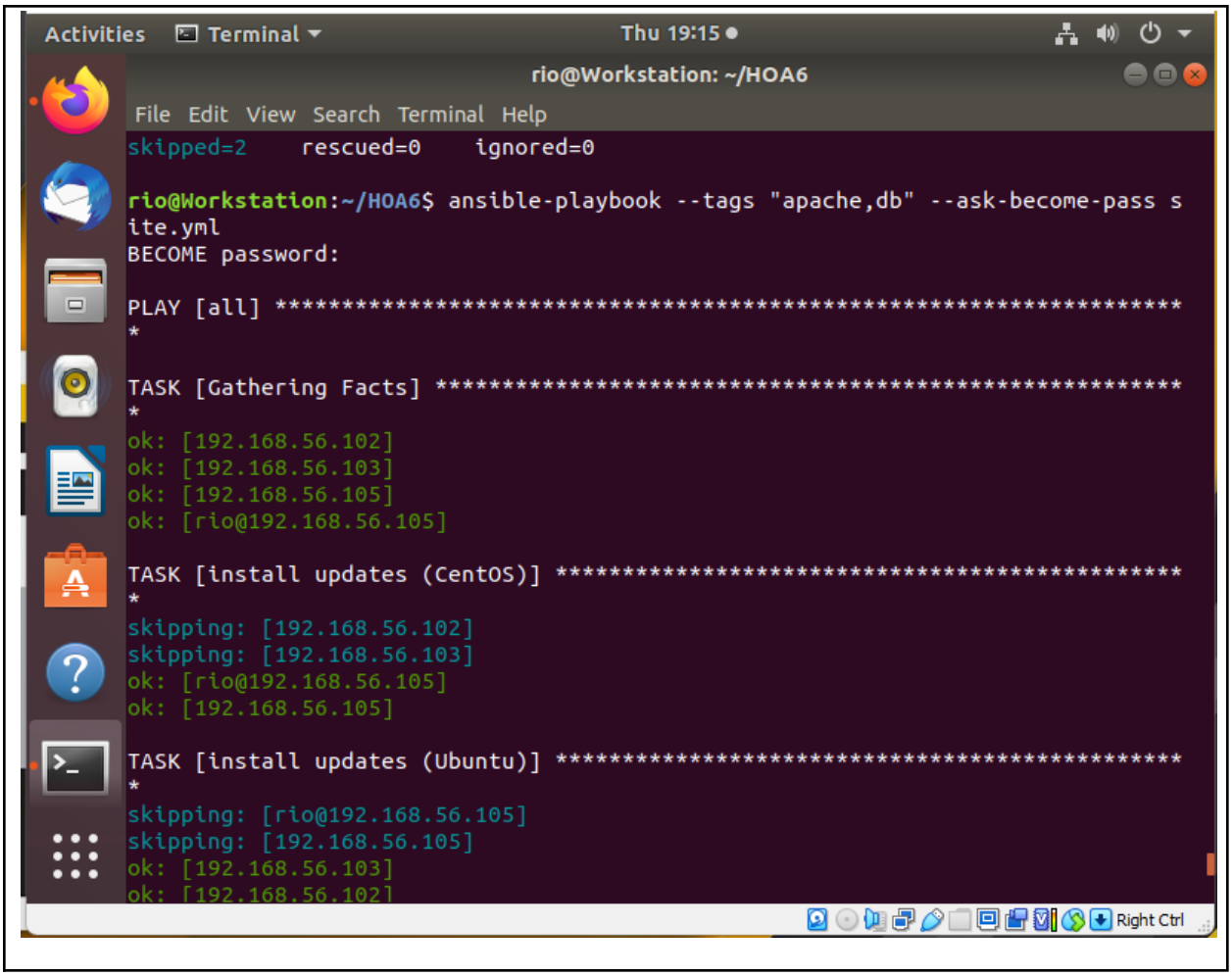


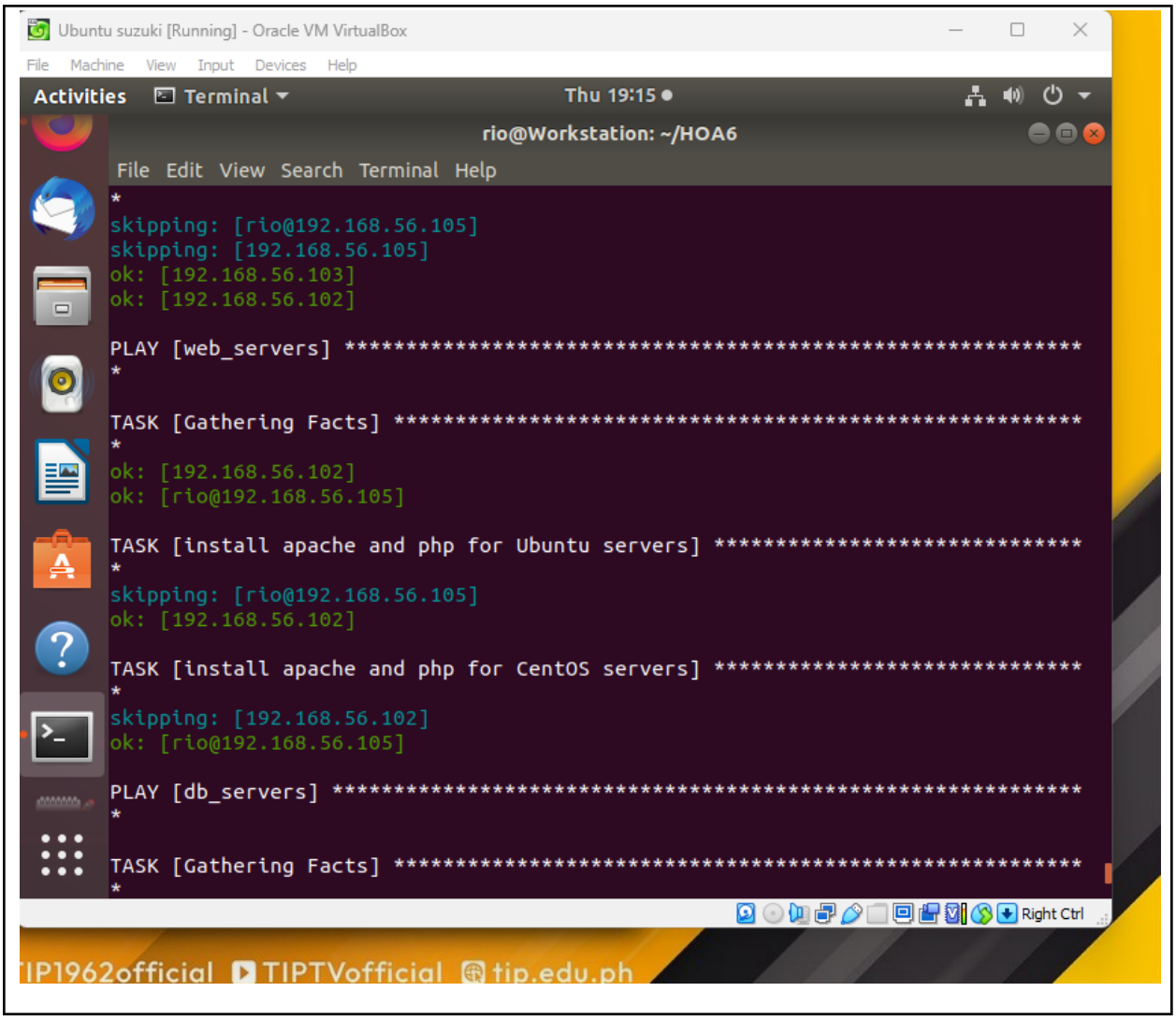
2.4 *ansible-playbook --tags apache --ask-become-pass site.yml*

