Name: JOZETTE FERMIN	**Date Performed: OCTOBER 9, 2023**
Course/Section: CPE31S4	**Date Submitted: OCTOBER 9, 2023**
Instructor: ENGR. JONATHAN TAYLAR	**Semester and SY: 1st sem 2023-2024**

Activity 7: Managing Files and Creating Roles in Ansible

1. **Objectives:**

1.1 Manage files in remote servers

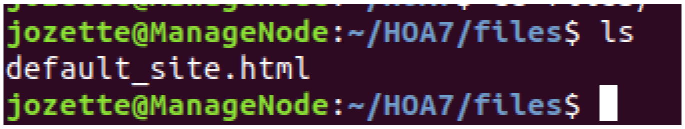
1.2 Implement roles in ansible

2. **Discussion**:

In this activity, we look at the concept of copying a file to a server. We are going to create a file into our git repository and use Ansible to grab that file and put it into a particular place so that we could do things like customize a default website, or maybe install a default configuration file. We will also implement roles to consolidate plays.

Task 1: Create a file and copy it to remote servers

1. Using the previous directory we created, create a directory, and named it "*files*." Create a file inside that directory and name it "*default_site.html*." Edit the file and put basic HTML syntax. Any content will do, as long as it will display text later. Save the file and exit.



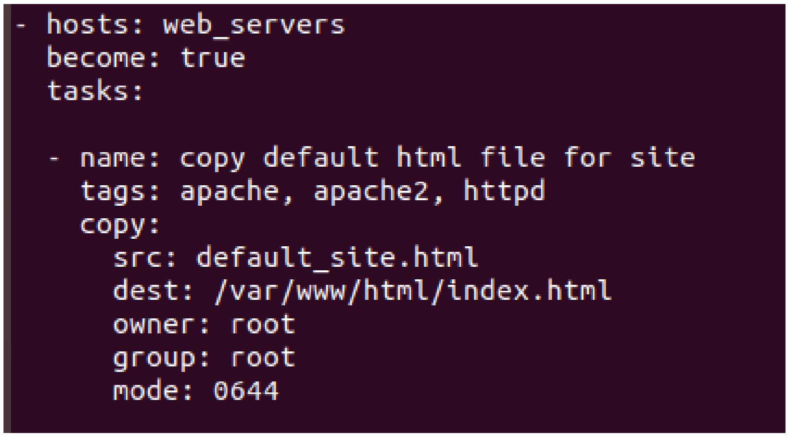
```
jozette@ManageNode:~/HOA7/files$ ls
default_site.html
jozette@ManageNode:~/HOA7/files$ 
```

2. Edit the *site.yml* file and just below the *web_servers* play, create a new file to copy the default html file for site:

```
-    name: copy default html file for site

     tags: apache, apache2, httpd
     copy:
        src: default_site.html
        dest: /var/www/html/index.html
        owner: root
        group: root
        mode: 0644
```

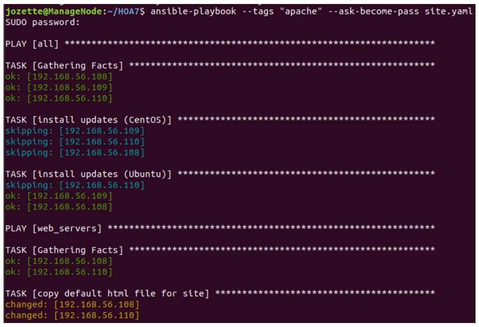
3. Run the playbook *site.yml*. Describe the changes.

Input

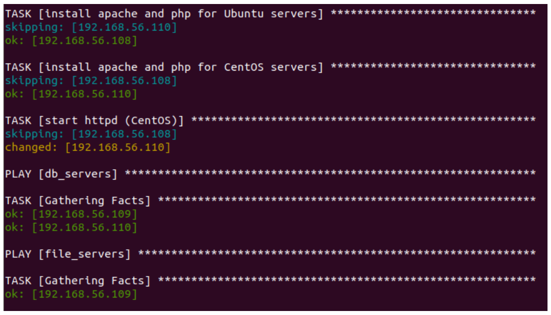


```
- hosts: web_servers
  become: true
  tasks:

  - name: copy default html file for site
    tags: apache, apache2, httpd
    copy:
      src: default_site.html
      dest: /var/www/html/index.html
      owner: root
      group: root
      mode: 0644
```

Process

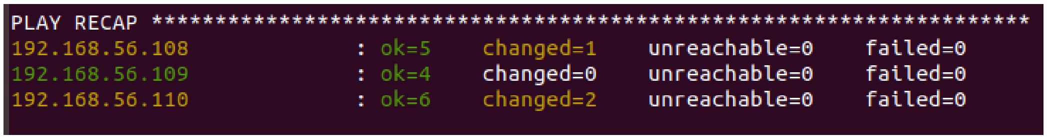


```
jozette@ManageNode:~/HOA7$ ansible-playbook --tags "apache" --ask-become-pass site.yaml
SUDO password:

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

TASK [Gathering Facts] *******************************************************
ok: [192.168.56.108]
ok: [192.168.56.109]
ok: [192.168.56.110]

TASK [install updates (CentOS)] **********************************************
skipping: [192.168.56.109]
skipping: [192.168.56.110]
skipping: [192.168.56.108]

TASK [install updates (Ubuntu)] **********************************************
skipping: [192.168.56.110]
ok: [192.168.56.109]
ok: [192.168.56.108]

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

TASK [Gathering Facts] *******************************************************
ok: [192.168.56.108]
ok: [192.168.56.110]

TASK [copy default html file for site] ***************************************
changed: [192.168.56.108]
changed: [192.168.56.110]
```



