

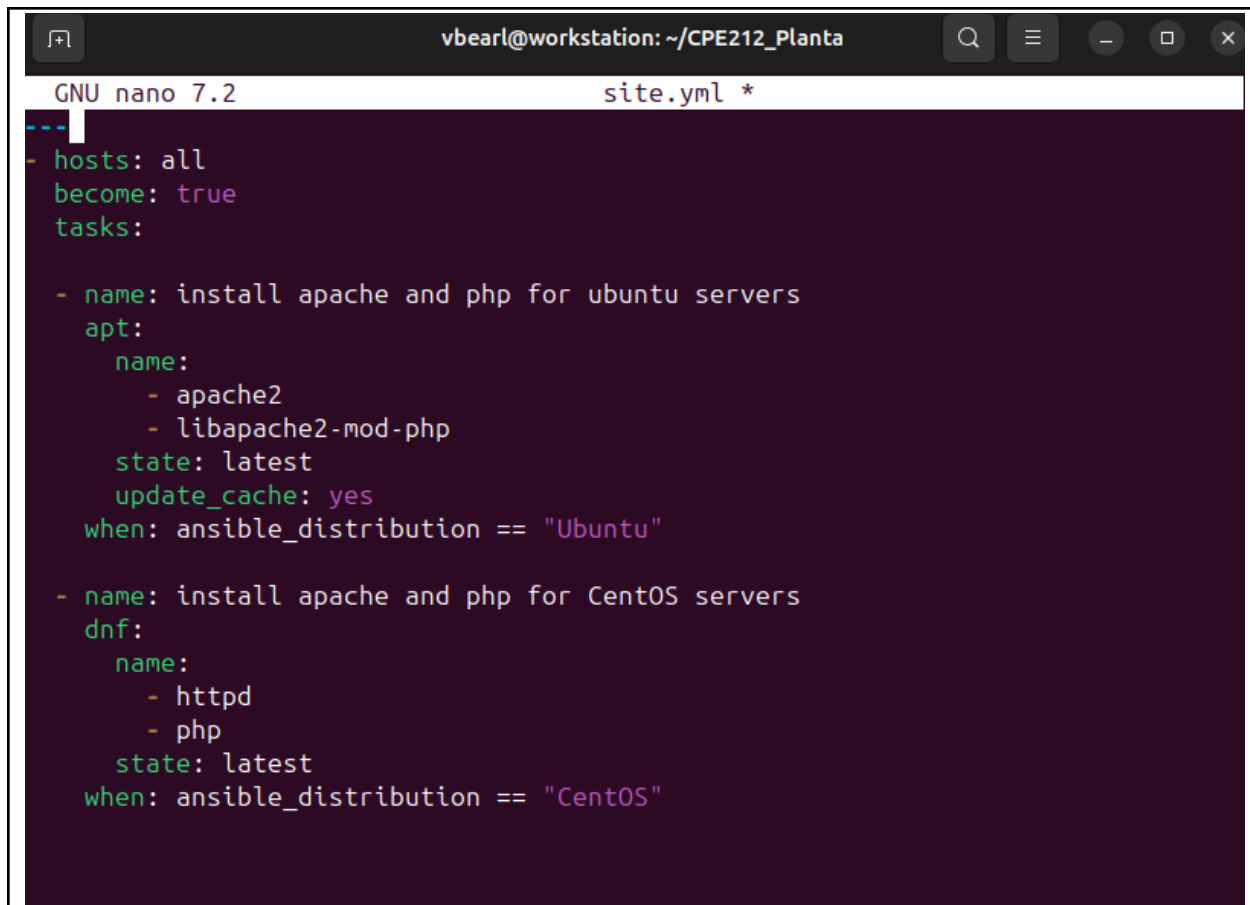
Name: Planta, Calvin Earl L.	Date Performed: 9/12/25
Course/Section: CPE 212 - CPE31S2	Date Submitted: 9/12/25
Instructor: Engr. Robin Valenzuela	Semester and SY: 1st Sem SY 2025-2026
Activity 6: Targeting Specific Nodes and Managing Services	
1. Objectives: 1.1 Individualize hosts 1.2 Apply tags in selecting plays to run 1.3 Managing Services from remote servers using playbooks	
2. Discussion: <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 the playbook. For example, when we install web servers or httpd for CentOS, we notice that the service did not start automatically.</p> <p>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 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>	

```
vbearl@workstation: ~  
vbearl@workstation:~$ ssh-copy-id -i ~/.ssh/id_rsa vbearl@192.168.56.129  
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/vbearl/.ssh/id_rsa.pub"  
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed  
  
/usr/bin/ssh-copy-id: WARNING: All keys were skipped because they already exist on the remote system.  
                (if you think this is a mistake, you may want to use -f option)  
vbearl@workstation:~$
```

