

Name: Aaron Martin P. Caro	Date Performed: 13/10/2023
Course/Section: CPE232/CPE31S5	Date Submitted: 14/10/2023
Instructor: Prof. Roman Richard	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.

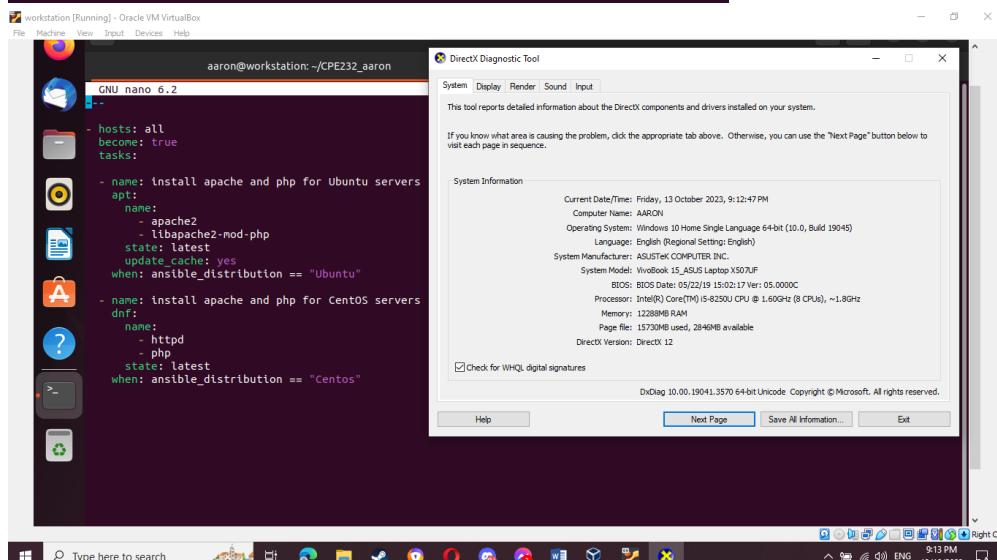
```

---
- 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:

```

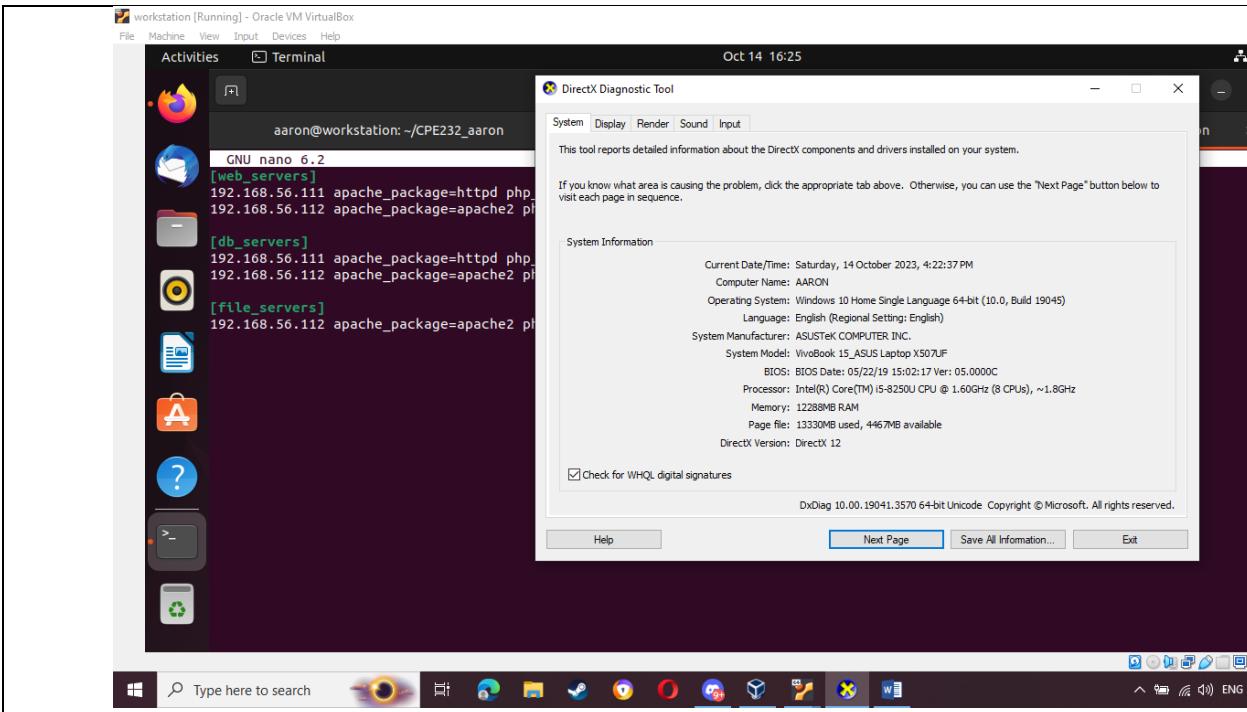
[web_servers]
192.168.56.120
192.168.56.121

[db_servers]
192.168.56.122

[file_servers]
192.168.56.123

```

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"
```

Make sure to save the file and exit.

```

workstation [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
site.yml

- hosts: all
  become: true
  pre_tasks:
    - name: install updates (CentOS)
      dnf:
        update_only: yes
        update_cache: yes
        use_backend: dnf4
      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 server
      apt:
        name:
          - apache2
          - libapache2-mod-php
        state: latest
      when: ansible_distribution == "Ubuntu"
    - name: install apache and php for CentOS servers

```

The screenshot shows a Windows desktop environment. In the foreground, there is a terminal window titled 'workstation [Running] - Oracle VM VirtualBox' containing an Ansible playbook named 'site.yml'. The playbook includes sections for 'hosts: all', 'hosts: web_servers', and tasks for installing updates and apache/php. In the background, a 'DirectX Diagnostic Tool' window is open, providing system information about the host machine.