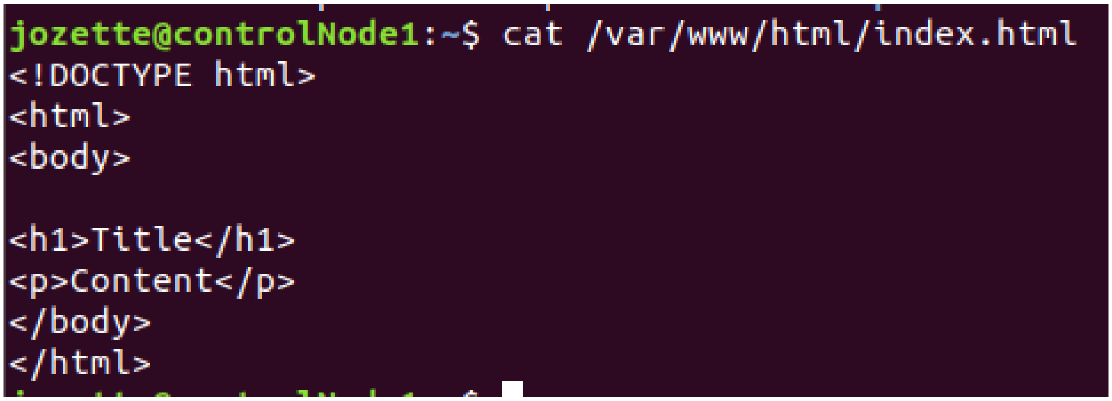
```
TASK [install apache and php for Ubuntu servers] *****************************
skipping: [192.168.56.110]
ok: [192.168.56.108]

TASK [install apache and php for CentOS servers] *****************************
skipping: [192.168.56.108]
ok: [192.168.56.110]

TASK [start httpd (CentOS)] **************************************************
skipping: [192.168.56.108]
changed: [192.168.56.110]

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

TASK [Gathering Facts] *******************************************************
ok: [192.168.56.109]
ok: [192.168.56.110]

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

TASK [Gathering Facts] *******************************************************
ok: [192.168.56.109]
```

Output



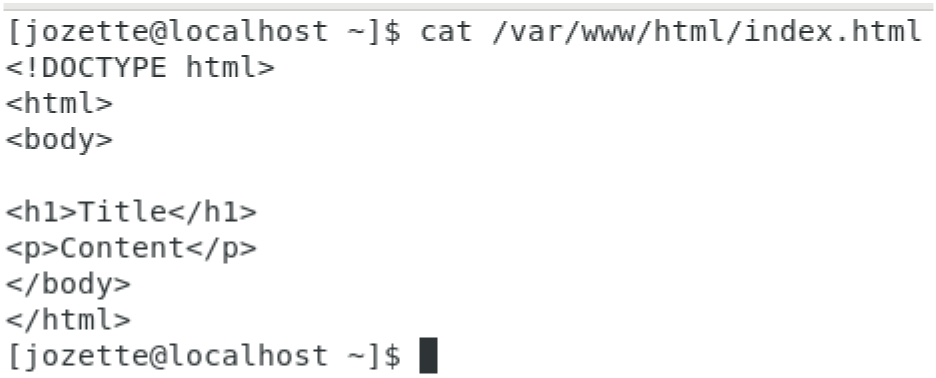
```
PLAY RECAP *********************************************************************
192.168.56.108             : ok=5    changed=1    unreachable=0    failed=0
192.168.56.109             : ok=4    changed=0    unreachable=0    failed=0
192.168.56.110             : ok=6    changed=2    unreachable=0    failed=0
```

4. Go to the remote servers (*web_servers*) listed in your inventory. Use cat command to check if the index.html is the same as the local repository file (*default_site.html*). Do both for Ubuntu and CentOS servers. On the CentOS server, go to the browser and type its IP address. Describe the output.
Ubuntu

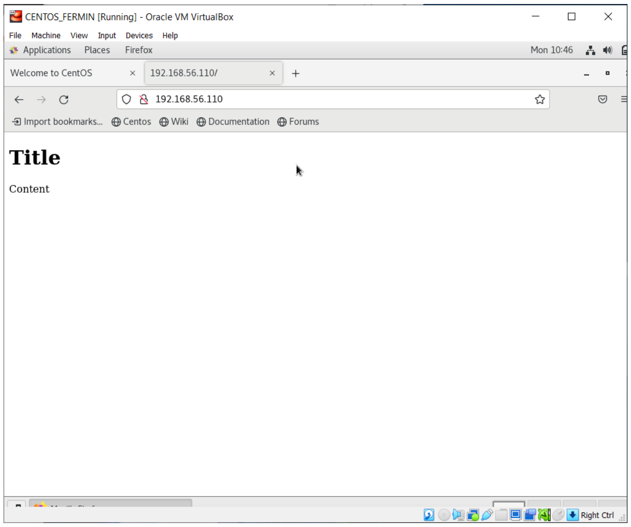


```
jozette@controlNode1:~$ cat /var/www/html/index.html
<!DOCTYPE html>
<html>
<body>

<h1>Title</h1>
<p>Content</p>
</body>
</html>
```