Task 1: Targeting Specific Nodes

1. Create a new playbook and name 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"
```

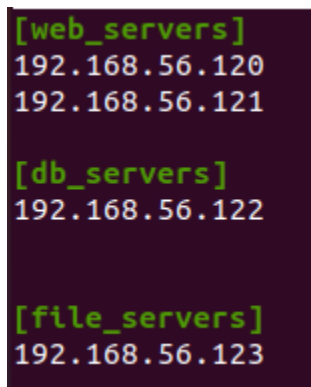


```
GNU nano 7.2 site.yml *
---
- hosts: all
  become: true
  tasks:

    - 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.

```
[web_servers]
192.168.56.106
192.168.56.107
192.168.56.117

[db_servers]
192.168.56.108

[file_servers]
192.168.56.117
```

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.

```
vbearl@workstation: ~/CPE212_Planta
GNU nano 7.2 site.yml *
---
- 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"
```

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.

```
vbearl@workstation:~/CPE212_Planta$ ansible-playbook --ask-become-pass site.yml
BECOME password:

PLAY [all] *****

TASK [Gathering Facts] *****
ok: [192.168.56.117]
ok: [192.168.56.108]
ok: [192.168.56.106]
ok: [192.168.56.107]

TASK [install updates (CentOS)] *****
skipping: [192.168.56.106]
skipping: [192.168.56.107]
skipping: [192.168.56.108]
ok: [192.168.56.117]

TASK [install updates (Ubuntu)] *****
skipping: [192.168.56.117]
changed: [192.168.56.108]
changed: [192.168.56.106]
changed: [192.168.56.107]

PLAY [web_servers] *****

TASK [Gathering Facts] *****
ok: [192.168.56.117]
ok: [192.168.56.107]
ok: [192.168.56.106]

TASK [install apache and php for Ubuntu servers] *****
skipping: [192.168.56.117]
changed: [192.168.56.106]
changed: [192.168.56.107]

TASK [install apache and php for CentOS servers] *****
skipping: [192.168.56.106]
skipping: [192.168.56.107]
ok: [192.168.56.117]

PLAY RECAP *****
192.168.56.106      : ok=4    changed=2    unreachable=0    failed=0    s
kipped=2    rescued=0    ignored=0
192.168.56.107      : ok=4    changed=2    unreachable=0    failed=0    s
kipped=2    rescued=0    ignored=0
192.168.56.108      : ok=2    changed=1    unreachable=0    failed=0    s
kipped=1    rescued=0    ignored=0
192.168.56.117      : ok=4    changed=0    unreachable=0    failed=0    s
kipped=2    rescued=0    ignored=0
```

- The playbook first performs the written pre tasks, which installs updates and upgrades for both distributions. The next task targets hosts within the `web_servers` group, which installs their own versions apache and php for each distribution.
4. Let's try to edit the `site.yml` file again. 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: 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"

```

[^]G Help [^]O Write Out [^]W Where Is [^]K Cut [^]T Execute [^]C Location
[^]X Exit [^]R Read File [^]\ Replace [^]U Paste [^]J Justify [^]/ Go To Line

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

```

PLAY [db_servers] *****

TASK [Gathering Facts] *****
ok: [192.168.56.108]

TASK [install mariadb package (CentOS)] *****
skipping: [192.168.56.108]

TASK [mariadb- Restarting/Enabling] *****
changed: [192.168.56.108]

TASK [install mariadb package (Ubuntu)] *****
ok: [192.168.56.108]

PLAY RECAP *****
192.168.56.106      : ok=4    changed=0    unreachable=0    failed=0    s
kipped=2    rescued=0    ignored=0
192.168.56.107      : ok=4    changed=0    unreachable=0    failed=0    s
kipped=2    rescued=0    ignored=0
192.168.56.108      : ok=5    changed=1    unreachable=0    failed=0    s
kipped=2    rescued=0    ignored=0
192.168.56.117      : ok=4    changed=0    unreachable=0    failed=0    s
kipped=2    rescued=0    ignored=0

```

- This playbook revision runs tasks targeted to hosts within the db_servers group. It installs the mariadb package in hosts that use CentOS and Ubuntu distribution and enables the service.
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.

Server 1

```
vbearl@server1:~$ sudo systemctl status mariadb
[sudo] password for vbearl:
● mariadb.service - MariaDB 10.11.13 database server
   Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; preset: ▶
   Active: active (running) since Fri 2025-09-12 08:44:50 UTC; 2h 31min ago
     Docs: man:mariadb(8)
           https://mariadb.com/kb/en/library/systemd/
  Main PID: 1318 (mariabdd)
    Status: "Taking your SQL requests now..."
     Tasks: 11 (limit: 30383)
    Memory: 107.9M (peak: 113.2M)
       CPU: 1.302s
    CGroup: /system.slice/mariadb.service
            └─1318 /usr/sbin/mariabdd