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.

```

workstation [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
aaron@workstation:~$ cd CPE232_aaron/
aaron@workstation:~/CPE232_aaron$ ansible-playbook --ask-become-pass site.yml
BECOME password:
PLAY [all] ****
TASK [Gathering Facts] ****
ok: [192.168.56.112]
ok: [192.168.56.111]

TASK [install updates (CentOS)] ****
skipping: [192.168.56.112]
ok: [192.168.56.111]

TASK [install updates (Ubuntu)] ****
skipping: [192.168.56.111]
ok: [192.168.56.112]

PLAY [web_servers] ****
TASK [Gathering Facts] ****
ok: [192.168.56.112]
ok: [192.168.56.111]

TASK [install apache and php for Ubuntu servers] *
skipping: [192.168.56.111]
ok: [192.168.56.112]

TASK [install apache and php for CentOS servers] *
skipping: [192.168.56.112]
ok: [192.168.56.111]

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

```

The screenshot shows a Windows desktop environment. In the foreground, there is a terminal window titled 'workstation [Running] - Oracle VM VirtualBox' containing the command to run the Ansible playbook 'site.yml'. The playbook runs through various tasks, including gathering facts and installing packages on both CentOS and Ubuntu servers. In the background, a 'DirectX Diagnostic Tool' window is open, providing system information about the host machine.

The playbook started with the **pre_task** before it went to the main task.

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.