CentOS



```
[jozette@localhost ~]$ cat /var/www/html/index.html
<!DOCTYPE html>
<html>
<body>

<h1>Title</h1>
<p>Content</p>
</body>
</html>
[jozette@localhost ~]$ 
```

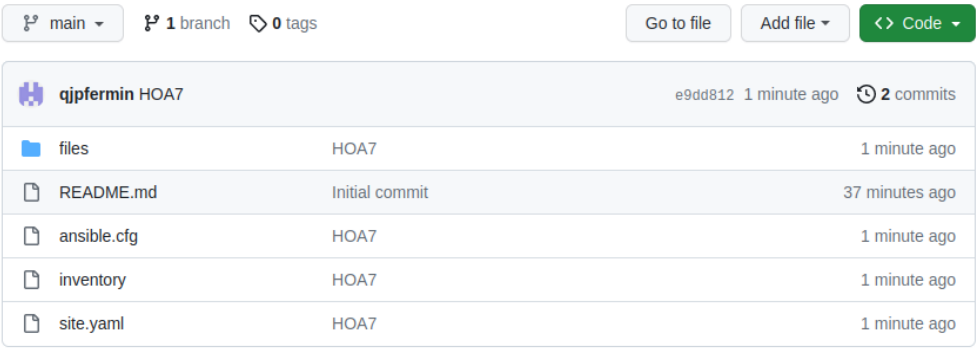


CENTOS_FERMIN [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places Firefox
Mon 10:46
Welcome to CentOS
192.168.56.110/
192.168.56.110
Import bookmarks... Centos Wiki Documentation Forums
Right Ctrl

Title

Content

5. Sync your local repository with GitHub and describe the changes.



qjpfermin HOA7 e9dd812 1 minute ago 2 commits

files	HOA7	1 minute ago
README.md	Initial commit	37 minutes ago
ansible.cfg	HOA7	1 minute ago
inventory	HOA7	1 minute ago
site.yaml	HOA7	1 minute ago

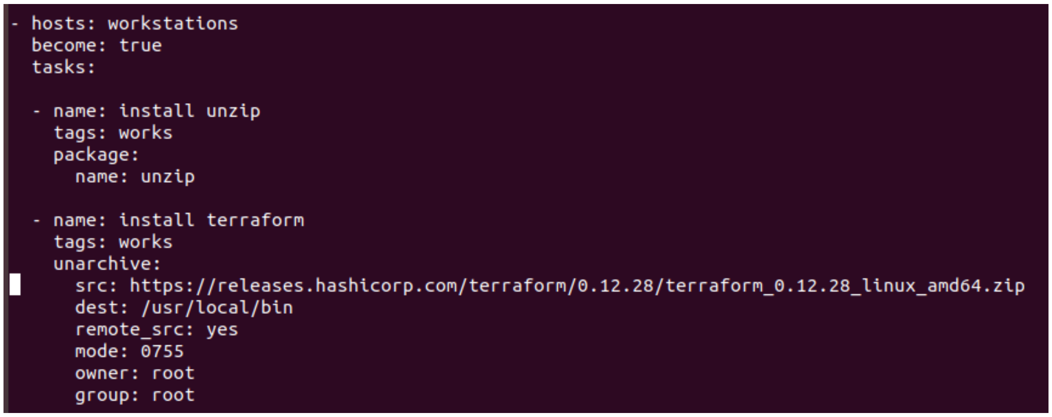
Task 2: Download a file and extract it to a remote server

1. Edit the site.yml. Just before the web_servers play, create a new play:
   - hosts: workstations
     become: true
     tasks:

```
- name: install unzip
  package:
     name: unzip

- name: install terraform
  unarchive:
    src: https://releases.hashicorp.com/terraform/0.12.28/terraform_0.12.28_linux_amd64.zip
    dest: /usr/local/bin
    remote_src: yes
    mode: 0755
    owner: root
    group: root
```

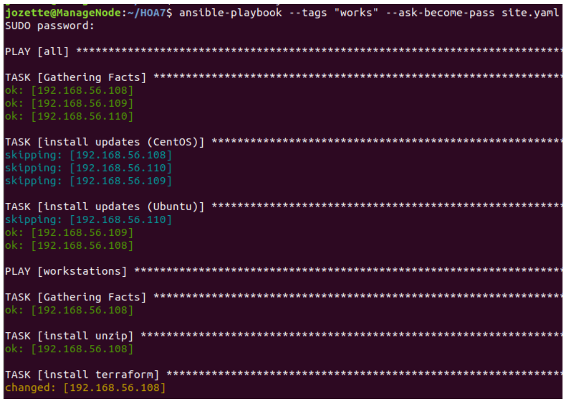
2. Edit the inventory file and add workstations group. Add any Ubuntu remote server. Make sure to remember the IP address.
3. Run the playbook. Describe the output.