Sep 12 08:44:50 server1 mariabdd[1318]: 2025-09-12  8:44:50 0 [Note] Plugin 'FE>
Sep 12 08:44:50 server1 mariabdd[1318]: 2025-09-12  8:44:50 0 [Note] InnoDB: Lo>
Sep 12 08:44:50 server1 mariabdd[1318]: 2025-09-12  8:44:50 0 [Note] InnoDB: Bu>
Sep 12 08:44:50 server1 mariabdd[1318]: 2025-09-12  8:44:50 0 [Warning] You nee>
Sep 12 08:44:50 server1 mariabdd[1318]: 2025-09-12  8:44:50 0 [Note] Server soc>
```

Server 2

```

vbearl@server2:~$ sudo systemctl status mariadb
[sudo] password for vbearl:
● mariadb.service - MariaDB 10.11.13 database server
   Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; preset: >
   Active: active (running) since Fri 2025-09-12 09:14:37 UTC; 2h 2min ago
     Docs: man:mariadb(8)
           https://mariadb.com/kb/en/library/systemd/
   Main PID: 1261 (mariabdd)
    Status: "Taking your SQL requests now..."
     Tasks: 10 (limit: 30383)
    Memory: 108.0M (peak: 113.4M)
       CPU: 1.751s
    CGroup: /system.slice/mariadb.service
            └─1261 /usr/sbin/mariabdd

Sep 12 09:14:37 server2 mariabdd[1261]: 2025-09-12  9:14:37 0 [Note] InnoDB: Lo>
Sep 12 09:14:37 server2 mariabdd[1261]: 2025-09-12  9:14:37 0 [Warning] You nee>
Sep 12 09:14:37 server2 mariabdd[1261]: 2025-09-12  9:14:37 0 [Note] InnoDB: Bu>
Sep 12 09:14:37 server2 mariabdd[1261]: 2025-09-12  9:14:37 0 [Note] Server soc>
Sep 12 09:14:37 server2 mariabdd[1261]: 2025-09-12  9:14:37 0 [Note] /usr/sbin/>
Sep 12 09:14:37 server2 mariabdd[1261]: Version: '10.11.13-MariaDB-0ubuntu0.24.>
Sep 12 09:14:37 server2 systemd[1]: Started mariadb.service - MariaDB 10.11.13 >
Sep 12 09:14:37 server2 /etc/mysql/debian-start[1447]: Upgrading MariaDB tables>
Sep 12 09:14:37 server2 /etc/mysql/debian-start[1463]: Checking for insecure ro>

```

Workstation

```

vbearl@workstation:~/CPE212_Planta$ sudo systemctl status mariadb
● mariadb.service - MariaDB 10.11.13 database server
   Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; preset: >
   Active: active (running) since Fri 2025-09-12 11:15:17 UTC; 1min 49s ago
     Docs: man:mariadb(8)
           https://mariadb.com/kb/en/library/systemd/
   Process: 16090 ExecStartPre=/usr/bin/install -m 755 -o mysql -g root -d /va>
   Process: 16092 ExecStartPre=/bin/sh -c systemctl unset-environment _WSREP_S>
   Process: 16095 ExecStartPre=/bin/sh -c [ ! -e /usr/bin/galera_recovery ] &&>
   Process: 16167 ExecStartPost=/bin/sh -c systemctl unset-environment _WSREP_>
   Process: 16169 ExecStartPost=/etc/mysql/debian-start (code=exited, status=0>
   Main PID: 16155 (mariabdd)
    Status: "Taking your SQL requests now..."
     Tasks: 10 (limit: 30383)
    Memory: 79.3M (peak: 83.1M)
       CPU: 787ms
    CGroup: /system.slice/mariadb.service
            └─16155 /usr/sbin/mariabdd

Sep 12 11:15:17 workstation mariabdd[16155]: 2025-09-12 11:15:17 0 [Note] InnoD>
Sep 12 11:15:17 workstation mariabdd[16155]: 2025-09-12 11:15:17 0 [Note] InnoD>
Sep 12 11:15:17 workstation mariabdd[16155]: 2025-09-12 11:15:17 0 [Note] Plugi>
Sep 12 11:15:17 workstation mariabdd[16155]: 2025-09-12 11:15:17 0 [Note] InnoD>
Sep 12 11:15:17 workstation mariabdd[16155]: 2025-09-12 11:15:17 0 [Warning] Yo>

```

CentOS

```
[bearl@server3 ~]$ systemctl status mariadb
● mariadb.service - MariaDB 10.5 database server
   Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; preset: >
   Active: active (running) since Thu 2025-09-18 22:33:37 PST; 1min 47s ago
     Docs: man:mariadbd(8)
           https://mariadb.com/kb/en/library/systemd/
   Process: 74770 ExecStartPre=/usr/libexec/mariadb-check-socket (code=exited,>
   Process: 74792 ExecStartPre=/usr/libexec/mariadb-prepare-db-dir mariadb.ser>
   Process: 74903 ExecStartPost=/usr/libexec/mariadb-check-upgrade (code=exite>
  Main PID: 74881 (mariadbd)
    Status: "Taking your SQL requests now..."
     Tasks: 8 (limit: 22967)
    Memory: 71.4M (peak: 104.5M)
       CPU: 367ms
    CGroup: /system.slice/mariadb.service
            └─74881 /usr/libexec/mariadbd --basedir=/usr
```

Describe the output.

- In the Ubuntu nodes, the service is already active. However, the service is not yet started for CentOS at first, which is why we added a task that dynamically enables the service.
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
```

