

<b>Name: Khelvin P. Nicolas</b>	<b>Date Performed: 10/07/24</b>
<b>Course/Section: CPE 212 - CPE31S2</b>	<b>Date Submitted: 10/07/24</b>
<b>Instructor: Engr. Robin Valenzuela</b>	<b>Semester and SY: 1st Sem - '24-'25</b>
<b>Activity 7: Managing Files and Creating Roles in Ansible</b>	
<b>1. Objectives:</b> 1.1 Manage files in remote servers 1.2 Implement roles in ansible	
<b>2. Discussion:</b>  <p>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 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.</p>	
<b>Task 1: Create a file and copy it to remote servers</b>  <ol style="list-style-type: none"> <li>Using the previous directory we created, create a directory, and name it "<i>files</i>." Create a file inside that directory and name it "<i>default_site.html</i>." 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.</li> <li>Edit the <i>site.yml</i> file and just below the <i>web_servers</i> play, create a new file to copy the default html file for site: <ul style="list-style-type: none"> <li>name: copy default html file for site</li> <li>tags: apache, apache2, httpd</li> <li>copy: <ul style="list-style-type: none"> <li>src: default_site.html</li> <li>dest: /var/www/html/index.html</li> <li>owner: root</li> <li>group: root</li> <li>mode: 0644</li> </ul> </li> </ul> </li> <li>Run the playbook <i>site.yml</i>. Describe the changes.</li> </ol>	

```

punopaughey@workstation: ~/CPE212_Ac... x punopaughey@workstation: ~/CPE212_Ac... x
skipping: [server1]
skipping: [server2]

PLAY [db_servers] *****

PLAY [file_servers] *****

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

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

PLAY RECAP *****
centos                : ok=4    changed=0    unreachable=0    failed=0    s
kipped=1    rescued=0    ignored=0
server1             : ok=5    changed=0    unreachable=0    failed=0    s
kipped=3    rescued=0    ignored=0
server2             : ok=5    changed=0    unreachable=0    failed=0    s
kipped=3    rescued=0    ignored=0
server3             : ok=0    changed=0    unreachable=1    failed=0    s
kipped=0    rescued=0    ignored=0

punopaughey@workstation:~/CPE212_Activity7$

```

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.
  - the contents of the `default_file.html` was copied into the `index.html` in the remote servers using the playbook file. and when entering the IP address of CentOS in the browser it runs the `index.html` file found in the `/var/www/html` directory.

```
punopaughey@workstation:~/CPE212_Activity7$ ssh punopaughey@server1
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 5.4.0-150-generic x86_64)
```

```
* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:        https://ubuntu.com/pro
```

Expanded Security Maintenance for Infrastructure is not enabled.

0 updates can be applied immediately.

226 additional security updates can be applied with ESM Infra.  
Learn more about enabling ESM Infra service for Ubuntu 18.04 at  
<https://ubuntu.com/18-04>

New release '20.04.6 LTS' available.  
Run 'do-release-upgrade' to upgrade to it.

Your Hardware Enablement Stack (HWE) is supported until April 2023.  
Last login: Mon Oct 7 08:55:46 2024 from 192.168.56.10

```
punopaughey@server1:~$ ls -l /var/www/html
```

```
total 4
```

```
-rw-r--r-- 1 root root 105 Oct 7 08:33 index.html
```

```
punopaughey@server1:~$ cat /var/www/html/index.html
```

```
<html>
```

```
  <head>
```

```
    <meta charset="utf-8">
```

```
  </head>
```

```
  <body>
```

```
    <header>Sample Text</header>
```

```
  </body>
```

```
</html>
```

```
punopaughey@server1:~$ █
```

```
punopaughey@workstation:~/CPE212_Activity7$ ssh khlvn@centos
Last login: Sun Oct 6 21:13:38 2024 from 192.168.56.10
```

```
[khlvn@centos ~]$ cat /var/www/html/index.html
```

```
<html>
```

```
  <head>
```

```
    <meta charset="utf-8">
```

```
  </head>
```