<table>
<tr>
<td>Input</td>
<td>

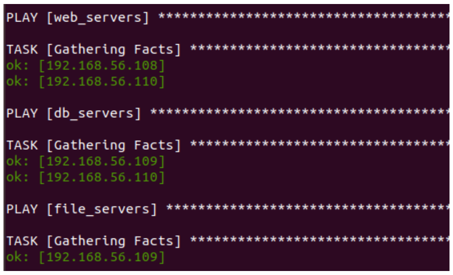


```
- hosts: workstations
  become: true
  tasks:

  - name: install unzip
    tags: works
    package:
      name: unzip

  - name: install terraform
    tags: works
    unarchive:
      src: https://releases.hashicorp.com/terraform/0.12.28/terraform_0.12.28_linux_amd64.zip
      dest: /usr/local/bin
      remote_src: yes
      mode: 0755
      owner: root
      group: root
```

</td>
</tr>
<tr>
<td>Process</td>
<td>



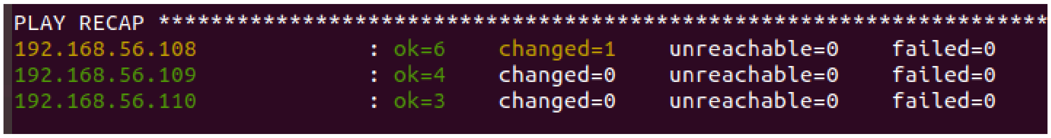
```
jozette@ManageNode:~/HOA7$ ansible-playbook --tags "works" --ask-become-pass site.yaml
SUDO password:

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

TASK [Gathering Facts] *************************************************************
ok: [192.168.56.108]
ok: [192.168.56.109]
ok: [192.168.56.110]

TASK [install updates (CentOS)] ****************************************************
skipping: [192.168.56.108]
skipping: [192.168.56.110]
skipping: [192.168.56.109]

TASK [install updates (Ubuntu)] ****************************************************
skipping: [192.168.56.110]
ok: [192.168.56.109]
ok: [192.168.56.108]

PLAY [workstations] ****************************************************************

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

TASK [install unzip] ***************************************************************
ok: [192.168.56.108]

TASK [install terraform] ***********************************************************
changed: [192.168.56.108]
```



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

TASK [Gathering Facts] *********************************
ok: [192.168.56.108]
ok: [192.168.56.110]

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

TASK [Gathering Facts] *********************************
ok: [192.168.56.109]
ok: [192.168.56.110]

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

TASK [Gathering Facts] *********************************
ok: [192.168.56.109]
```

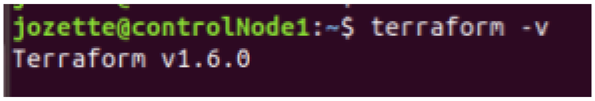
</td>
</tr>
</table>

Output



```
PLAY RECAP *************************************************************************
192.168.56.108             : ok=6    changed=1    unreachable=0    failed=0
192.168.56.109             : ok=4    changed=0    unreachable=0    failed=0
192.168.56.110             : ok=3    changed=0    unreachable=0    failed=0
```

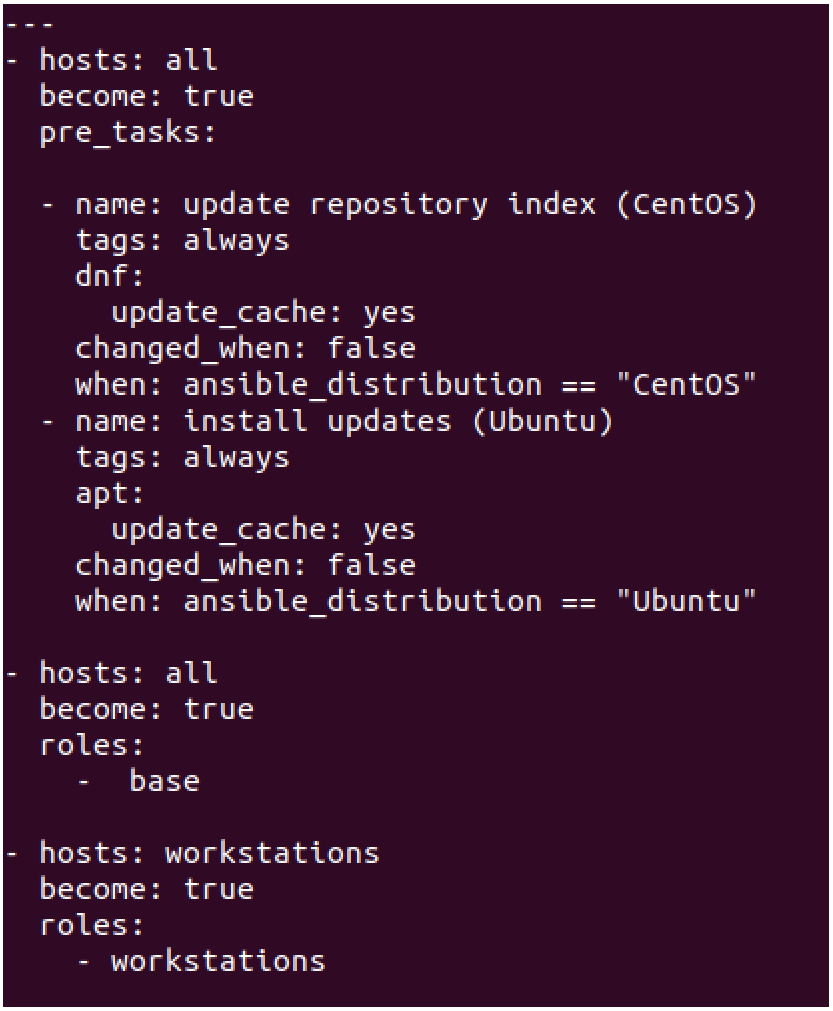
4. On the Ubuntu remote workstation, type terraform to verify installation of terraform. Describe the output.



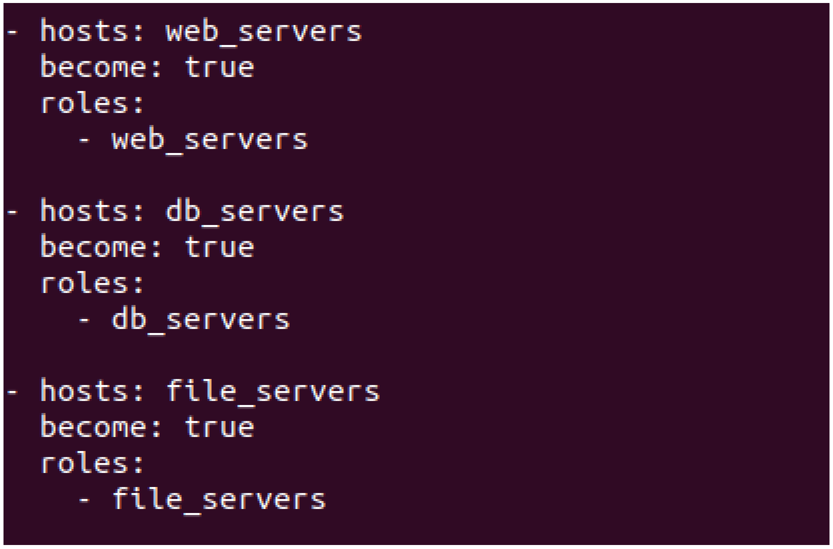
```
jozette@controlNode1:~$ terraform -v
Terraform v1.6.0
```

Task 3: Create roles

1. Edit the site.yml. Configure roles as follows: (make sure to create a copy of the old site.yml file because you will be copying the specific plays for all groups)



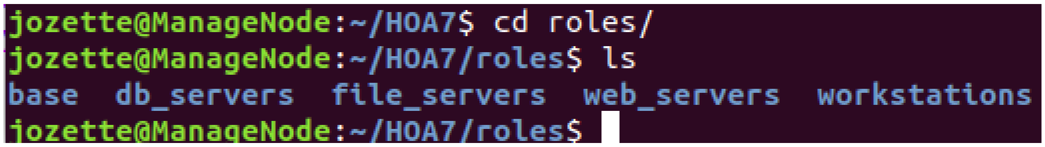
```
---
- hosts: all
  become: true
  pre_tasks:

  - name: update repository index (CentOS)
    tags: always
    dnf:
      update_cache: yes
    changed_when: false
    when: ansible_distribution == "CentOS"
  - name: install updates (Ubuntu)
    tags: always
    apt:
      update_cache: yes
    changed_when: false
    when: ansible_distribution == "Ubuntu"

- hosts: all
  become: true
  roles:
    -  base

- hosts: workstations
  become: true
  roles:
    - workstations
```



```
- hosts: web_servers
  become: true
  roles:
    - web_servers

- hosts: db_servers
  become: true
  roles:
    - db_servers

- hosts: file_servers
  become: true
  roles:
    - file_servers
```

Save the file and exit.

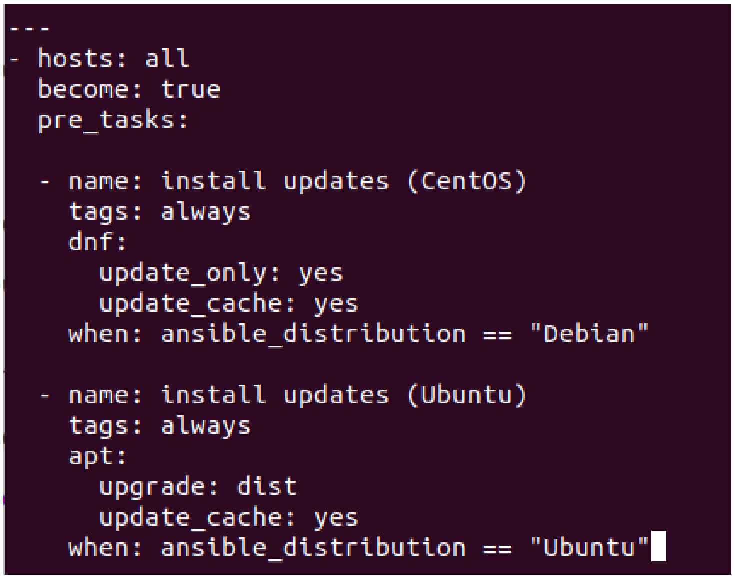
2. Under the same directory, create a new directory and name it roles. Enter the roles directory and create new directories: base, web_servers, file_servers, db_servers and workstations. For each directory, create a directory and name it tasks.