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

```
vbearl@workstation:~/CPE212_Planta$ ansible-playbook --ask-become-pass site.yml  
BECOME password:
```

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

```
TASK [Gathering Facts] *****
```

```
ok: [192.168.56.106]
```

```
ok: [192.168.56.105]
```

```
ok: [192.168.56.107]
```

```
ok: [192.168.56.108]
```

```
TASK [install updates (CentOS)] *****
```

```
skipping: [192.168.56.106]
```

```
skipping: [192.168.56.107]
```

```
skipping: [192.168.56.105]
```

```
ok: [192.168.56.108]
```

```
TASK [install updates (Ubuntu)] *****
```

```
skipping: [192.168.56.108]
```

```
ok: [192.168.56.106]
```

```
ok: [192.168.56.105]
```

```
ok: [192.168.56.107]
```

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

```
TASK [Gathering Facts] *****
```

```
ok: [192.168.56.106]
```

```
ok: [192.168.56.108]
```

```
ok: [192.168.56.107]
```

```
TASK [install apache and php for Ubuntu servers] *****
```

```
skipping: [192.168.56.108]
```

```
ok: [192.168.56.106]
```

```
changed: [192.168.56.107]
```

```
TASK [intall apache and php for CentOS servers] *****
```

```
skipping: [192.168.56.106]
```

```
skipping: [192.168.56.107]
```

```
ok: [192.168.56.108]
```

```
PLAY [db_servers] *****
```

```
TASK [Gathering Facts] *****
```

```
ok: [192.168.56.105]
```

```
ok: [192.168.56.108]
```

```
TASK [install mariadb package (CentOS)] *****
```

```

TASK [install mariadb package (CentOS)] *****
skipping: [192.168.56.105]
ok: [192.168.56.108]

TASK [Mariadb Restarting/Enabling] *****
changed: [192.168.56.108]
changed: [192.168.56.105]

TASK [install mariadb package (Ubuntu)] *****
skipping: [192.168.56.108]
ok: [192.168.56.105]

PLAY [file_servers] *****

TASK [Gathering Facts] *****
ok: [192.168.56.108]

TASK [install samba package] *****
ok: [192.168.56.108]

```

```

PLAY RECAP *****
192.168.56.105      : ok=5    changed=1    unreachable=0    failed=0    s
kipped=2    rescued=0    ignored=0
192.168.56.106      : ok=4    changed=0    unreachable=0    failed=0    s
kipped=2    rescued=0    ignored=0
192.168.56.107      : ok=4    changed=1    unreachable=0    failed=0    s
kipped=2    rescued=0    ignored=0
192.168.56.108      : ok=9    changed=1    unreachable=0    failed=0    s
kipped=3    rescued=0    ignored=0

```

- This playbook revision adds new tasks that targets the hosts within the `file_servers` group. The task installs the samba package.

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.

GNU nano 7.2

site.yml *

```
---
- 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"
```

GNU nano 7.2

site.yml *

```
- 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
```

```
GNU nano 7.2                                site.yml *
- 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: Mariadb Restarting/Enabling
    service:
      name: mariadb
      state: restarted
      enabled: true
```

```
GNU nano 7.2                                site.yml *
      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
```

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

```
vbearl@workstation:~/CPE212_Planta$ ansible-playbook --ask-become-pass site.yml  
BECOME password:
```

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

```
TASK [Gathering Facts] *****
```

```
ok: [192.168.56.106]
```

```
ok: [192.168.56.105]
```

```
ok: [192.168.56.107]
```

```
ok: [192.168.56.108]
```

```
TASK [install updates (CentOS)] *****
```

```
skipping: [192.168.56.106]
```

```
skipping: [192.168.56.107]
```

```
skipping: [192.168.56.105]
```

```
ok: [192.168.56.108]
```

```
TASK [install updates (Ubuntu)] *****
```

```
skipping: [192.168.56.108]
```

```
ok: [192.168.56.106]
```

```
ok: [192.168.56.105]
```

```
ok: [192.168.56.107]
```