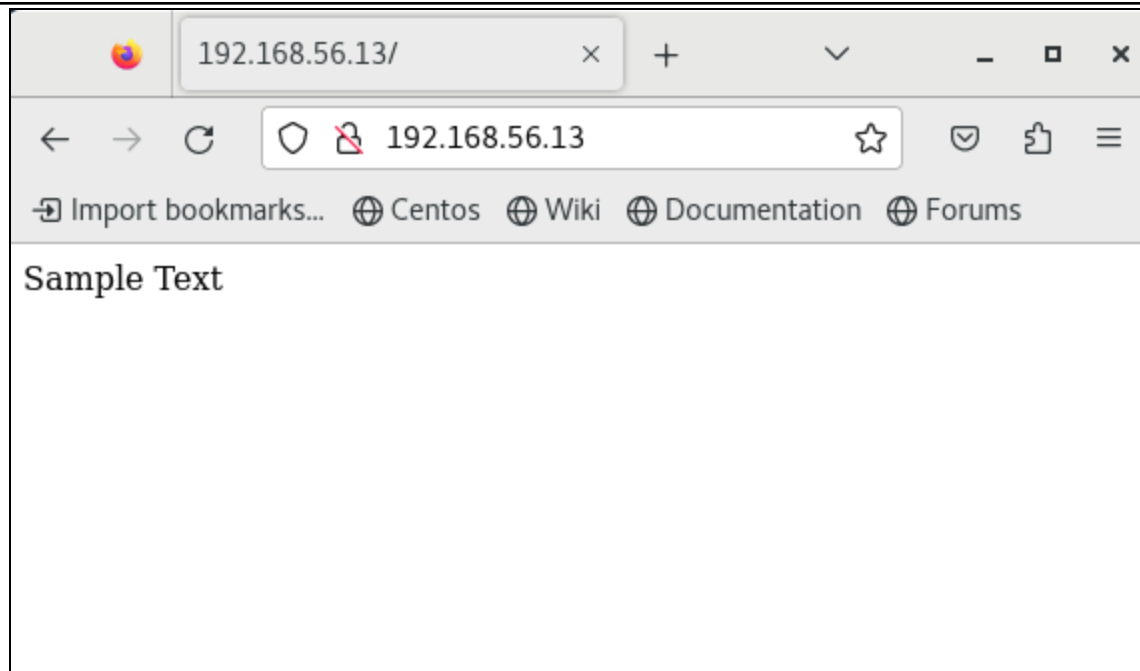
```
  <body>
```

```
    <header>Sample Text</header>
```

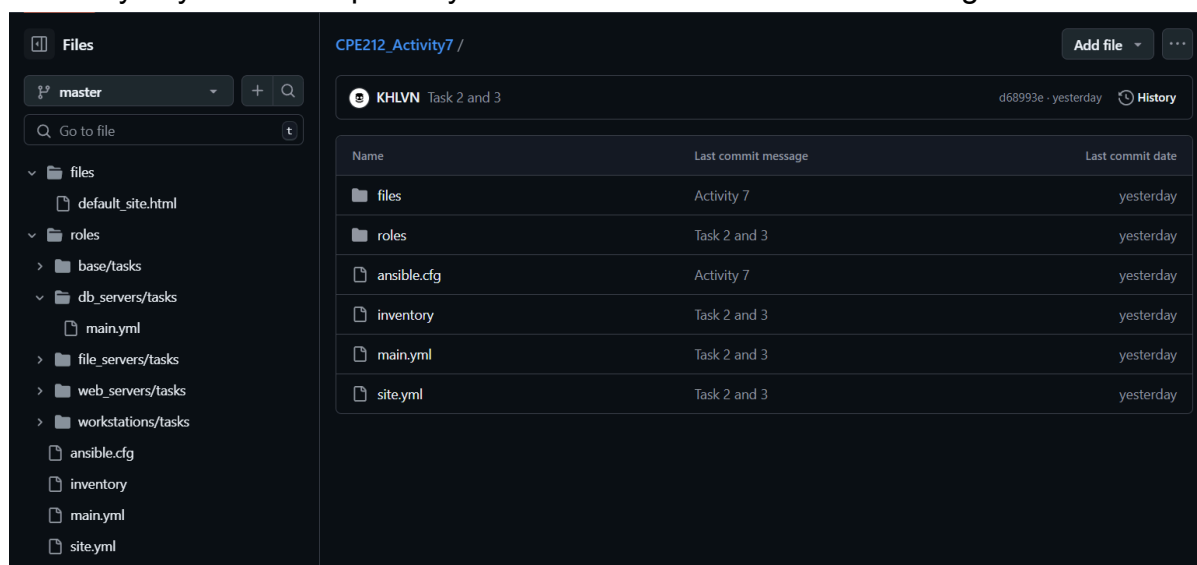
```
  </body>
```

```
</html>
```

```
[khlvn@centos ~]$ █
```



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



**I pushed the files from my local repository to the remote repository to upload/update my created playbooks and directories into the latest version to GitHub.**

## 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:

<https://github.com/prometheus/prometheus/releases/download/v2.30.0/prometheus-2.30.0.linux-amd64.tar.gz>

src:

[https://releases.hashicorp.com/terraform/0.12.28/terraform\\_0.12.28\\_linux\\_amd64.zip](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

```
- 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.

- **Since I have put server1 inside the inventory file under the workstations group, when running the playbook, it performs installation of the unzip package and downloads terraform using the official website for terraform, by unarchiving the contents that were downloaded in the website into the local bin directory in linux.**

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

```
punopaughey@server1:~$ terraform
Usage: terraform [-version] [-help] <command> [args]

The available commands for execution are listed below.
The most common, useful commands are shown first, followed by
less common or more advanced commands. If you're just getting
started with Terraform, stick with the common commands. For the
other commands, please read the help and docs before usage.