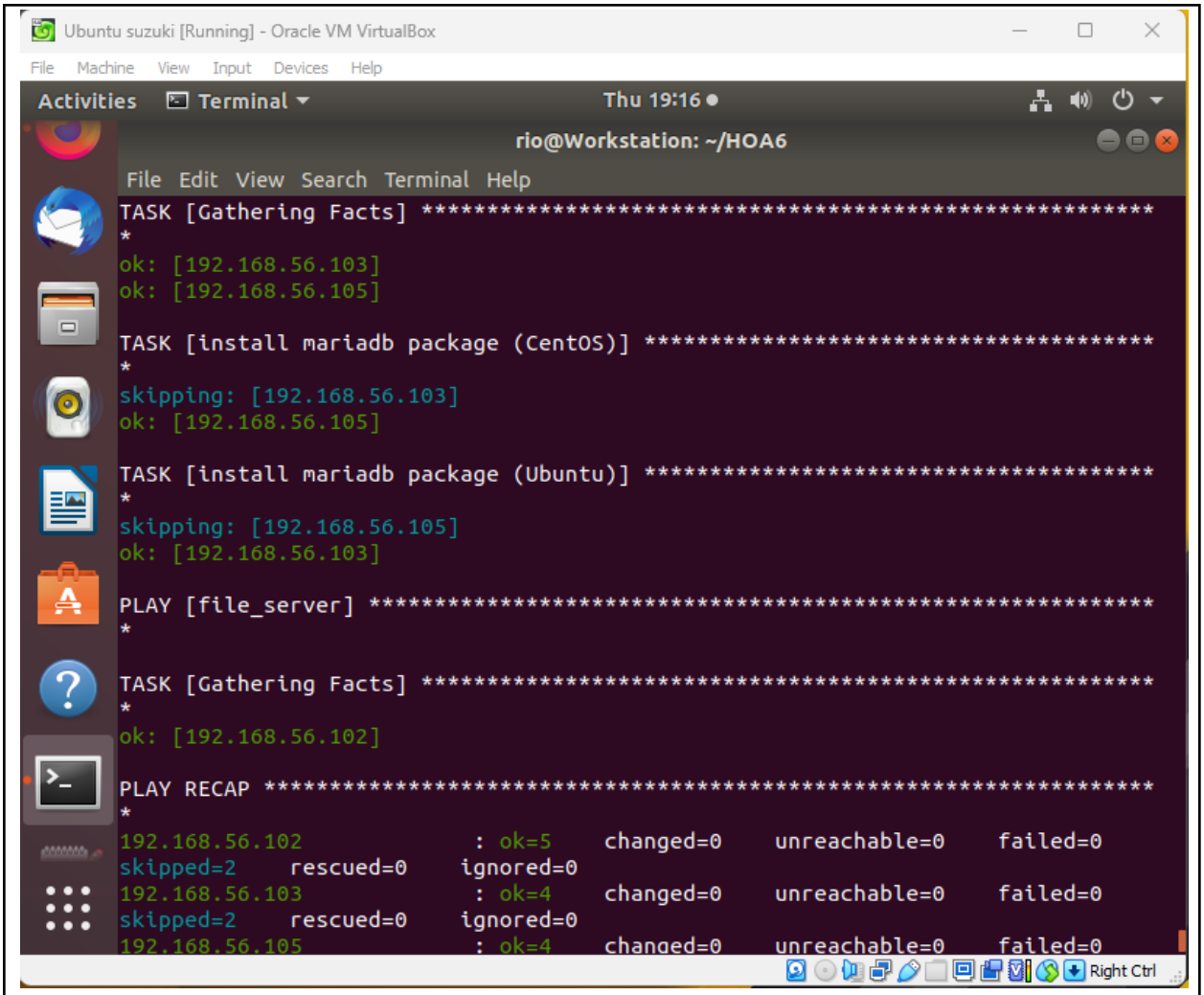

Activities
Terminal
Thu 19:13
rio@Workstation: ~/HOA6
File Edit View Search Terminal Help
rio@Workstation:~/HOA6$ ansible-playbook --tags apache --ask-become-pass site.yml
BECOME password:
PLAY [all] ***
TASK [Gathering Facts] ***
ok: [192.168.56.102]
ok: [192.168.56.103]
ok: [rio@192.168.56.105]
ok: [192.168.56.105]
TASK [install updates (CentOS)] **
skipping: [192.168.56.102]
skipping: [192.168.56.103]
ok: [192.168.56.105]
ok: [rio@192.168.56.105]
TASK [install updates (Ubuntu)] **
skipping: [rio@192.168.56.105]
skipping: [192.168.56.105]
ok: [192.168.56.103]
ok: [192.168.56.102]
PLAY [web servers] ***
Right Ctrl



2.5 *ansible-playbook --tags “apache,db” --ask-become-pass site.yml*


Activities
Terminal
Thu 19:15
rio@Workstation: ~/HOA6
File Edit View Search Terminal Help
skipped=2 rescued=0 ignored=0
rio@Workstation:~/HOA6$ ansible-playbook --tags "apache,db" --ask-become-pass site.yml
BECOME password:
PLAY [all] ***********
TASK [Gathering Facts] ***********
ok: [192.168.56.102]
ok: [192.168.56.103]
ok: [192.168.56.105]
ok: [rio@192.168.56.105]
TASK [install updates (CentOS)] ***********
skipping: [192.168.56.102]
skipping: [192.168.56.103]
ok: [rio@192.168.56.105]
ok: [192.168.56.105]
TASK [install updates (Ubuntu)] ***********
skipping: [rio@192.168.56.105]
skipping: [192.168.56.105]
ok: [192.168.56.103]
Right Ctrl


Ubuntu suzuki [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Terminal
Thu 19:15
rio@Workstation: ~/HOA6
File Edit View Search Terminal Help
*
skipping: [rio@192.168.56.105]
skipping: [192.168.56.105]
