

Name: Tendencia, Jasmin Raiza S.	Date Performed: 10/02/2023
Course/Section: CPE31S4	Date Submitted: 10/03/2023
Instructor: Dr. Jonathan Taylar	Semester and SY: 1st/2023 - 2024
Activity 6: Targeting Specific Nodes and Managing Services	
<p>1. Objectives:</p> <ul style="list-style-type: none"> 1.1 Individualize hosts 1.2 Apply tags in selecting plays to run 1.3 Managing Services from remote servers using playbooks 	
<p>2. Discussion:</p> <p>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.</p> <p>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.</p> <p>Requirement:</p> <p>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 <i>ssh-copy-id</i> to copy the public key to Server 3. Verify if you can successfully SSH to Server 3.</p>	
Task 1: Targeting Specific Nodes	
<ul style="list-style-type: none"> 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"

```

```

---
- 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"

```

Figure 1.1. Creation of new playbook named site.yml

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
```

```
GNU nano 6.2 inventory
[web_servers]
192.168.56.102 ansible_python_interpreter=/usr/bin/python3
192.168.56.104 ansible_python_interpreter=/usr/bin/python

[db_servers]
192.168.56.104 ansible_python_interpreter=/usr/bin/python
192.168.56.103 ansible_python_interpreter=/usr/bin/python3

[file_servers]
192.168.56.103 ansible_python_interpreter=/usr/bin/python3
```

Figure 1.2. Edited inventory file

Input: The web_servers are: server 1 (192.168.56.102) and centOS (192.168.56.104); the db_servers are: centOS and server 2 (192.168.56.103); and server 2 in the file_servers.

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"
```

```

--
- hosts: all
  become: true
  pre_tasks:

    - name: install updates (CentOS)
      yum:
        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
        update_cache: yes
      when: ansible_distribution == "Ubuntu"

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

```

Figure 1.3. Edited site.yml

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.

```
tendencia@workstation:~/H0A6$ 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: [192.168.56.104]

TASK [install updates (CentOS)] *****
skipping: [192.168.56.102]
skipping: [192.168.56.103]
ok: [192.168.56.104]

TASK [install updates (Ubuntu)] *****
skipping: [192.168.56.104]
ok: [192.168.56.103]
ok: [192.168.56.102]

PLAY [web_servers] *****

TASK [Gathering Facts] *****
ok: [192.168.56.102]
ok: [192.168.56.104]

TASK [install apache and php for Ubuntu servers] *****
skipping: [192.168.56.104]
ok: [192.168.56.102]

TASK [install apache and php for CentOS servers] *****
skipping: [192.168.56.102]
ok: [192.168.56.104]

PLAY RECAP *****
192.168.56.102      : ok=4    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.56.103      : ok=2    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
192.168.56.104      : ok=4    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
```

Figure 1.4. Result after running site.yml

Observation: *PLAY [all]*: This is the beginning of the Ansible playbook. It specifies which hosts (all hosts) the playbook should run on. The tasks did run in the *Gathering Facts* without any errors, *TASK [install updates (CentOS)]*: This task is intended to install updates on CentOS hosts (192.168.56.104) using the yum/dnf package manager. It skips the task on Ubuntu hosts (192.168.56.102 and 192.168.56.103) because it's conditioned to run only on CentOS hosts. *TASK [install updates (Ubuntu)]*: this one is intended to install updates on Ubuntu hosts using the apt package manager. It skips the task on the CentOS host because it's conditioned to run only on Ubuntu hosts. *PLAY [web_servers]*: second play of playbook, which targets the web_servers group of hosts, then gathering facts. *TASK [install apache and php for Ubuntu servers]*: intended to install Apache and PHP on Ubuntu servers using the apt package manager. It skips the task on the CentOS host because it's conditioned to run only on Ubuntu servers. *TASK [install apache and php for CentOS servers]*: intended to install Apache and PHP on CentOS server using the yum/dnf package manager. It skips the task on the Ubuntu host because it's conditioned to run only on CentOS servers. *PLAY RECAP*: summary of what happened during the playbook run for each host, there are 4, 2, and 4 ok for server 1, 2, and CentOS, respectively. To sum it up, all tasks related to package installation were skipped on certain hosts based on the distribution-specific conditions specified in the playbook.

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

Make sure to save the file and exit.