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

```
TASK [Gathering Facts] *****
```

```
ok: [192.168.56.106]
```

```
ok: [192.168.56.108]
```

```
ok: [192.168.56.107]
```

```
TASK [install apache and php for Ubuntu servers] *****
```

```
skipping: [192.168.56.108]
```

```
ok: [192.168.56.106]
```

```
changed: [192.168.56.107]
```

```
TASK [intall apache and php for CentOS servers] *****
```

```
skipping: [192.168.56.106]
```

```
skipping: [192.168.56.107]
```

```
ok: [192.168.56.108]
```

```
PLAY [db_servers] *****
```

```
TASK [Gathering Facts] *****
```

```
ok: [192.168.56.105]
```

```
ok: [192.168.56.108]
```

```
TASK [install mariadb package (CentOS)] *****
```

```

TASK [install mariadb package (CentOS)] *****
skipping: [192.168.56.105]
ok: [192.168.56.108]

TASK [Mariadb Restarting/Enabling] *****
changed: [192.168.56.108]
changed: [192.168.56.105]

TASK [install mariadb package (Ubuntu)] *****
skipping: [192.168.56.108]
ok: [192.168.56.105]

PLAY [file_servers] *****

TASK [Gathering Facts] *****
ok: [192.168.56.108]

TASK [install samba package] *****
ok: [192.168.56.108]

```

```

PLAY RECAP *****
192.168.56.105      : ok=5    changed=1    unreachable=0    failed=0    s
kipped=2    rescued=0    ignored=0
192.168.56.106      : ok=4    changed=0    unreachable=0    failed=0    s
kipped=2    rescued=0    ignored=0
192.168.56.107      : ok=4    changed=1    unreachable=0    failed=0    s
kipped=2    rescued=0    ignored=0
192.168.56.108      : ok=9    changed=1    unreachable=0    failed=0    s
kipped=3    rescued=0    ignored=0

```

- This playbook revision does the same thing and produces similar output as the previous ones. We only added tags for each play, which we will use in order to demonstrate its functionality later on.

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

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

```

vbearl@workstation:~/CPE212_Planta$ 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]

```

- This command lists all the included tags for each play.

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

```

vbearl@workstation:~/CPE212_Planta$ ansible-playbook --tags centos --ask-become-
pass site.yml
BECOME password:

PLAY [all] *****

TASK [Gathering Facts] *****
ok: [192.168.56.106]
ok: [192.168.56.108]
ok: [192.168.56.105]
ok: [192.168.56.107]

TASK [install updates (CentOS)] *****
skipping: [192.168.56.106]
skipping: [192.168.56.107]
skipping: [192.168.56.105]
ok: [192.168.56.108]

TASK [install updates (Ubuntu)] *****
skipping: [192.168.56.108]
ok: [192.168.56.106]
ok: [192.168.56.105]
ok: [192.168.56.107]

```

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

TASK [Gathering Facts] *****
ok: [192.168.56.106]
ok: [192.168.56.107]
ok: [192.168.56.108]

TASK [install apache and php for CentOS servers] *****
skipping: [192.168.56.106]
skipping: [192.168.56.107]
ok: [192.168.56.108]

PLAY [db_servers] *****

TASK [Gathering Facts] *****
ok: [192.168.56.105]
ok: [192.168.56.108]

TASK [install mariadb package (CentOS)] *****
skipping: [192.168.56.105]
ok: [192.168.56.108]
```

```
PLAY [file_servers] *****

TASK [Gathering Facts] *****
ok: [192.168.56.108]

PLAY RECAP *****
192.168.56.105      : ok=3    changed=0    unreachable=0    failed=0    s
kipped=2    rescued=0    ignored=0
192.168.56.106      : ok=3    changed=0    unreachable=0    failed=0    s
kipped=2    rescued=0    ignored=0
192.168.56.107      : ok=3    changed=0    unreachable=0    failed=0    s
kipped=2    rescued=0    ignored=0
192.168.56.108      : ok=7    changed=0    unreachable=0    failed=0    s
kipped=1    rescued=0    ignored=0
```

- This command executes all plays with the tags “centos” included in it. The rest of the plays without the matching tags are excluded and will not be run.

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

```
vbearl@workstation:~/CPE212_Planta$ ansible-playbook --tags db --ask-become-pass
site.yml
BECOME password:
```

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

```
TASK [Gathering Facts] *****
```

```
ok: [192.168.56.106]
```

```
ok: [192.168.56.105]
```

```
ok: [192.168.56.108]
```

```
ok: [192.168.56.107]
```

```
TASK [install updates (CentOS)] *****
```

```
skipping: [192.168.56.106]
```

```
skipping: [192.168.56.107]
```

```
skipping: [192.168.56.105]
```

```
ok: [192.168.56.108]
```

```
TASK [install updates (Ubuntu)] *****
```

```
skipping: [192.168.56.108]
```

```
ok: [192.168.56.106]
```

```
ok: [192.168.56.105]
```

```
ok: [192.168.56.107]
```

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