ok: [192.168.56.103]
ok: [192.168.56.102]
PLAY [web_servers] **********
*
TASK [Gathering Facts] **********
*
ok: [192.168.56.102]
ok: [rio@192.168.56.105]
TASK [install apache and php for Ubuntu servers] **********
*
skipping: [rio@192.168.56.105]
ok: [192.168.56.102]
TASK [install apache and php for CentOS servers] **********
*
skipping: [192.168.56.102]
ok: [rio@192.168.56.105]
PLAY [db_servers] **********
*
TASK [Gathering Facts] **********
*
Right Ctrl
IP1962official
TIPTVofficial
tip.edu.ph




```
*
192.168.56.102              : ok=5    changed=0    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
192.168.56.103              : ok=4    changed=0    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
192.168.56.105              : ok=4    changed=0    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
rio@192.168.56.105          : ok=4    changed=0    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0

rio@Workstation:~/HOA6$
```

Task 3: Managing Services

1. Edit the file site.yml and add a play that will automatically start the httpd on CentOS server.

```
- name: install apache and php for CentOS servers
  tags: apache,centos,httpd
  dnf:
    name:
      - httpd
      - php
    state: latest
  when: ansible_distribution == "CentOS"

- name: start httpd (CentOS)
  tags: apache, centos,httpd
  service:
    name: httpd
    state: started
  when: ansible_distribution == "CentOS"
```

Figure 3.1.1
Make sure to save the file and exit.

You would also notice from our previous activity that we already created a module that runs a service.