```
jozette@ManageNode:~/HOA7$ cd roles/
jozette@ManageNode:~/HOA7/roles$ ls
base  db_servers  file_servers  web_servers  workstations
jozette@ManageNode:~/HOA7/roles$ 
```

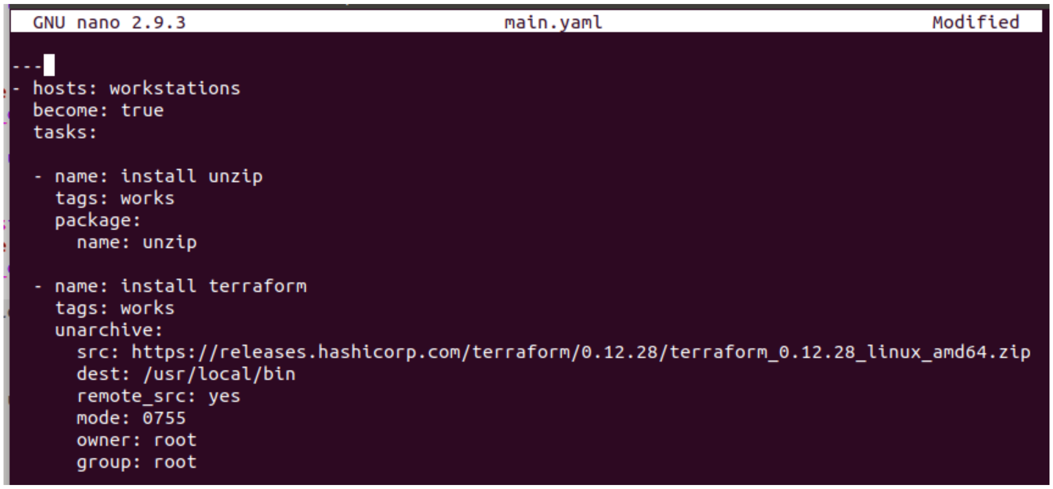
3. Go to tasks for all directory and create a file. Name it main.yml. In each of the tasks for all directories, copy and paste the code from the old site.yml file. Show all contents of main.yml files for all tasks.

Base



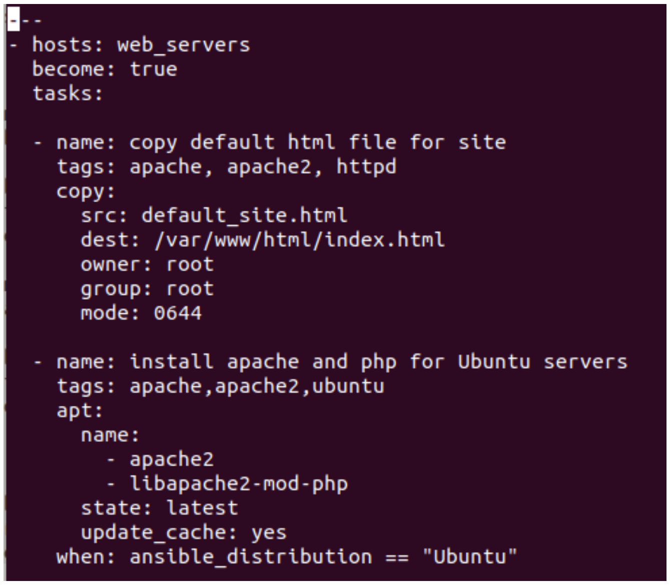
```
---
- hosts: all
  become: true
  pre_tasks:

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

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

Workstations

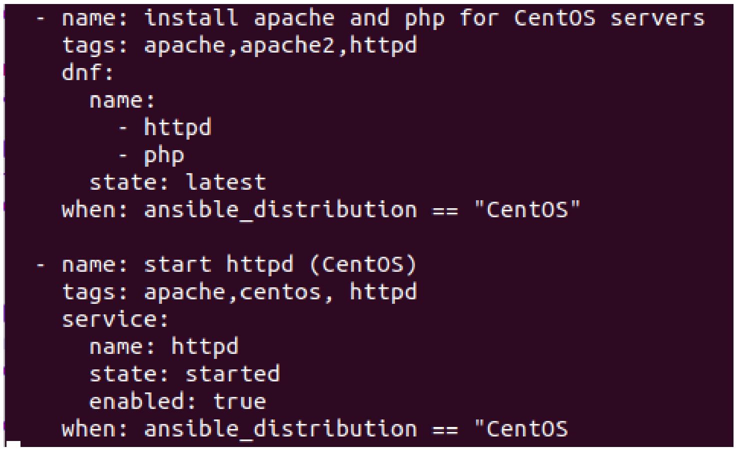


```
GNU nano 2.9.3                    main.yaml                    Modified

---
- hosts: workstations
  become: true
  tasks:

  - name: install unzip
    tags: works
    package:
      name: unzip

  - name: install terraform
    tags: works
    unarchive:
      src: https://releases.hashicorp.com/terraform/0.12.28/terraform_0.12.28_linux_amd64.zip
      dest: /usr/local/bin
      remote_src: yes
      mode: 0755
      owner: root
      group: root
```