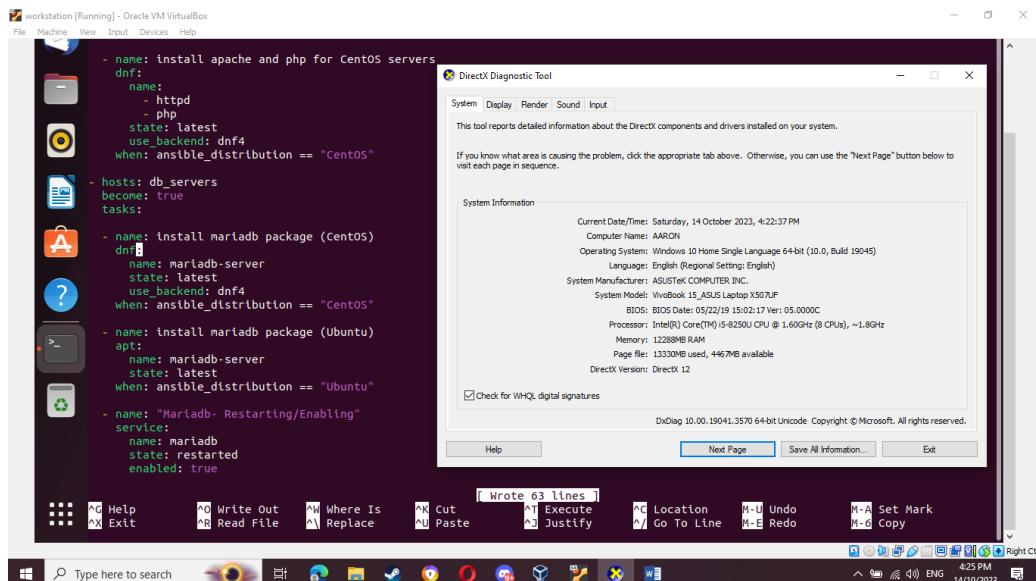
```
- hosts: db_servers
  become: true
  tasks:

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

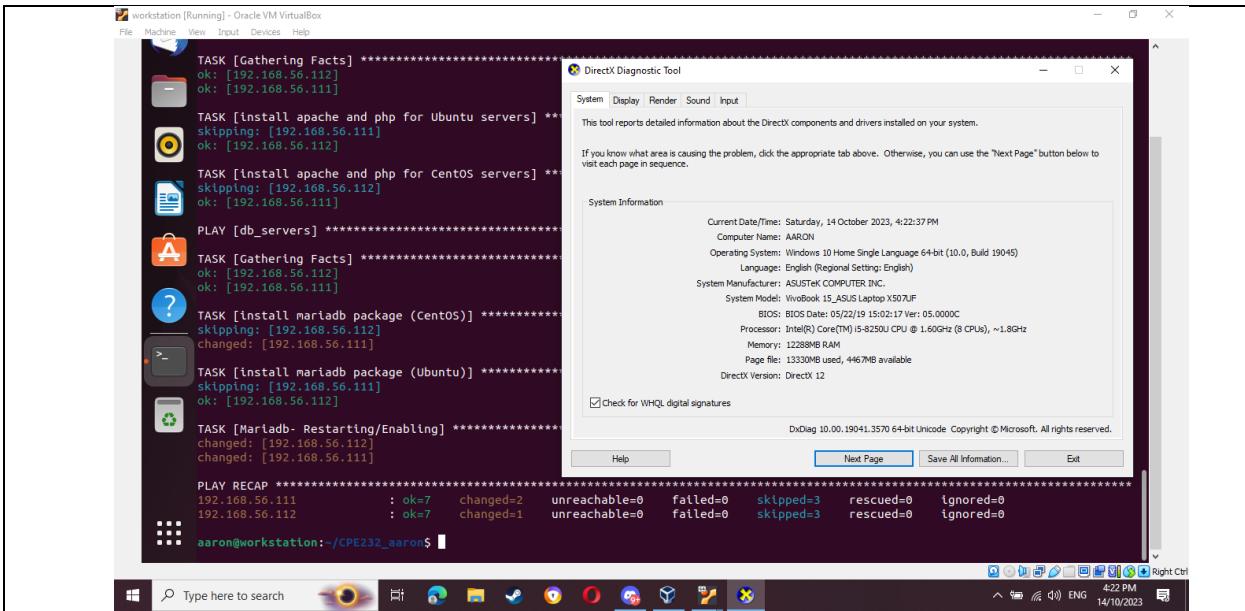
    - name: "Mariadb- Restarting/Enabling"
      service:
        name: mariadb
        state: restarted
        enabled: true

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

Make sure to save the file and exit.



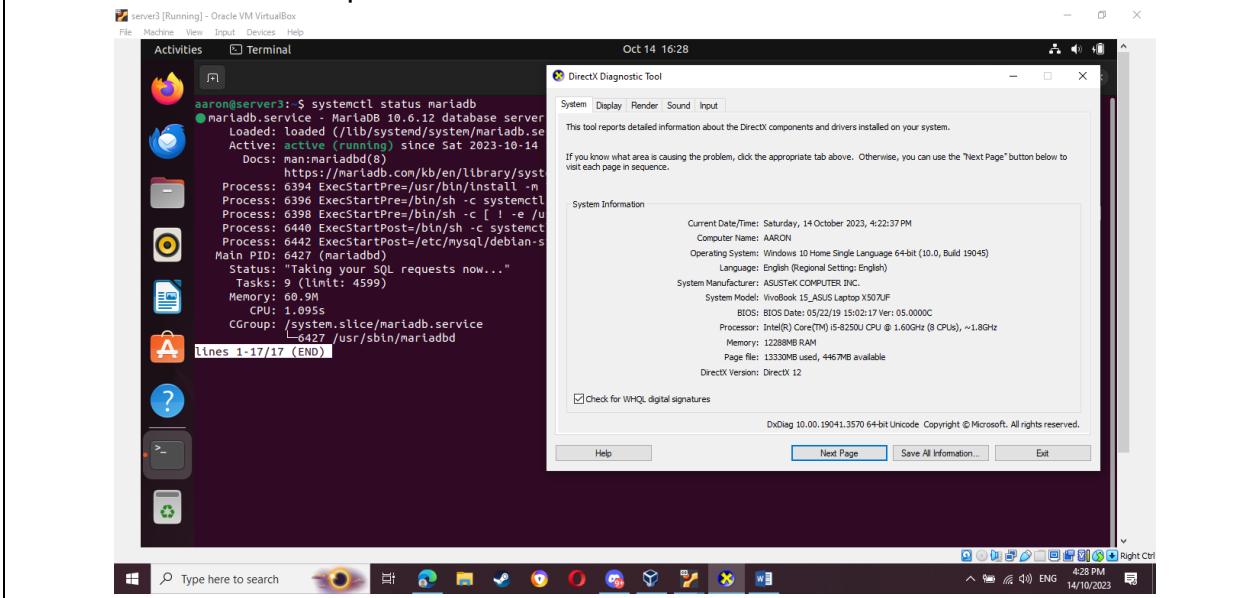
Run the *site.yml* file and describe the result.

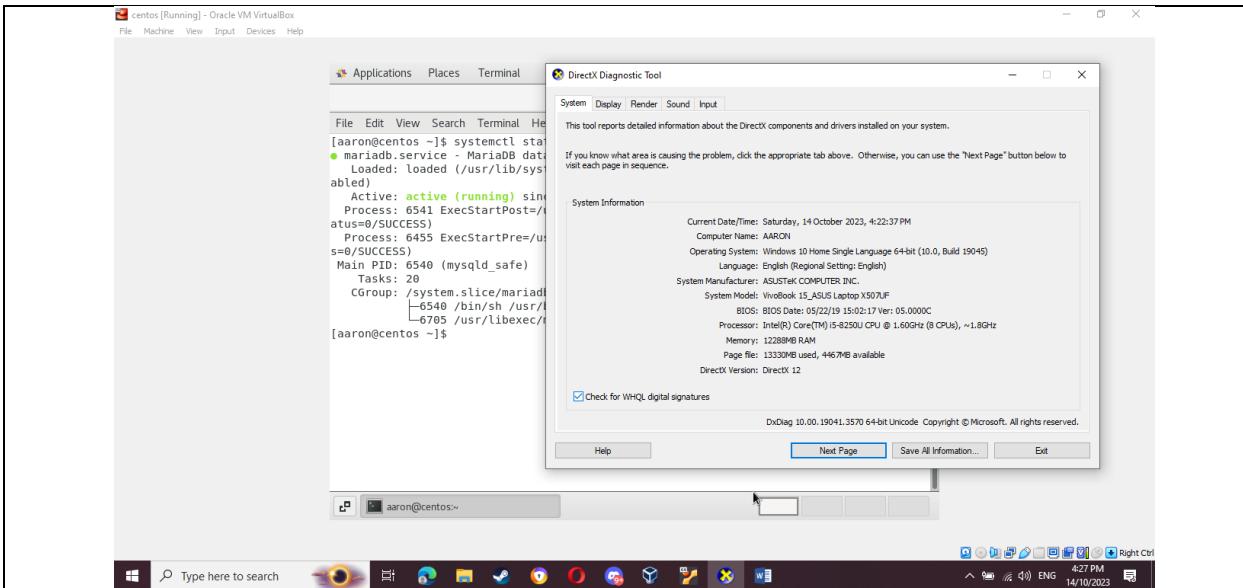


The order of task had to be rearranged for it to work.

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.

Describe the output.





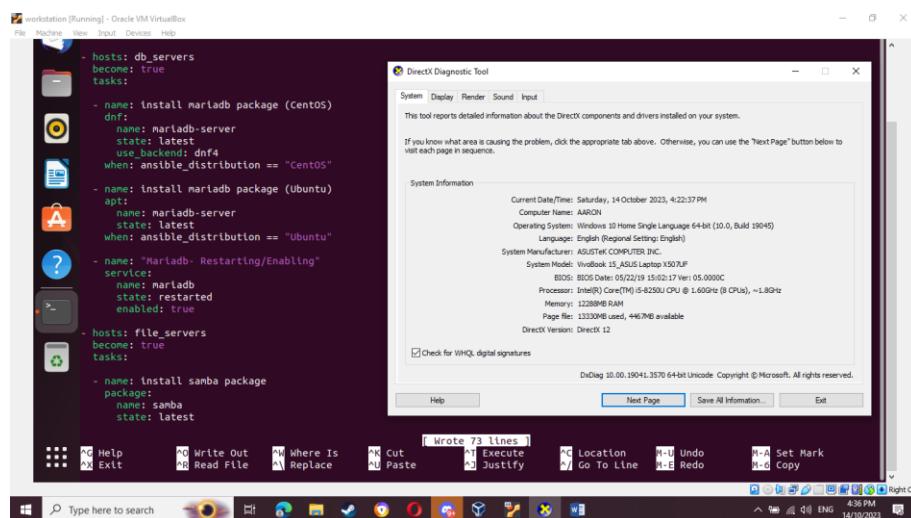
The mariadb server is active.

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.

The screenshot shows a Windows desktop environment. In the foreground, there is a window titled "workstation [Running] - Oracle VM VirtualBox". Inside this window, an Ansible playbook is being run against two hosts, 192.168.56.112 and 192.168.56.111. The tasks listed include installing apache and php for CentOS servers, gathering facts, installing mariadb packages (CentOS and Ubuntu), restarting mariadb, and installing samba. The output shows various task statuses like "ok", "changed", and "skipped". Below the tasks, a "PLAY RECAP" section provides a summary of the results for each host. In the background, a "DirectX Diagnostic Tool" window is open, displaying system information such as the current date/time (Saturday, 14 October 2023, 4:22:37 PM), operating system (Windows 10 Home Single Language 64-bit (10.0, Build 19045)), processor (Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz (8 CPUs), ~1.60GHz), and memory (12288MB RAM). The tool also includes tabs for System, Display, Render, Sound, and Input.

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
```