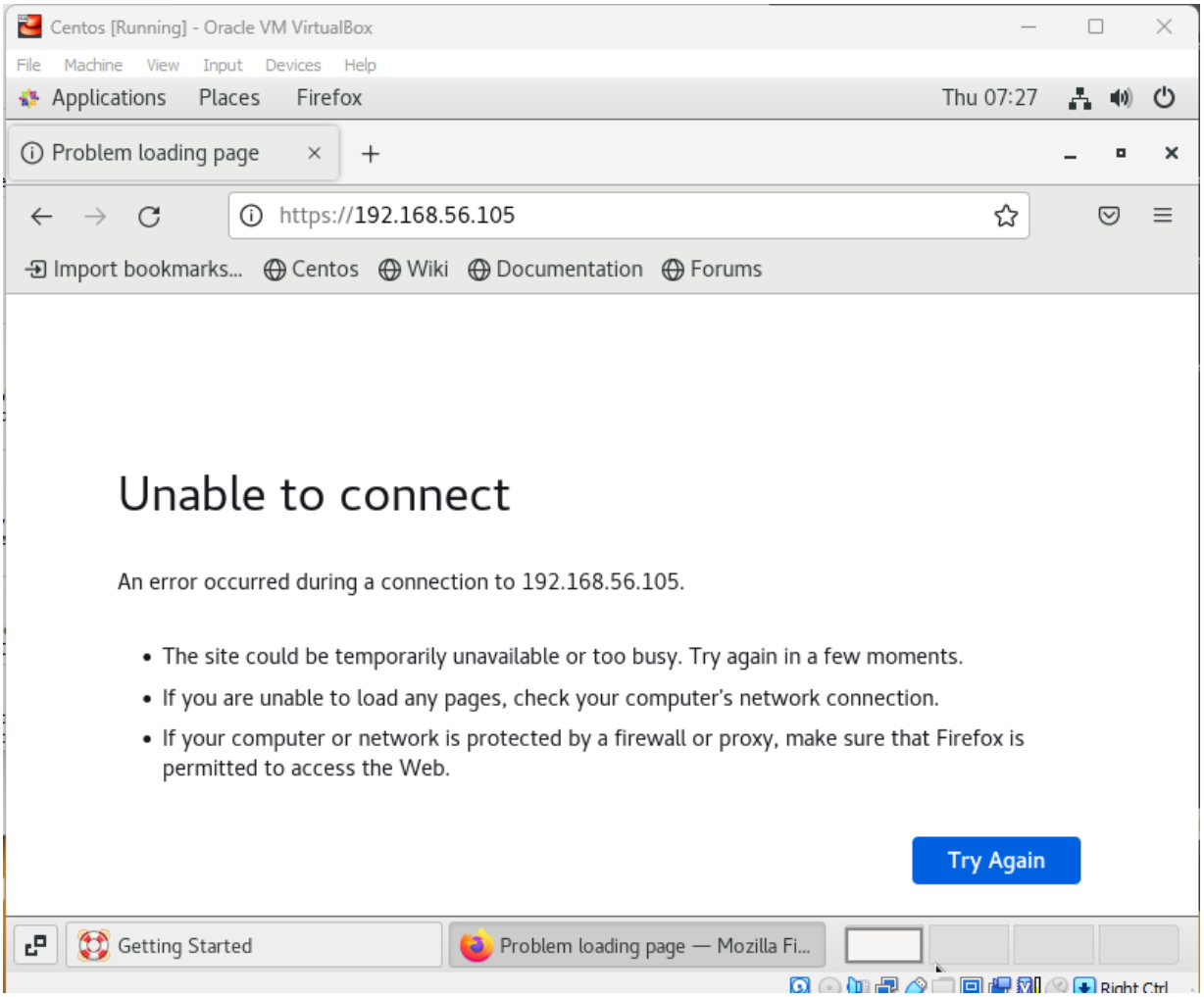
```
- hosts: db_servers
  become: true
  tasks:

  - name: install mariadb package (CentOS)
    tags: centos, db,mariadb
    dnf:
      name: mariadb-server
      state: latest
    when: ansible_distribution == "CentOS"

  - name: "Mariadb- Restarting/Enabling"
    service:
      name: mariadb
      state: restarted
      enabled: true
```