```

- hosts: web_servers
  become: true
  tasks:

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

    - name: install apache and php for CentOS server
      yum:
        name:
          - httpd
          - php
        state: latest
      when: ansible_distribution == "CentOS"

- hosts: db_servers
  become: true
  tasks:

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

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

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

```

Figure 1.5. Adding db_servers in site.yml

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


```
tendencia@workstation:~/H04a$ ansible-playbook --ask-become-pass site.yml
BECOME password:

PLAY [all] *****

TASK [Gathering Facts] *****
ok: [192.168.56.104]
ok: [192.168.56.103]
ok: [192.168.56.102]

TASK [install updates (CentOS)] *****
skipping: [192.168.56.102]
skipping: [192.168.56.103]
ok: [192.168.56.104]

TASK [install updates (Ubuntu)] *****
skipping: [192.168.56.104]
ok: [192.168.56.103]
ok: [192.168.56.102]

PLAY [web_servers] *****

TASK [Gathering Facts] *****
ok: [192.168.56.102]
ok: [192.168.56.104]

TASK [install apache and php for Ubuntu servers] *****
skipping: [192.168.56.104]
ok: [192.168.56.102]

TASK [install apache and php for CentOS servers] *****
skipping: [192.168.56.102]
ok: [192.168.56.104]

PLAY [db_servers] *****

TASK [Gathering Facts] *****
ok: [192.168.56.103]
ok: [192.168.56.104]

TASK [install mariadb package (CentOS)] *****
skipping: [192.168.56.103]
ok: [192.168.56.104]

TASK [install mariadb package (Ubuntu)] *****
skipping: [192.168.56.104]
ok: [192.168.56.103]

TASK [Mariadb - Restarting/Enabling] *****
changed: [192.168.56.103]
changed: [192.168.56.104]

PLAY RECAP *****
192.168.56.102      : ok=4    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.56.103      : ok=5    changed=1    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.56.104      : ok=7    changed=1    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0
```

Figure 1.6. Result of modified site.yml

Observation: All tasks related to package installation were skipped on certain hosts based on the distribution-specific conditions specified in the playbook. Also, *PLAY [db_servers]*: is the start of a new play in Ansible playbook, targeting the db_servers group of hosts. *TASK [install mariadb package (CentOS)]*: intended to install the MariaDB package on CentOS hosts using the yum package manager. It skips the task on the Ubuntu host because it's conditioned to run only on CentOS hosts. On CentOS, the task indicates a change, meaning the MariaDB package was installed. *TASK [install mariadb package (Ubuntu)]*: intended to install the MariaDB package on Ubuntu hosts using the apt package manager. It skips the task on the CentOS host because it's conditioned to run only on Ubuntu hosts. On server 2, the task indicates success (ok), meaning the MariaDB package was already installed or successfully

installed. **TASK [Mariadb package Restarting/Enabling]:** This task is intended to restart and enable the MariaDB service on both CentOS and server 3. It shows as "changed" on both hosts, indicating that the task had to make changes by restarting and enabling the service. This task ensures that the MariaDB service is running and set to start automatically on boot.

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.

Ubuntu Local Machine:

```
tendencia@workstation:~/HOA6$ sudo systemctl status mariadb
● mariadb.service - MariaDB 10.6.12 database server
   Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor preset: enable
   Active: active (running) since Tue 2023-10-03 17:13:27 PST; 1min 2s ago
     Docs: man:mariadb(8)
           https://mariadb.com/kb/en/library/systemd/
   Main PID: 4475 (mariabdd)
    Status: "Taking your SQL requests now..."
     Tasks: 11 (limit: 2261)
    Memory: 67.5M
       CPU: 436ms
    CGroup: /system.slice/mariadb.service
           └─4475 /usr/sbin/mariabdd