workstation [Running] - Oracle VM VirtualBox

```

- name: install updates (CentOS)
  tags: always
  dnf:
    - update_only: yes
    - update_cache: yes
    - use_backend: dnf4
  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:

```

Help Write Out Where Is Cut Execute Location Undo Set Mark

Exit Read File Replace Copy

Type here to search

System Information

Current Date/Time: Saturday, 14 October 2023, 4:22:37 PM
Computer Name: AARON
Operating System: Windows 10 Home Single Language 64-bit (10.0, Build 19045)
Language: English (Regional Setting: English)
System Manufacturer: ASUSTeK COMPUTER INC.
System Model: VivoBook 15_ASUS Laptop X507UF
BIOS: BIOS Date: 05/22/19 15:02:17 Ver: 05.0000C
Processor: Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz (8 CPUs), ~1.8GHz
Memory: 12288MB RAM
Page file: 13330MB used, 4467MB available
DirectX Version: DirectX 12

Check for WHQL digital signatures

DxDiag 10.00.19041.3570 64-bit Unicode Copyright © Microsoft. All rights reserved.

Help Next Page Save All Information... Exit

Right Ctrl

workstation [Running] - Oracle VM VirtualBox

```

- hosts: db_servers
  become: true
  tasks:
    - name: install mariadb package (CentOS)
      tags: centos, db,mariadb
      dnf:
        name: mariadb-server
        state: latest
        use_backend: dnf4
      when: ansible_distribution == "CentOS"

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

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

    - hosts: file_servers
      become: true
      tasks:
        - name: install samba package
          tags: samba
          package:
            name: samba

```

Help Write Out Where Is Cut Execute Location Undo Set Mark

Exit Read File Replace Copy

Type here to search

System Information

Current Date/Time: Saturday, 14 October 2023, 4:22:37 PM
Computer Name: AARON
Operating System: Windows 10 Home Single Language 64-bit (10.0, Build 19045)
Language: English (Regional Setting: English)
System Manufacturer: ASUSTeK COMPUTER INC.
System Model: VivoBook 15_ASUS Laptop X507UF
BIOS: BIOS Date: 05/22/19 15:02:17 Ver: 05.0000C
Processor: Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz (8 CPUs), ~1.8GHz
Memory: 12288MB RAM
Page file: 13330MB used, 4467MB available
DirectX Version: DirectX 12

Check for WHQL digital signatures

DxDiag 10.00.19041.3570 64-bit Unicode Copyright © Microsoft. All rights reserved.