Figure 3.1.2

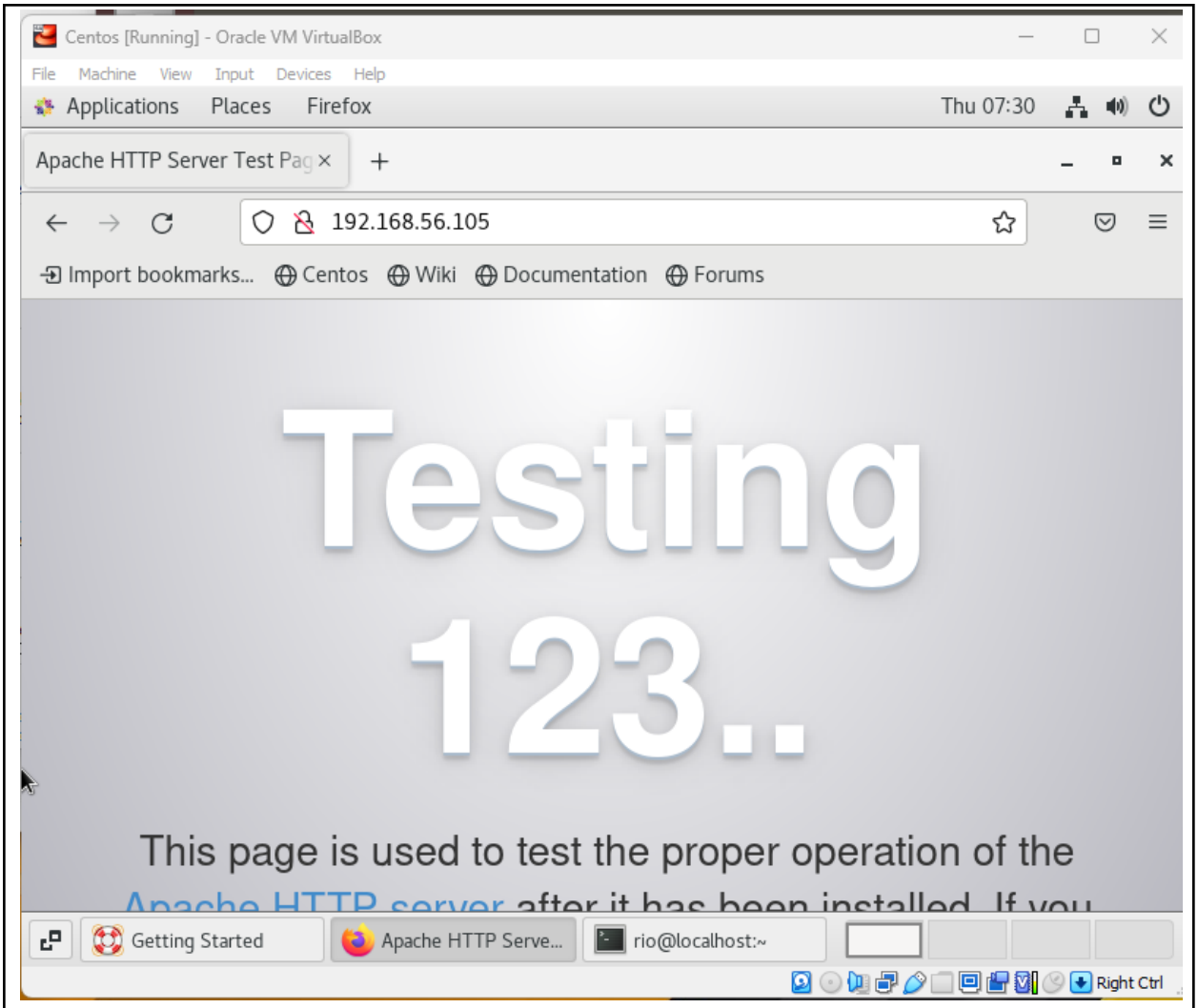
This is because in CentOS, installed packages' services are not run automatically. Thus, we need to create the module to run it automatically.