Oct 03 17:13:27 workstation mariabdd[4475]: 2023-10-03 17:13:27 0 [Note] InnoDB: 10.6.12
Oct 03 17:13:27 workstation mariabdd[4475]: 2023-10-03 17:13:27 0 [Note] Plugin 'FEEDBAC
Oct 03 17:13:27 workstation mariabdd[4475]: 2023-10-03 17:13:27 0 [Note] InnoDB: Loading
Oct 03 17:13:27 workstation mariabdd[4475]: 2023-10-03 17:13:27 0 [Warning] You need to >
Oct 03 17:13:27 workstation mariabdd[4475]: 2023-10-03 17:13:27 0 [Note] Server socket c>
Oct 03 17:13:27 workstation mariabdd[4475]: 2023-10-03 17:13:27 0 [Note] InnoDB: Buffer >
Oct 03 17:13:27 workstation mariabdd[4475]: 2023-10-03 17:13:27 0 [Note] /usr/sbin/maria>
Oct 03 17:13:27 workstation mariabdd[4475]: Version: '10.6.12-MariaDB-0ubuntu0.22.04.1' >
Oct 03 17:13:27 workstation systemd[1]: Started MariaDB 10.6.12 database server.
Oct 03 17:13:27 workstation /etc/mysql/debian-start[4505]: Checking for insecure root ac>
lines 1-23/23 (END)
```

CentOS server:

```
[tendencia@centoslocal ~]$ sudo systemctl status mariadb
● mariadb.service - MariaDB database server
   Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; vendor preset: dis
  abled)
   Active: active (running) since Tue 2023-10-03 16:40:11 PST; 13min ago
     Main PID: 5331 (mysqld_safe)
    CGroup: /system.slice/mariadb.service
           └─5331 /bin/sh /usr/bin/mysqld_safe --basedir=/usr
             └─5496 /usr/libexec/mysqld --basedir=/usr --datadir=/var/lib/mysql --plug...

Oct 03 16:40:07 centoslocal systemd[1]: Starting MariaDB database server...
Oct 03 16:40:07 centoslocal mariadb-prepare-db-dir[5297]: Database MariaDB is probab...
Oct 03 16:40:07 centoslocal mariadb-prepare-db-dir[5297]: If this is not the case, m...
Oct 03 16:40:08 centoslocal mysqld_safe[5331]: 231003 16:40:08 mysqld_safe Logging ....
Oct 03 16:40:08 centoslocal mysqld_safe[5331]: 231003 16:40:08 mysqld_safe Starting...l
Oct 03 16:40:11 centoslocal systemd[1]: Started MariaDB database server.
Hint: Some lines were ellipsized, use -l to show in full.
[tendencia@centoslocal ~]$
```

Observation: It indicates that both Ubuntu and CentOS machines show that the MariaDB database server is active and running.

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.

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

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

Figure 1.7. Adding file_servers in site.yml

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

```
tendencia@workstation:~/H04a$ ansible-playbook --ask-become-pass site.yml
BECOME password:

PLAY [all] *****

TASK [Gathering Facts] *****
ok: [192.168.56.104]
ok: [192.168.56.103]
ok: [192.168.56.102]

TASK [install updates (CentOS)] *****
skipping: [192.168.56.102]
skipping: [192.168.56.103]
ok: [192.168.56.104]

TASK [install updates (Ubuntu)] *****
skipping: [192.168.56.104]
ok: [192.168.56.102]
ok: [192.168.56.103]

PLAY [web_servers] *****

TASK [Gathering Facts] *****
ok: [192.168.56.102]
ok: [192.168.56.104]

TASK [install apache and php for Ubuntu servers] *****
skipping: [192.168.56.104]
ok: [192.168.56.102]

TASK [install apache and php for CentOS servers] *****
skipping: [192.168.56.102]
ok: [192.168.56.104]

PLAY [db_servers] *****

TASK [Gathering Facts] *****
ok: [192.168.56.103]
ok: [192.168.56.104]

TASK [install mariadb package (CentOS)] *****
skipping: [192.168.56.103]
ok: [192.168.56.104]

TASK [install mariadb package (Ubuntu)] *****
skipping: [192.168.56.104]
ok: [192.168.56.103]

TASK [Mariadb- Restarting/Enabling] *****
changed: [192.168.56.103]
changed: [192.168.56.104]

PLAY [file_servers] *****

TASK [Gathering Facts] *****
ok: [192.168.56.103]

TASK [install samba package] *****
changed: [192.168.56.103]

PLAY RECAP *****
192.168.56.102      : ok=4    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.56.103      : ok=7    changed=2    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.56.104      : ok=7    changed=1    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0
```

Figure 1.8. Installation of the `file_servers` group

Observation: The Samba package was successfully installed on host 192.168.56.103 (server 2), and all other tasks on various hosts either completed successfully or were skipped because it is not relevant to those hosts.

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 package (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.