Help Next Page Save All Information... Exit

Right Ctrl

Make sure to save the file and exit.
Run the [site.yml](#) file and describe the result.

The screenshot shows a Windows desktop environment. In the foreground, there is a terminal window titled "workstation [Running] - Oracle VM VirtualBox". The terminal is displaying the output of an Ansible playbook named "site.yml". The output shows tasks being run on two hosts with IP addresses 192.168.56.112 and 192.168.56.111. The tasks include installing Apache, PHP, MariaDB, Samba, and file servers. The status for most tasks is "ok".

In the background, a "DirectX Diagnostic Tool" window is open. It displays system information such as the current date/time (Saturday, 14 October 2023, 4:22:37 PM), computer name (AARON), operating system (Windows 10 Home Single Language 64-bit (10.0, Build 19045)), processor (Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz (8 CPUs), ~1.8GHz), memory (1288MB RAM), and page file (13330MB used, 4467MB available). The DirectX version is listed as Directx 12.

The result is the same as the previous outcome

2. On the local machine, try to issue the following commands and describe each result:

2.1 *ansible-playbook --list-tags site.yml*

The screenshot shows a Windows desktop environment. In the foreground, there is a terminal window titled "workstation [Running] - Oracle VM VirtualBox". The terminal is displaying the output of the command "ansible-playbook --list-tags site.yml". The output lists the tags defined in the playbook: "file_servers", "gather_facts", "install_samba_package", "PLAY RECAP", "playbook: site.yml", "play #1 (all: all)", "play #2 (web_servers:)", "play #3 (db_servers:)", "play #4 (file_servers:)", "play #5 (centos:)", and "PLAY [all]".

In the background, a "DirectX Diagnostic Tool" window is open, showing the same system information as the previous screenshot.

Shows the playbooks list of tags

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

```

workstation [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
TASK TAGS: [apache, apache2, centos, httpd, ubuntu]
play #3 (db_servers): db_servers TAGS: []
TASK TAGS: [centos, db, mariadb, ubuntu]
play #4 (file_servers): file_servers TAGS: []
TASK TAGS: [samba]
aaron@workstation:/CPE232_aaron$ ansible-playbook --tags centos --ask-become-pass site.yml
BECOME password:

PLAY [all] *****
TASK [Gathering Facts] *****
ok: [192.168.56.112]
ok: [192.168.56.111]

TASK [Install updates (CentOS)] *****
skipping: [192.168.56.112]
ok: [192.168.56.111]

TASK [Install updates (Ubuntu)] *****
skipping: [192.168.56.111]
ok: [192.168.56.112]
ok: [192.168.56.111]

PLAY [web_servers] *****
TASK [Gathering Facts] *****
ok: [192.168.56.112]
ok: [192.168.56.111]

TASK [Install apache and php for CentOS server]
skipping: [192.168.56.112]
ok: [192.168.56.111]

TASK [Install apache and php for Ubuntu server]
skipping: [192.168.56.112]
ok: [192.168.56.111]

PLAY [db_servers] *****
TASK [Gathering Facts] *****

```

DirectX Diagnostic Tool

System Information

Current Date/Time: Saturday, 14 October 2023, 4:22:37 PM
Computer Name: AARON
Operating System: Windows 10 Home Single Language 64-bit (10.0, Build 19045)
Language: English (Regional Setting: English)
System Manufacturer: ASUSTeK COMPUTER INC.
System Model: VivoBook 15_ASUS Laptop X507UF
BIOS: BIOS Date: 05/22/19 15:02:17 Ver: 05.0000C
Processor: Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz (8 CPUs), ~1.6GHz
Memory: 12288MB RAM
Page file: 13330MB used, 4467MB available
DirectX Version: DirectX 12

Check for WHQL digital signatures

DxDiag 10.00.19041.3570 64-bit Unicode Copyright © Microsoft. All rights reserved.

Runs playbook with the centos tag

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

```

workstation [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
PLAY RECAP *****
192.168.56.111 : ok=6 changed=0 unreachable=0 failed=0 skipped=1 rescued=0 ignored=0
192.168.56.112 : ok=5 changed=0 failed=0 skipped=0 rescued=0 ignored=0

aaron@workstation:/CPE232_aaron$ ansible-playbook --tags db --ask-become-pass site.yml
BECOME password:

PLAY [all] *****
TASK [Gathering Facts] *****
ok: [192.168.56.112]
ok: [192.168.56.111]

TASK [Install updates (CentOS)] *****
skipping: [192.168.56.112]
ok: [192.168.56.111]

TASK [Install updates (Ubuntu)] *****
skipping: [192.168.56.111]
ok: [192.168.56.112]
ok: [192.168.56.111]

PLAY [web_servers] *****
TASK [Gathering Facts] *****
ok: [192.168.56.112]
ok: [192.168.56.111]

TASK [Install apache and php for CentOS server]
skipping: [192.168.56.112]
ok: [192.168.56.111]

TASK [Install apache and php for Ubuntu server]
skipping: [192.168.56.112]
ok: [192.168.56.111]

PLAY [db_servers] *****
TASK [Install mariadb package (CentOS)] *****
skipping: [192.168.56.112]
ok: [192.168.56.111]