2. To test it, before you run the saved playbook, go to the CentOS server and stop the currently running httpd using the command *sudo systemctl stop httpd*.

When prompted, enter the sudo password. After that, open the browser and enter the CentOS server's IP address. You should not be getting a display because we stopped the httpd service already.



3. Go to the local machine and this time, run the *site.yml* file. Then after running the file, go again to the CentOS server and enter its IP address on the browser. Describe the result.
   To automatically enable the service every time we run the playbook, use the command *enabled: true* similar to Figure 7.1.2 and save the playbook.



Reflections:

Answer the following:

1. What is the importance of putting our remote servers into groups?
   - For efficient resource management, it is crucial to group remote servers. Simpler IT infrastructure management and increased reliability. By assembling computers with similar features or services into clusters, organizations may maximize resource allocation, carry out load balancing to guarantee fair work distribution, and promote high availability through redundancy.
   -
2. What is the importance of tags in playbooks?
   - Configuration management typically uses playbook tags. Ansible-like tools are essential for carrying out certain tasks or the responsibilities in a playbook. They enable the implementation of a plan to be fine-tuned. playbook, which is particularly helpful in complex settings with several setups or servers.

3. Why do think some services need to be managed automatically in playbooks?

   - Playbooks are crucial for numerous services to manage services automatically. reasons. First and foremost, it contributes to maintaining a reliable and accurate approach by ensuring uniform settings and procedures throughout the infrastructure of an organization. Automation also makes rapid decisions possible. Ability to adapt and respond to changing requirements, allowing services to be established, extended, or terminated as needed without the need for manual involvement. Time is saved, and human error is reduced as a result of mistakes.

Conclusion

To conclude this activity aims for the student to learn how to Individualize hosts, Apply tags in selecting plays to run, and Managing Services from remote servers using playbooks. Using a specific plans and tools to control computer systems helps organizations manage their technology smoothly. It lets them use resources well, fix problems easily, and grow when needed. These plans also make things safer and easier to watch from a distance. Overall, these strategies help organizations stay strong and flexible in the ever-changing tech world.

Criteria

Hands-On Rubirc (2)

Criteria	Ratings					Pts
Completeness This criterion specifies the analysis of the student of the task given.	**5 pts** **Excellent** Components of all tasks are present in the documentation and execution.	**4 pts** **Good** Components of most of the tasks are present in the documentation and execution.	**3 pts** **Ok** Components of half of the tasks are present in documentation and execution.	**2 pts** **Poor** Components some tasks are present in documentation and execution	**1 pts** **Bad** Components of all tasks lacks data in documentation and execution.	5 pts
Design This criterion measures the components and engineering of the Hands-on activity.	**5 pts** **Execellent** Design is robust and acceptable in the industry	**4 pts** **Good** Design is acceptable in the industry but can be improved.	**3 pts** **Ok** Design is a satisfactory level in the industry.	**2 pts** **Poor** Design is poorly architected and engineered needs improvement.	**1 pts** **Bad** Design is badly architectured and engineered needs revisiting and rework.	5 pts
Documentation This criterion measures the context and completeness of artifacts of the activity.	**5 pts** **Excellent** The context of documentation is precise and understandable to readers.	**4 pts** **Good** The context of documentation is acceptable for readers.	**3 pts** **Ok** The documentation is satisfactory, has the main components needed, and grammar is acceptable.	**2 pts** **Poor** The documentation needs grammar checks but the content is complete.	**1 pts** **Bad** Documentation needs revisions from grammar to contexts.	5 pts
						Total Points: 15