```
---
- hosts: all
  become: true
  pre_tasks:

  - name: install updates (CentOS)
    tags: always
    yum:
      name: '*'
      state: latest
    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
      update_cache: yes
    when: ansible_distribution == "Ubuntu"

  - name: install apache and php for CentOS servers
    tags: apache,centos,httpd
    yum:
      name:
        - httpd
        - php
      state: latest
    when: ansible_distribution == "CentOS" and ansible_distribution_major_version == "7"
```



```

- hosts: db_servers
  become: true
  tasks:

    - name: install mariadb package (CentOS)
      tags: centos, db, mariadb
      yum:
        name: mariadb-server
        state: latest
      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
        state: latest

```

Figure 2.1. Placing of tags in the site.yml

Latest edit:

```

- hosts: all
  become: true
  pre_tasks:

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

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

```

```

- hosts: db_servers
  become: true
  tasks:

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

  - name: install mariadb package (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
    package:
      name: samba
      state: latest

```

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

```
tendencia@workstation:~/H04$ ansible-playbook --ask-become-pass site.yml
BECOME password:

PLAY [all] *****

TASK [Gathering Facts] *****
ok: [192.168.56.104]
ok: [192.168.56.102]
ok: [192.168.56.103]

TASK [install updates (CentOS)] *****
skipping: [192.168.56.102]
skipping: [192.168.56.103]
ok: [192.168.56.104]

TASK [install updates (Ubuntu)] *****
skipping: [192.168.56.104]
ok: [192.168.56.102]
ok: [192.168.56.103]

PLAY [web_servers] *****

TASK [Gathering Facts] *****
ok: [192.168.56.102]
ok: [192.168.56.104]

TASK [install apache and php for Ubuntu servers] *****
skipping: [192.168.56.104]
ok: [192.168.56.102]

TASK [install apache and php for CentOS servers] *****
skipping: [192.168.56.102]
ok: [192.168.56.104]

PLAY [db_servers] *****

TASK [Gathering Facts] *****
ok: [192.168.56.103]
ok: [192.168.56.104]

TASK [install mariadb package (CentOS)] *****
skipping: [192.168.56.103]
ok: [192.168.56.104]

TASK [install mariadb package (Ubuntu)] *****
skipping: [192.168.56.104]
ok: [192.168.56.103]

TASK [Mariadb- Restarting/Enabling] *****
changed: [192.168.56.103]
changed: [192.168.56.104]

PLAY [file_servers] *****

TASK [Gathering Facts] *****
ok: [192.168.56.103]

TASK [install samba package] *****
ok: [192.168.56.103]

PLAY RECAP *****
192.168.56.102      : ok=4    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.56.103      : ok=7    changed=1    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.56.104      : ok=7    changed=1    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0

tendencia@workstation:~/H04$
```

Figure 2.2. Running site.yml after placing tags

Observation: The use of tags allows the control of the execution of tasks on which hosts, which makes the playbook more flexible. In 192.168.56.102: it had a total of four tasks in the playbook and all were completed successfully ("ok"). There were no changes ("changed") or failures ("failed"). Two tasks were skipped because they were not relevant to this host, and they were skipped due to the use of tags. In 192.168.56.103: This host had a total of seven tasks in the playbook, and one of them resulted in a change. The "ok" count indicates that six tasks were executed successfully. There were no failures. Two tasks were skipped due to tags, meaning they didn't apply to this host. In 192.168.56.104: Similar to host 192.168.56.103, this host also had seven tasks in the playbook, and one of them resulted in a change. The "ok" count indicates that six tasks were executed successfully. There were no failures. Three tasks were skipped due to tags, indicating they were not relevant to this host.

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

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

```
tendencia@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_servers): file_servers TAGS: []
    TASK TAGS: [samba]
tendencia@workstation:~/HOA6$
```

Observation: In this result, the playbook contains four plays: one for all, one for web_servers, one for db_servers, and one for file_servers. In this, there are no tags explicitly defined for any tasks.

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

```
tendencia@workstation:~/H04$ ansible-playbook --tags centos --ask-become-pass site.yml
BECOME password:

PLAY [all] *****

TASK [Gathering Facts] *****
ok: [192.168.56.104]
ok: [192.168.56.102]
ok: [192.168.56.103]