```

DirectX Diagnostic Tool

System Information

Current Date/Time: Saturday, 14 October 2023, 4:22:37 PM
Computer Name: AARON
Operating System: Windows 10 Home Single Language 64-bit (10.0, Build 19045)
Language: English (Regional Setting: English)
System Manufacturer: ASUSTeK COMPUTER INC.
System Model: VivoBook 15_ASUS Laptop X507UF
BIOS: BIOS Date: 05/22/19 15:02:17 Ver: 05.0000C
Processor: Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz (8 CPUs), ~1.6GHz
Memory: 12288MB RAM
Page file: 13330MB used, 4467MB available
DirectX Version: DirectX 12

Check for WHQL digital signatures

DxDiag 10.00.19041.3570 64-bit Unicode Copyright © Microsoft. All rights reserved.

Runs playbook with the db tag

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

```

workstation [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
192.168.56.111 : ok=5 changed=0 unreachable=0 failed=0 skipped=2 rescued=0 ignored=0
192.168.56.112 : ok=6 changed=0 unreachable=0 failed=0 skipped=2 rescued=0 ignored=0

aaron@workstation:~/CPE232_aaron$ ansible-playbook --tags apache --ask-become-pass site.yml
BECOME password:

PLAY [all] *****
  TASK [Gathering Facts] *****
    ok: [192.168.56.112]
    ok: [192.168.56.111]

  TASK [Install updates (CentOS)] *****
    skipping: [192.168.56.112]
    ok: [192.168.56.111]

  TASK [Install updates (Ubuntu)] *****
    skipping: [192.168.56.111]
    ok: [192.168.56.112]

  PLAY [web_servers] *****
    TASK [Gathering Facts] *****
      ok: [192.168.56.112]
      ok: [192.168.56.111]

    TASK [Install apache and php for Ubuntu server]
      skipping: [192.168.56.111]
      ok: [192.168.56.112]

    TASK [Install apache and php for CentOS server]
      skipping: [192.168.56.112]
      ok: [192.168.56.111]

    PLAY [db_servers] *****
      TASK [Gathering Facts] *****
        ok: [192.168.56.112]
        ok: [192.168.56.111]

```

Runs playbook with the apache tag

2.5 `ansible-playbook --tags "apache,db" --ask-become-pass site.yml`

```

workstation [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
PLAY [file_servers] *****
  TASK [Gathering Facts] *****
    ok: [192.168.56.112]

  PLAY RECAP *****
  192.168.56.111 : ok=5 changed=0 unreachable=0 failed=0 skipped=2 rescued=0 ignored=0
  192.168.56.112 : ok=6 changed=0 unreachable=0 failed=0 skipped=2 rescued=0 ignored=0

aaron@workstation:~/CPE232_aaron$ ansible-playbook --tags apache,db --ask-become-pass site.yml
BECOME password:

PLAY [all] *****
  TASK [Gathering Facts] *****
    ok: [192.168.56.112]
    ok: [192.168.56.111]

  TASK [Install updates (CentOS)] *****
    skipping: [192.168.56.112]
    ok: [192.168.56.111]

  TASK [Install updates (Ubuntu)] *****
    skipping: [192.168.56.111]
    ok: [192.168.56.112]

  PLAY [web_servers] *****
    TASK [Gathering Facts] *****
      ok: [192.168.56.112]
      ok: [192.168.56.111]

    TASK [Install apache and php for Ubuntu server]
      skipping: [192.168.56.111]
      ok: [192.168.56.112]

    TASK [Install apache and php for CentOS server]
      skipping: [192.168.56.112]
      ok: [192.168.56.111]

    TASK [Install apache and php for CentOS server]
      skipping: [192.168.56.111]
      ok: [192.168.56.112]

  PLAY [db_servers] *****
    TASK [Gathering Facts] *****
      ok: [192.168.56.112]
      ok: [192.168.56.111]