Web_servers

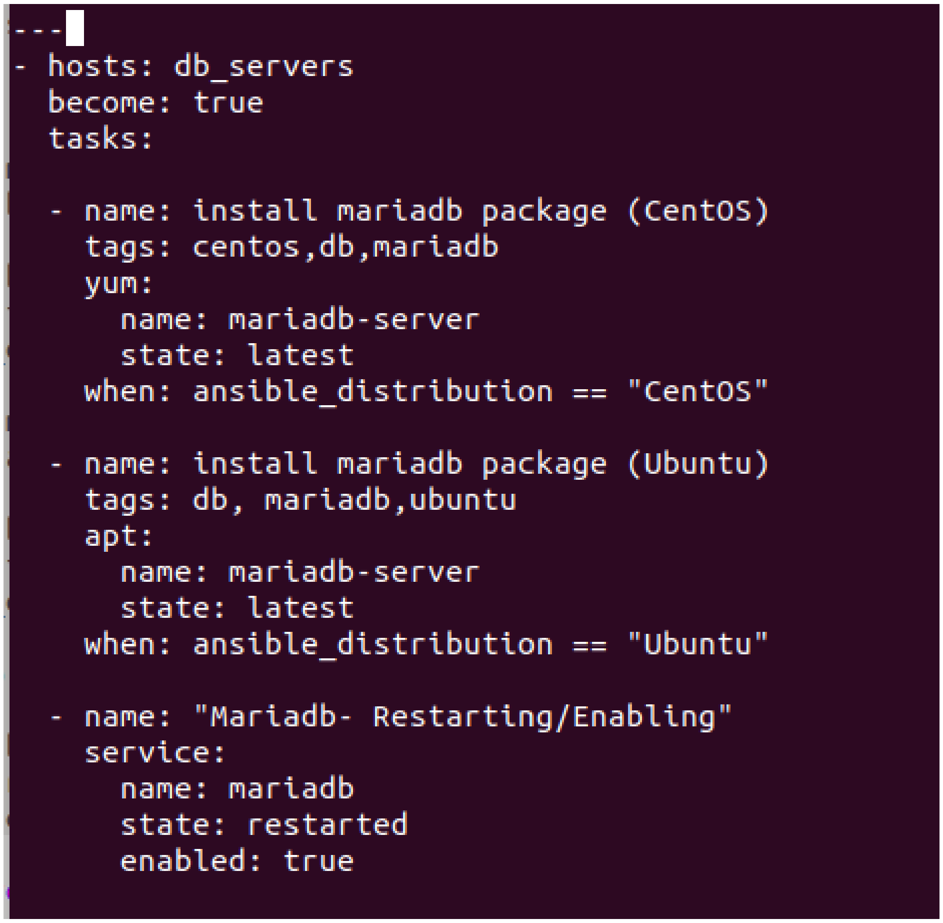


```
---
- hosts: web_servers
  become: true
  tasks:

  - name: copy default html file for site
    tags: apache, apache2, httpd
    copy:
      src: default_site.html
      dest: /var/www/html/index.html
      owner: root
      group: root
      mode: 0644

  - 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"
```



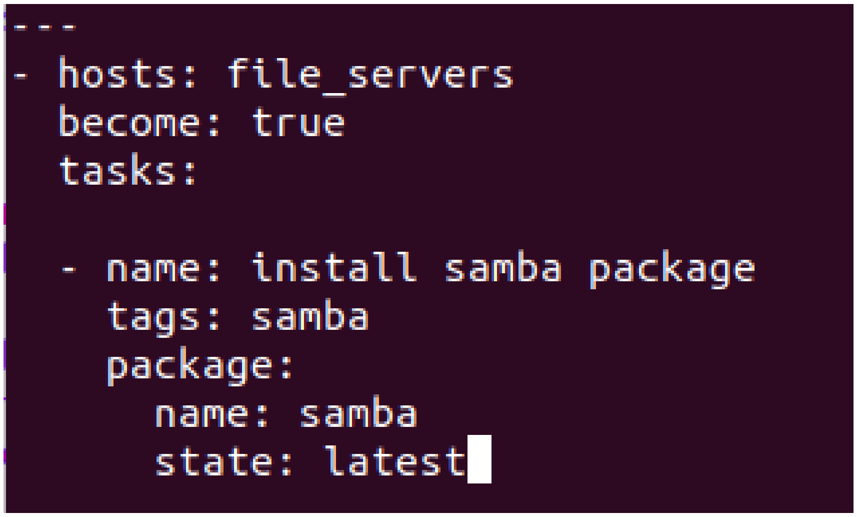
```
  - name: install apache and php for CentOS servers
    tags: apache,apache2,httpd
    dnf:
      name:
        - httpd
        - php
      state: latest
    when: ansible_distribution == "CentOS"

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

Db_servers



```
---
- 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
```

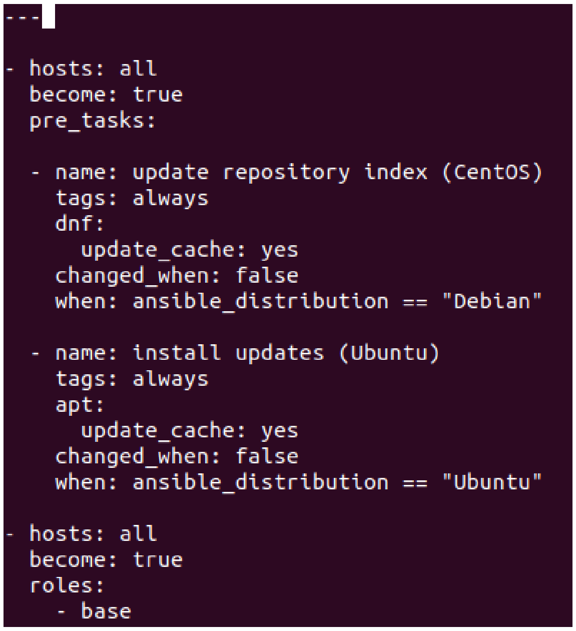
file_servers



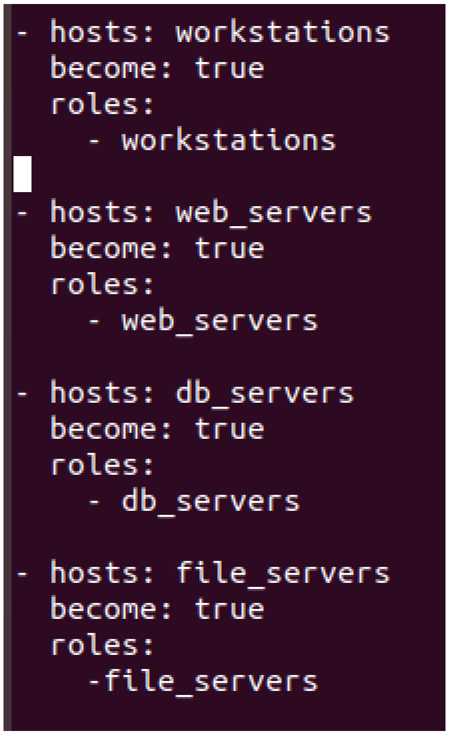
```
---
- hosts: file_servers
  become: true
  tasks:

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

4. Run the site.yml playbook and describe the output.

Input

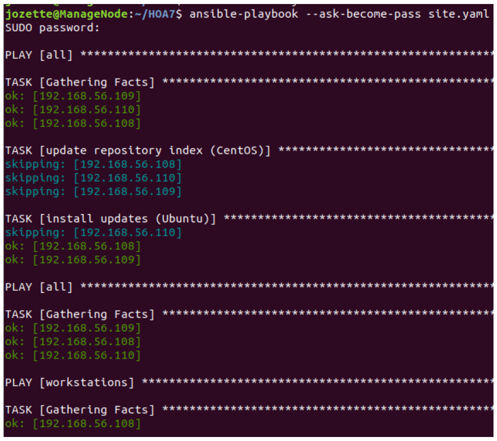


```
---

- hosts: all
  become: true
  pre_tasks:

  - name: update repository index (CentOS)
    tags: always
    dnf:
      update_cache: yes
    changed_when: false
    when: ansible_distribution == "Debian"

  - name: install updates (Ubuntu)
    tags: always
    apt:
      update_cache: yes
    changed_when: false
    when: ansible_distribution == "Ubuntu"

- hosts: all
  become: true
  roles:
    - base
```

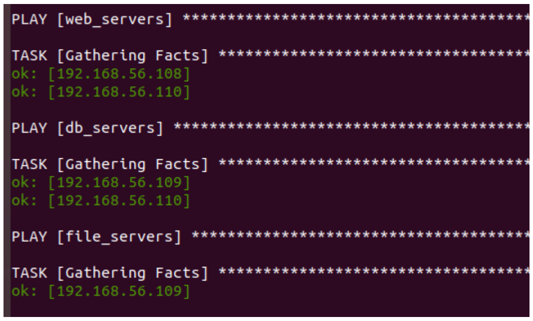


```
- hosts: workstations
  become: true
  roles:
    - workstations

- hosts: web_servers
  become: true
  roles:
    - web_servers

- hosts: db_servers
  become: true
  roles:
    - db_servers

- hosts: file_servers
  become: true
  roles:
    -file_servers
```

Process

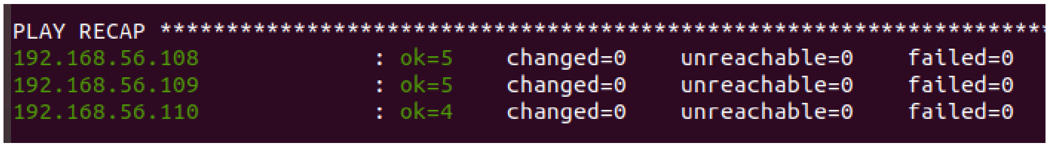


```
jozette@ManageNode:~/HOA7$ ansible-playbook --ask-become-pass site.yaml
SUDO password:

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

TASK [Gathering Facts] ************************************************
ok: [192.168.56.109]
ok: [192.168.56.110]
ok: [192.168.56.108]

TASK [update repository index (CentOS)] *******************************
skipping: [192.168.56.108]
skipping: [192.168.56.110]
skipping: [192.168.56.109]

TASK [install updates (Ubuntu)] ***************************************
skipping: [192.168.56.110]
ok: [192.168.56.108]
ok: [192.168.56.109]

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

TASK [Gathering Facts] ************************************************
ok: [192.168.56.109]
ok: [192.168.56.108]
ok: [192.168.56.110]

PLAY [workstations] ***************************************************

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



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

TASK [Gathering Facts] ************************************
ok: [192.168.56.108]
ok: [192.168.56.110]

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

TASK [Gathering Facts] ************************************
ok: [192.168.56.109]
ok: [192.168.56.110]

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

TASK [Gathering Facts] ************************************
ok: [192.168.56.109]
```

Output



```
PLAY RECAP **************************************************************
192.168.56.108             : ok=5    changed=0    unreachable=0    failed=0
192.168.56.109             : ok=5    changed=0    unreachable=0    failed=0
192.168.56.110             : ok=4    changed=0    unreachable=0    failed=0
```

Reflections:

Answer the following:

1. What is the importance of creating roles?

   The importance of creating roles is crucial as it defines clear responsibilities, promotes accountability, streamlines workflows, and maximizes the efficient utilization of skills within an organization.

2. What is the importance of managing files?

   The importance of managing files is essential to maintain data organization, integrity, security, and compliance, while also enabling effective collaboration and optimizing resource usage.

Conclusion:

In this, Hands-on activity, we manage files in remote servers and implement roles in ansible. I also learned about the importance of creating roles within an organization, emphasizing how they provide clarity, accountability, and efficient task allocation. Additionally, I explored the significance of managing files, highlighting its role in data organization, integrity, security, collaboration, compliance, and resource optimization. Both practices are integral to enhancing overall efficiency and effectiveness within an organization, ultimately contributing to its success and smooth operation.