TASK [install updates (CentOS)] *****
skipping: [192.168.56.102]
skipping: [192.168.56.103]
ok: [192.168.56.104]

TASK [install updates (Ubuntu)] *****
skipping: [192.168.56.104]
ok: [192.168.56.102]
ok: [192.168.56.103]

PLAY [web_servers] *****

TASK [Gathering Facts] *****
ok: [192.168.56.102]
ok: [192.168.56.104]

TASK [install apache and php for CentOS servers] *****
skipping: [192.168.56.102]
ok: [192.168.56.104]

PLAY [db_servers] *****

TASK [Gathering Facts] *****
ok: [192.168.56.103]
ok: [192.168.56.104]

TASK [install mariadb package (CentOS)] *****
skipping: [192.168.56.103]
ok: [192.168.56.104]

PLAY [file_servers] *****

TASK [Gathering Facts] *****
ok: [192.168.56.103]

PLAY RECAP *****
192.168.56.102      : ok=3    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
192.168.56.104      : ok=6    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0

tendencia@workstation:~/H04$
```

Observation: The playbook starts by gathering facts for all hosts. It then checks if the centos tag is applied to the "install updates (CentOS)" task, which it is. Next, it proceeds to the web_servers play, gathers facts for the web servers, and checks for tasks tagged with centos. Since the "install apache and php for CentOS servers" task has the centos tag, it gets executed. The result shows that only tasks with the centos tag were executed, and tasks without this tag were skipped for the specified hosts.

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

```

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

PLAY [all] *****

TASK [Gathering Facts] *****
ok: [192.168.56.104]
ok: [192.168.56.102]
ok: [192.168.56.103]

TASK [install updates (CentOS)] *****
skipping: [192.168.56.102]
skipping: [192.168.56.103]
ok: [192.168.56.104]

TASK [install updates (Ubuntu)] *****
skipping: [192.168.56.104]
ok: [192.168.56.102]
ok: [192.168.56.103]

PLAY [web_servers] *****

TASK [Gathering Facts] *****
ok: [192.168.56.102]
ok: [192.168.56.104]

PLAY [db_servers] *****

TASK [Gathering Facts] *****
ok: [192.168.56.102]
ok: [192.168.56.104]

TASK [install mariadb package (CentOS)] *****
skipping: [192.168.56.103]
ok: [192.168.56.104]

TASK [install mariadb package (Ubuntu)] *****
skipping: [192.168.56.104]
ok: [192.168.56.103]

PLAY [file_servers] *****

TASK [Gathering Facts] *****
ok: [192.168.56.103]

PLAY RECAP *****
192.168.56.102 : ok=3  changed=0  unreachable=0  failed=0  skipped=1  rescued=0  ignored=0
192.168.56.103 : ok=5  changed=0  unreachable=0  failed=0  skipped=2  rescued=0  ignored=0
192.168.56.104 : ok=5  changed=0  unreachable=0  failed=0  skipped=2  rescued=0  ignored=0

vondencia@workstation: ~/db$

```

Observation: In this result, the playbook starts by gathering facts for all hosts. It then checks if the db tag is applied to the tasks related to database installation, which it is. Next, it proceeds to the db_servers play, where it gathers facts for the database servers. The tasks tagged with db are executed for the specified hosts, and they include installing MariaDB packages and restarting/enabling the MariaDB service. The result shows that only tasks with the db tag were executed, and tasks without this tag were skipped for the specified hosts.

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

```
tendencia@workstation: ~/HOM$ ansible-playbook --tags apache --ask-become-pass site.yml
BECOME password:

PLAY [all] *****

TASK [Gathering Facts] *****
ok: [192.168.56.104]
ok: [192.168.56.102]
ok: [192.168.56.103]

TASK [Install updates (CentOS)] *****
skipping: [192.168.56.102]
skipping: [192.168.56.103]
ok: [192.168.56.104]

TASK [Install updates (Ubuntu)] *****
skipping: [192.168.56.104]
ok: [192.168.56.102]
ok: [192.168.56.103]

PLAY [web_servers] *****

TASK [Gathering Facts] *****
ok: [192.168.56.102]
ok: [192.168.56.104]

TASK [Install apache and php for Ubuntu servers] *****
skipping: [192.168.56.104]
ok: [192.168.56.102]