```

Runs playbook with the apache and db tag

Task 3: Managing Services

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

```

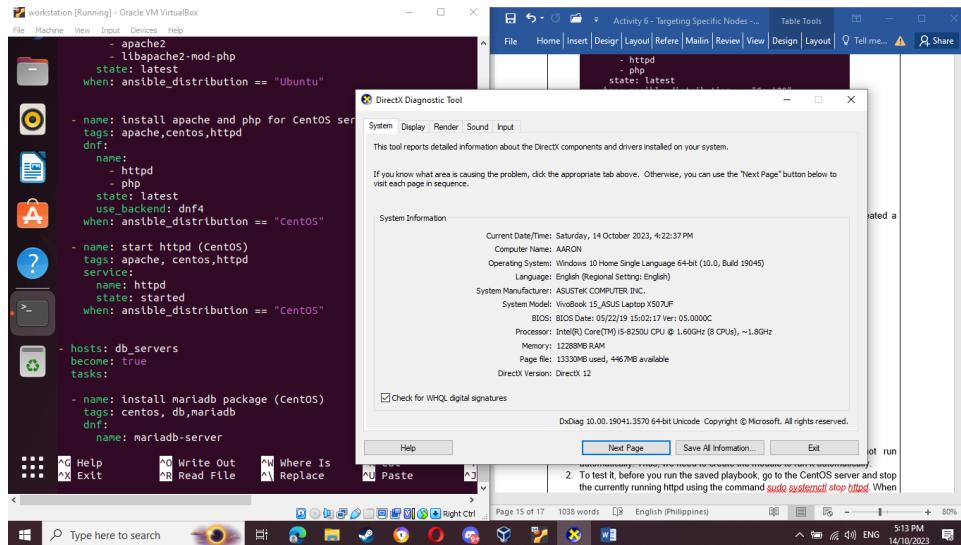
- name: install apache and php for CentOS servers
  tags: apache,centos,httpd
  dnf:
    name:
      - httpd
      - php
    state: latest
  when: ansible_distribution == "CentOS"

- 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"

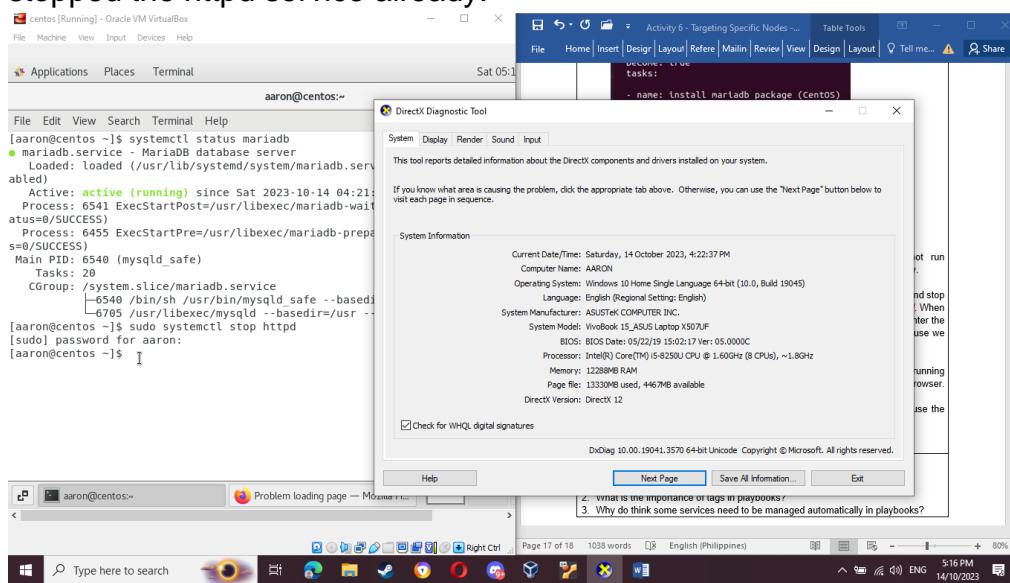
    - name: "Mariadb- Restarting/Enabling"
      service:
        name: mariadb
        state: restarted
        enabled: true

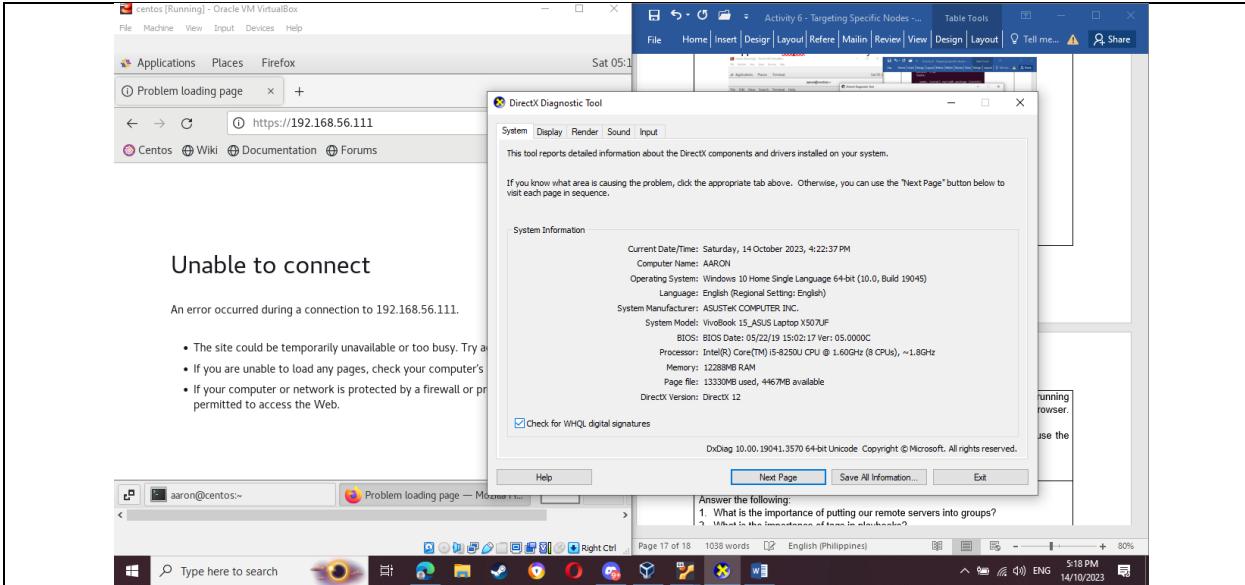
```