```
TASK [Gathering Facts] *****
```

```
ok: [192.168.56.106]
```

```
ok: [192.168.56.108]
```

```
ok: [192.168.56.107]
```

```
PLAY [db_servers] *****
```

```
TASK [Gathering Facts] *****
```

```
ok: [192.168.56.105]
```

```
ok: [192.168.56.108]
```

```
TASK [install mariadb package (CentOS)] *****
```

```
skipping: [192.168.56.105]
```

```
ok: [192.168.56.108]
```

```
TASK [install mariadb package (Ubuntu)] *****
```

```
skipping: [192.168.56.108]
```

```
ok: [192.168.56.105]
```

```
PLAY [file_servers] *****

TASK [Gathering Facts] *****
ok: [192.168.56.108]

PLAY RECAP *****
192.168.56.105      : ok=4    changed=0    unreachable=0    failed=0    s
kipped=2    rescued=0    ignored=0
192.168.56.106      : ok=3    changed=0    unreachable=0    failed=0    s
kipped=1    rescued=0    ignored=0
192.168.56.107      : ok=3    changed=0    unreachable=0    failed=0    s
kipped=1    rescued=0    ignored=0
192.168.56.108      : ok=6    changed=0    unreachable=0    failed=0    s
kipped=2    rescued=0    ignored=0
```

- This command executes all plays with the tags “db” included in it. The rest of the plays without the matching tags are excluded and will not be run.

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

```
vbearl@workstation:~/CPE212_Planta$ ansible-playbook --tags apache --ask-become-
pass site.yml
BECOME password:

PLAY [all] *****

TASK [Gathering Facts] *****
ok: [192.168.56.105]
ok: [192.168.56.106]
ok: [192.168.56.108]
ok: [192.168.56.107]

TASK [install updates (CentOS)] *****
skipping: [192.168.56.106]
skipping: [192.168.56.107]
skipping: [192.168.56.105]
ok: [192.168.56.108]

TASK [install updates (Ubuntu)] *****
skipping: [192.168.56.108]
ok: [192.168.56.106]
ok: [192.168.56.105]
ok: [192.168.56.107]
```



```

PLAY [web_servers] *****

TASK [Gathering Facts] *****
ok: [192.168.56.106]
ok: [192.168.56.108]
ok: [192.168.56.107]

TASK [install apache and php for Ubuntu servers] *****
skipping: [192.168.56.108]
ok: [192.168.56.106]
ok: [192.168.56.107]

TASK [intall apache and php for CentOS servers] *****
skipping: [192.168.56.106]
skipping: [192.168.56.107]
ok: [192.168.56.108]

PLAY [db_servers] *****

TASK [Gathering Facts] *****
ok: [192.168.56.105]
ok: [192.168.56.108]

PLAY [file_servers] *****

TASK [Gathering Facts] *****
ok: [192.168.56.108]

PLAY RECAP *****
192.168.56.105      : ok=3    changed=0    unreachable=0    failed=0    s
kipped=1    rescued=0    ignored=0
192.168.56.106      : ok=4    changed=0    unreachable=0    failed=0    s
kipped=2    rescued=0    ignored=0
192.168.56.107      : ok=4    changed=0    unreachable=0    failed=0    s
kipped=2    rescued=0    ignored=0
192.168.56.108      : ok=6    changed=0    unreachable=0    failed=0    s
kipped=2    rescued=0    ignored=0

```

- This command executes all plays with the tags “apache” included in it. The rest of the plays without the matching tags are excluded and will not be run.

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

```
vbearl@workstation:~/CPE212_Planta$ ansible-playbook --tags "apache,db" --ask-become-pass site.yml  
BECOME password:
```

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

```
TASK [Gathering Facts] *****
```

```
ok: [192.168.56.106]
```

```
ok: [192.168.56.108]
```

```
ok: [192.168.56.107]
```

```
ok: [192.168.56.105]
```

```
TASK [install updates (CentOS)] *****
```

```
skipping: [192.168.56.106]
```

```
skipping: [192.168.56.107]
```

```
skipping: [192.168.56.105]
```

```
ok: [192.168.56.108]
```

```
TASK [install updates (Ubuntu)] *****
```

```
skipping: [192.168.56.108]
```

```
ok: [192.168.56.105]
```

```
ok: [192.168.56.106]
```

```
ok: [192.168.56.107]
```

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

```
TASK [Gathering Facts] *****
```

```
ok: [192.168.56.106]
```

```
ok: [192.168.56.108]
```

```
ok: [192.168.56.107]
```

```
TASK [install apache and php for Ubuntu servers] *****
```

```
skipping: [192.168.56.108]
```

```
ok: [192.168.56.106]
```

```
ok: [192.168.56.107]
```

```
TASK [install apache and php for CentOS servers] *****
```

```
skipping: [192.168.56.106]
```

```
skipping: [192.168.56.107]
```

```
ok: [192.168.56.108]
```

```
PLAY [db_servers] *****
```

```
TASK [Gathering Facts] *****
```

```
ok: [192.168.56.105]
```

```
ok: [192.168.56.108]
```

```

TASK [install mariadb package (CentOS)] *****
skipping: [192.168.56.105]
ok: [192.168.56.108]

TASK [install mariadb package (Ubuntu)] *****
skipping: [192.168.56.108]
ok: [192.168.56.105]