TASK [Install apache and php for CentOS servers] *****
skipping: [192.168.56.102]
ok: [192.168.56.104]

PLAY [db_servers] *****

TASK [Gathering Facts] *****
ok: [192.168.56.103]
ok: [192.168.56.104]

PLAY [file_servers] *****

TASK [Gathering Facts] *****
ok: [192.168.56.103]

PLAY RECAP *****
192.168.56.102 : ok=4 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=1 rescued=0 ignored=0
192.168.56.104 : ok=5 changed=0 unreachable=0 failed=0 skipped=2 rescued=0 ignored=0

tendencia@workstation: ~/HOM$
```

Observation: The result shows that only tasks with the apache tag were executed, and tasks without this tag were skipped for the specified hosts.

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

```
tendencia@workstation: ~/HOM$ 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: [192.168.56.104]

TASK [Install updates (CentOS)] *****
skipping: [192.168.56.102]
skipping: [192.168.56.103]
ok: [192.168.56.104]

TASK [Install updates (Ubuntu)] *****
skipping: [192.168.56.104]
ok: [192.168.56.102]
ok: [192.168.56.103]

PLAY [web_servers] *****

TASK [Gathering Facts] *****
ok: [192.168.56.102]
ok: [192.168.56.104]

TASK [Install apache and php for Ubuntu servers] *****
skipping: [192.168.56.104]
ok: [192.168.56.102]

TASK [Install apache and php for CentOS servers] *****
skipping: [192.168.56.102]
ok: [192.168.56.104]

PLAY [db_servers] *****

TASK [Gathering Facts] *****
ok: [192.168.56.104]
ok: [192.168.56.103]

TASK [Install mariadb package (CentOS)] *****
skipping: [192.168.56.103]
ok: [192.168.56.104]

TASK [Install mariadb package (Ubuntu)] *****
skipping: [192.168.56.104]
ok: [192.168.56.103]

PLAY [file_servers] *****

TASK [Gathering Facts] *****
ok: [192.168.56.103]

PLAY RECAP *****
192.168.56.102 : ok=4 changed=0 unreachable=0 failed=0 skipped=2 rescued=0 ignored=0
192.168.56.103 : ok=5 changed=0 unreachable=0 failed=0 skipped=2 rescued=0 ignored=0
192.168.56.104 : ok=6 changed=0 unreachable=0 failed=0 skipped=3 rescued=0 ignored=0

tendencia@workstation: ~/HOM$
```

Observation: Upon running the playbook, it executes tasks that have been tagged with either the apache or 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.

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

Adding of start httpd in site.yml

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.

```

[tendencia@centoslocal ~]$ sudo systemctl stop httpd
[sudo] password for tendencia:
[tendencia@centoslocal ~]$

```

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.

```
tendencia@workstation: ~/M04$ ansible-playbook --ask-become-pass site.yml
BECOME password:

PLAY [all] *****

TASK [Gathering Facts] *****
ok: [192.168.56.104]
ok: [192.168.56.102]
ok: [192.168.56.103]

TASK [install updates (CentOS)] *****
skipping: [192.168.56.102]
skipping: [192.168.56.103]
ok: [192.168.56.104]

TASK [install updates (Ubuntu)] *****
skipping: [192.168.56.104]
ok: [192.168.56.102]
ok: [192.168.56.103]

PLAY [web_servers] *****

TASK [Gathering Facts] *****
ok: [192.168.56.104]
ok: [192.168.56.102]

TASK [install apache and php for Ubuntu servers] *****
skipping: [192.168.56.104]
ok: [192.168.56.102]

TASK [install apache and php for CentOS servers] *****
skipping: [192.168.56.102]
ok: [192.168.56.104]

TASK [start httpd (CentOS)] *****
skipping: [192.168.56.102]
changed: [192.168.56.104]

PLAY [db_servers] *****

TASK [Gathering Facts] *****
ok: [192.168.56.104]
ok: [192.168.56.103]

TASK [install mariadb package (CentOS)] *****
skipping: [192.168.56.103]
ok: [192.168.56.104]

TASK [install mariadb package (Ubuntu)] *****
skipping: [192.168.56.104]
ok: [192.168.56.103]

TASK [Mariadb- Restarting/Enabling] *****
changed: [192.168.56.103]
changed: [192.168.56.104]