Figure 3.1.2

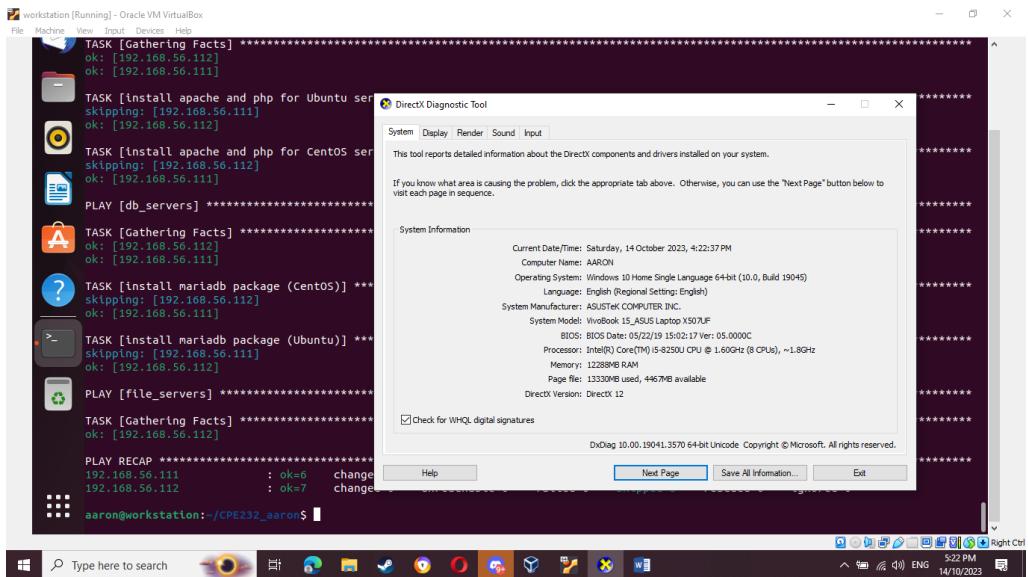
This is because in CentOS, installed packages' services are not run automatically. Thus, we need to create the module to run it automatically.

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.

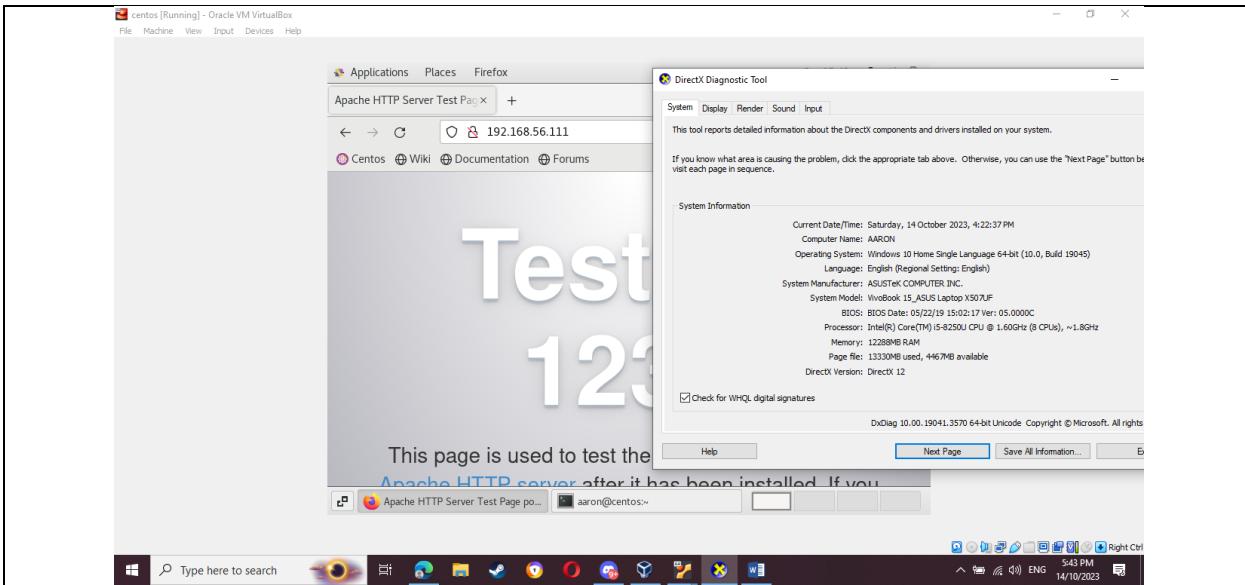




3. Go to the local machine and this time, run the **site.yml** file. Then after running the file, go again to the CentOS server and enter its IP address on the browser. Describe the result.



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?

Gathering distant servers is fundamental for proficient server the board as it permits chairmen to apply setups and execute undertakings on unambiguous arrangements of servers with comparative jobs or attributes, upgrading association and improving on upkeep.

2. What is the importance of tags in playbooks?

Tags in playbooks give a significant approach to specifically run explicit undertakings or jobs on servers, empowering fine-grained command over computerization processes and guaranteeing that the important activities are executed, reducing the risk of unintended consequences.

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

A few administrations require mechanized administration in playbooks to guarantee quick reaction to evolving conditions, keep up with framework security, and limit human mediation, which can be mistake inclined and tedious for dull undertakings, improving generally speaking framework unwavering quality.