PLAY [file_servers] *****

TASK [Gathering Facts] *****
ok: [192.168.56.108]

PLAY RECAP *****
192.168.56.105      : ok=4    changed=0    unreachable=0    failed=0    s
kipped=2    rescued=0    ignored=0
192.168.56.106      : ok=4    changed=0    unreachable=0    failed=0    s
kipped=2    rescued=0    ignored=0
192.168.56.107      : ok=4    changed=0    unreachable=0    failed=0    s
kipped=2    rescued=0    ignored=0
192.168.56.108      : ok=7    changed=0    unreachable=0    failed=0    s

```

- This command executes all plays with the tags “apache” and “db” included in it. The rest of the plays without the matching tags are excluded and will not be run.

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"
```

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

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

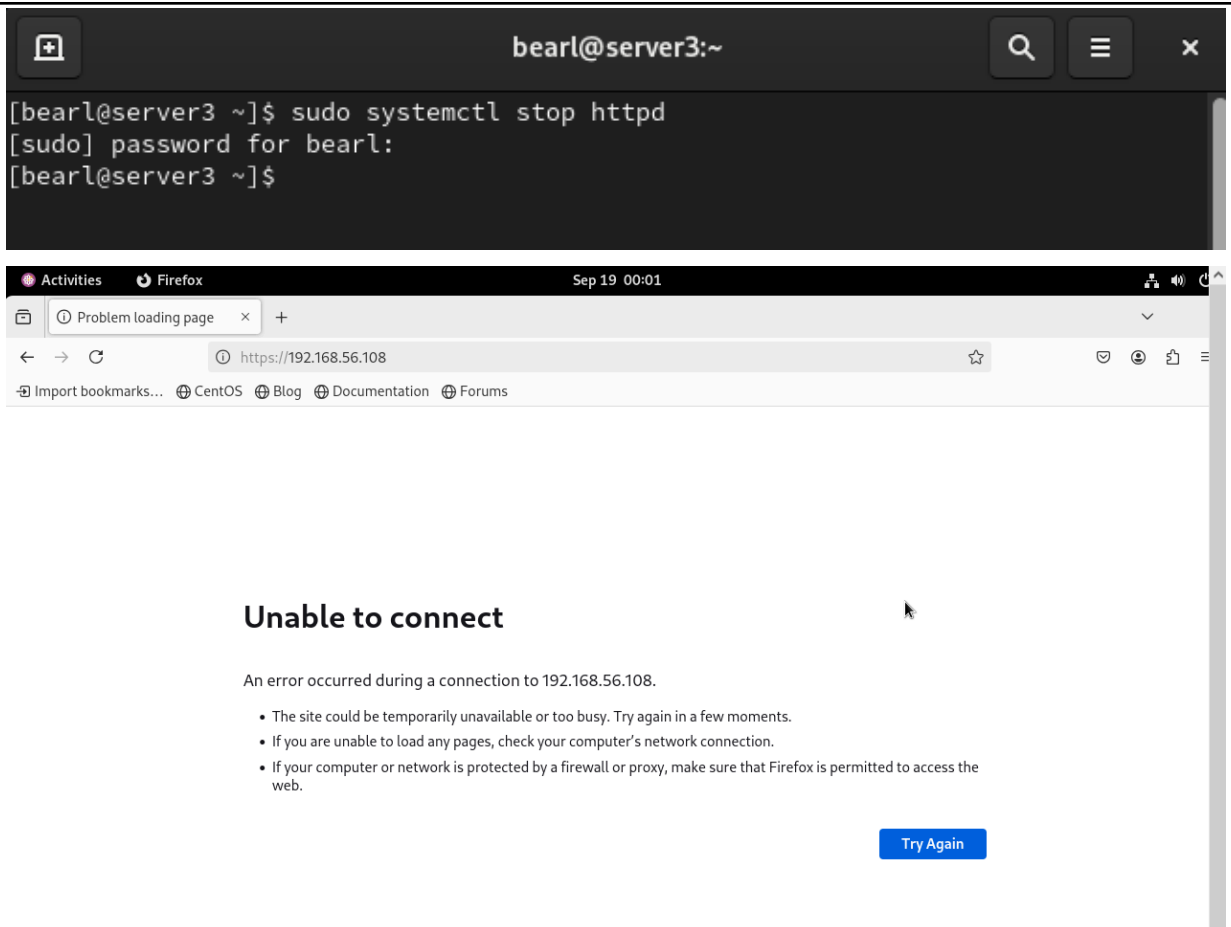
    - name: install mariadb package (CentOS)
      tags: centos, db,mariadb
      dnf:
        name: mariadb-server
        state: latest
      when: ansible_distribution == "CentOS"

    - 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.