PLAY [file_servers] *****

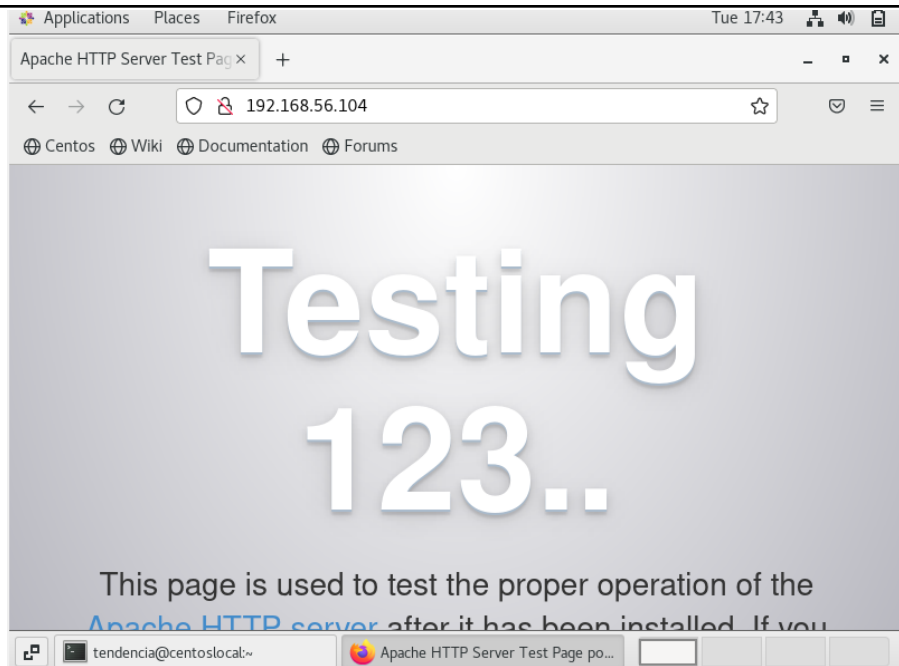
TASK [Gathering Facts] *****
ok: [192.168.56.103]

TASK [install samba package] *****
ok: [192.168.56.103]

PLAY RECAP *****
192.168.56.102      : ok=4    changed=0    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0
192.168.56.103      : ok=7    changed=1    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.56.104      : ok=9    changed=2    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0

tendencia@workstation: ~/M04$
```

Observation: In the output, you can see that after installing Apache and PHP on the CentOS server, it proceeds to start the httpd service, and the status changes to "changed" to indicate that the service was started as a result of the task.



Observation: The Apache (httpd) was successfully installed and started by the playbook, and there are no firewall rules or network issues blocking the connection, so the default Apache web page or the web page was shown upon entering the ip address of CentOS on the browser.

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.

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

I tried running it:

```
tendencia@workstation:~/HOA6$ ansible-playbook --ask-become-pass site.yml
BECOME password:

PLAY [all] *****

TASK [Gathering Facts] *****
ok: [192.168.56.103]
ok: [192.168.56.102]
ok: [192.168.56.104]

TASK [install updates (CentOS)] *****
skipping: [192.168.56.102]
skipping: [192.168.56.103]
ok: [192.168.56.104]

TASK [install updates (Ubuntu)] *****
skipping: [192.168.56.104]
ok: [192.168.56.103]
ok: [192.168.56.102]

PLAY [web_servers] *****

TASK [Gathering Facts] *****
ok: [192.168.56.102]
ok: [192.168.56.104]

TASK [install apache and php for Ubuntu servers] *****
skipping: [192.168.56.104]
ok: [192.168.56.102]

TASK [install apache and php for CentOS servers] *****
skipping: [192.168.56.102]
ok: [192.168.56.104]

TASK [start httpd (CentOS)] *****
skipping: [192.168.56.102]
ok: [192.168.56.104]
```

```

PLAY [db_servers] *****
TASK [Gathering Facts] *****
ok: [192.168.56.103]
ok: [192.168.56.104]

TASK [install mariadb package (CentOS)] *****
skipping: [192.168.56.103]
ok: [192.168.56.104]

TASK [install mariadb package (Ubuntu)] *****
skipping: [192.168.56.104]
ok: [192.168.56.103]

TASK [Mariadb - Restarting/Enabling] *****
changed: [192.168.56.103]
changed: [192.168.56.104]

PLAY [file_servers] *****
TASK [Gathering Facts] *****
ok: [192.168.56.103]

TASK [install samba package] *****
ok: [192.168.56.103]

PLAY RECAP *****
192.168.56.102      : ok=4    changed=0    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0
192.168.56.103      : ok=7    changed=1    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.56.104      : ok=8    changed=1    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0

```

But if it is run normally, the MariaDB service will be both started and enabled on the target servers.

Reflections:

Answer the following:

- What is the importance of putting our remote servers into groups?
 - To organize and manage them more effectively. Grouping servers allows us to apply configurations and tasks to specific sets of servers based on their roles or characteristics. It simplifies playbook management and makes it easier to perform actions on a subset of servers, improving overall organization and maintainability.
- What is the importance of tags in playbooks?
 - Tags in playbooks provide a way to selectively run specific tasks within a playbook. It allows us to choose which tasks to execute based on user-defined tags associated with each task. This is valuable because it offers fine-grained control over playbook execution, helping to save time and resources. Tags are especially useful when you want to rerun only certain parts of a playbook or when you want to skip specific tasks altogether.
- Why do think some services need to be managed automatically in playbooks?
 - Some services need to be managed automatically in playbooks to ensure consistent, reliable, and reproducible server configurations. Automation helps eliminate human errors and ensures that services are set up correctly and consistently across multiple servers. It also allows for easy scaling and provisioning of resources, making it more efficient to manage infrastructure. Additionally, automation enables version control and documentation of

configurations, making it easier to maintain and troubleshoot server environments over time.

Conclusion:

Throughout the activity, I have learned the importance of organizing remote servers into groups to streamline management and apply configurations more efficiently. Tags in playbooks were introduced as a way to selectively execute tasks, offering fine-grained control over playbook execution. Automatic management of services through playbooks ensures consistent and reliable server configurations, allowing for scalability and simplified maintenance. Overall, this activity emphasized the significance of automation and organization in effective server management.

Additional info: Some of the errors I encountered are:

TASK [install updates (CentOS)]:

fatal: [192.168.56.104]: FAILED! => {"ansible_facts": {"pkg_mgr": "yum"}, "changed": false, "msg": ["Could not detect which major revision of dnf is in use, which is required to determine module backend.", "You should manually specify use_backend to tell the module whether to use the dnf4 or dnf5 backend"]}].

- I just edited the site.yml into using the yum package instead of dnf package since it is not working although the CentOS 7 machine already has installed dnf in it. Before doing this, I tried repeating the process of the installation of dnf and it shows that it has been installed already. I also tried to include a specific dnf in the playbook but it still has not been recognized. Also, I updated the ansible for both manage node and control node (CentOS).

TASK [install apache and php for CentOS servers]:

fatal: [192.168.56.104]: FAILED! => {"ansible_facts": {"pkg_mgr": "yum"}, "changed": false, "msg": ["Could not detect which major revision of dnf is in use, which is required to determine module backend.", "You should manually specify use_backend to tell the module whether to use the dnf4 or dnf5 backend"]}].

- Here, I modified the site.yml by adding `ansible_distribution_major_version == "7"` after `when: ansible_distribution == "CentOS"` in the portion where there's a need to install the apache and php for CentOS servers; and by also using yum as a package instead of dnf.
- just changing the dnf into yum in the latest script

TASK [install updates (Ubuntu)]:

```
fatal: [192.168.56.103]: FAILED! => {"changed": false, "msg": "'/usr/bin/apt-get dist-upgrade ' failed: E: dpkg was interrupted, you must manually run 'sudo dpkg --configure -a' to correct the problem. \n", "rc": 100, "stdout": "", "stdout_lines": []}
```

- login to host, then: `sudo dpkg --configure -a`
- `sudo apt-get install -f`

TASK [install updates (Ubuntu)]

```
fatal: [192.168.56.102]: FAILED! => {"changed": false, "msg": "'/usr/bin/apt-get dist-upgrade ' failed: E: Could not get lock /var/lib/dpkg/lock-frontent. It is held by process 2285 (apt-get)\nE: Unable to acquire the dpkg frontend lock (/var/lib/dpkg/lock-frontent), is another process using it?\n", "rc": 100, "stdout": "", "stdout_lines": []}
```

- rerun the playbook