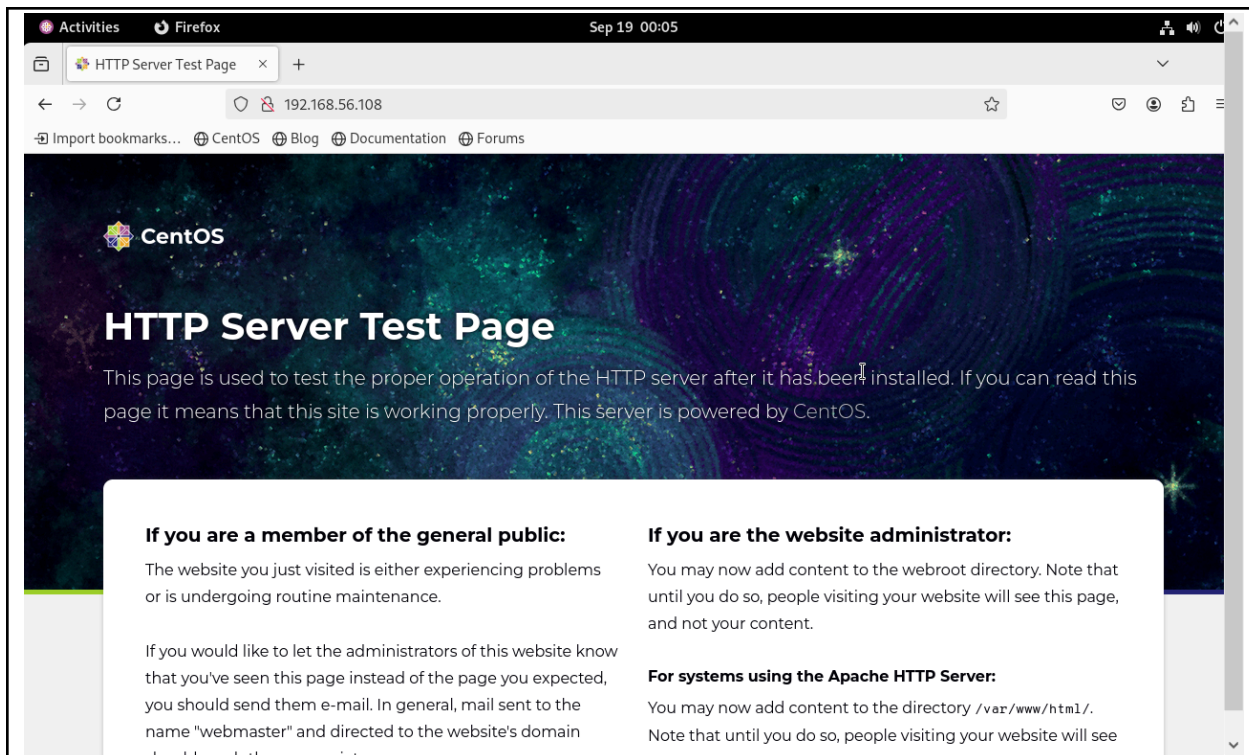
2. To test it, before you run the saved playbook, go to the CentOS server and stop the currently running httpd using the command *sudo systemctl stop httpd*. When prompted, enter the sudo password. After that, open the browser and enter the CentOS server's IP address. You should not be getting a display because we stopped the httpd service already.



- 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.

```
TASK [start httpd (CentOS)] *****
changed: [192.168.56.108]

PLAY RECAP *****
192.168.56.105      : ok=3    changed=0    unreachable=0    failed=0    s
kipped=1    rescued=0    ignored=0
192.168.56.106      : ok=3    changed=0    unreachable=0    failed=0    s
kipped=2    rescued=0    ignored=0
192.168.56.107      : ok=3    changed=0    unreachable=0    failed=0    s
kipped=2    rescued=0    ignored=0
192.168.56.108      : ok=7    changed=1    unreachable=0    failed=0    s
kipped=1    rescued=0    ignored=0
```



- After adding the given tasks into the playbook and inputting my node's IP address into the browser search bar, it redirected me to httpd home page, confirming that the play has successfully enabled the said service.

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"
```

Reflections:

Answer the following:

1. What is the importance of putting our remote servers into groups?
 - Partitioning remote servers into groups in our inventory file can be useful so that we can specify only the nodes that we want the playbook to run to. We can group nodes according to whatever we wish to, whether by distribution or by node type (control node/ managed node) so that for instance, if we wish to

update all servers, we can create a playbook and target only the servers by specifying their group name as the target hosts.

2. What is the importance of tags in playbooks?

- Adding tasks in a playbook gives a metadata to a task within the playbook. This is useful if we wish to run only specific tasks listed in the playbook. If we wish to run a certain task, we can simply call the tags assigned under that specific task so that the playbook will play that specific task only, rather than having to go through all the tasks which is generally time consuming.

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

- Some services should be automated in playbooks so that we can reduce workload and be able to manage our server nodes more efficiently. Doing so would also compensate for the fact that CentOS does not automatically start a service upon being installed, whereas creating a playbook to automate service management solves this problem effortlessly.