Common commands:
  apply          Builds or changes infrastructure
  console        Interactive console for Terraform interpolations
  destroy        Destroy Terraform-managed infrastructure
  env            Workspace management
  fmt            Rewrites config files to canonical format
  get            Download and install modules for the configuration
  graph          Create a visual graph of Terraform resources
  import         Import existing infrastructure into Terraform
  init           Initialize a Terraform working directory
```

When typing terraform in the terminal, it shows the usage, options and common commands that we can use.

### 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.
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.

```
punopaughey@workstation:~/CPE212_Activity7$ mv site1.yml main.yml
punopaughey@workstation:~/CPE212_Activity7$ ls
ansible.cfg  files  inventory  main.yml  roles  site.yml
punopaughey@workstation:~/CPE212_Activity7$ ls -la roles
total 28
drwxrwxr-x 7 punopaughey punopaughey 4096 Oct  7 09:40 .
drwxrwxr-x 5 punopaughey punopaughey 4096 Oct  7 09:43 ..
drwxrwxr-x 3 punopaughey punopaughey 4096 Oct  7 09:40 base
drwxrwxr-x 3 punopaughey punopaughey 4096 Oct  7 09:40 db_servers
drwxrwxr-x 3 punopaughey punopaughey 4096 Oct  7 09:40 file_servers
drwxrwxr-x 3 punopaughey punopaughey 4096 Oct  7 09:41 web_servers
drwxrwxr-x 3 punopaughey punopaughey 4096 Oct  7 09:41 workstations
punopaughey@workstation:~/CPE212_Activity7$ cp main.yml /roles/b*/tasks
cp: cannot create regular file '/roles/b*/tasks': No such file or directory
punopaughey@workstation:~/CPE212_Activity7$ cp main.yml roles/b*/tasks
punopaughey@workstation:~/CPE212_Activity7$ cp main.yml roles/d*/tasks
punopaughey@workstation:~/CPE212_Activity7$ cp main.yml roles/f*/tasks
punopaughey@workstation:~/CPE212_Activity7$ cp main.yml roles/web*/tasks
punopaughey@workstation:~/CPE212_Activity7$ cp main.yml roles/wor*/tasks
punopaughey@workstation:~/CPE212_Activity7$ ls roles/b*/tasks
main.yml
punopaughey@workstation:~/CPE212_Activity7$
```

	main.yml
base	<pre>--- - 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"</pre>



<b>db_servers</b>	<pre>--- - 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</pre>
<b>file_servers</b>	<pre>--- - name: install samba package   tags: samba   package:     name: samba     state: latest</pre>

## web\_servers

```
---
- 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, 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
    enabled: true
  when: ansible_distribution == "CentOS"
```

## workstations

```
---
- 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
```

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

```
punopaughey@workstation: ~/CPE212_Ac... x punopaughey@workstation: ~/CPE212_Ac... x
PLAY [db_servers] *****
TASK [Gathering Facts] *****
ok: [centos]

TASK [db_servers : install mariadb package (CentOS)] *****
ok: [centos]

TASK [db_servers : install mariadb package (Ubuntu)] *****
skipping: [centos]

TASK [db_servers : Mariadb- Restarting/Enabling] *****
changed: [centos]

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

TASK [file_servers : install samba package] *****
ok: [centos]

PLAY RECAP *****
centos                : ok=13    changed=1    unreachable=0    failed=0    s
kipped=4             rescued=0    ignored=0
server1              : ok=10    changed=0    unreachable=0    failed=0    s
kipped=4             rescued=0    ignored=0
server2              : ok=7     changed=0    unreachable=0    failed=0    s
kipped=4             rescued=0    ignored=0
server3              : ok=0     changed=0    unreachable=1    failed=0    s
kipped=0             rescued=0    ignored=0

punopaughey@workstation: ~/CPE212 Activity7$
```

- In the `site.yml` file we created earlier, we assigned roles to each group of remote servers (workstations, database servers, web servers, file servers) and created a directory for each role. Inside each role's directory, there is a `tasks` directory, which contains a `main.yml` playbook that executes specific tasks.

**Reflections:**

Answer the following:

**1. What is the importance of creating roles?**

- Creating roles allows you to organize related tasks, handlers, and configurations into modular pieces, making playbooks easier to understand and be more manageable. Similar to programming, code reusability across multiple projects can be done. Also it promotes easier maintenance and scalability among your servers.

**2. What is the importance of managing files?**

- File management keeps files structured, making it easier to manage and locate information. It also promotes security, ensuring configuration files are stored securely and hidden within only the specified roles. Effective file management also helps in backup and recovery, ensuring that data can be restored quickly in the event of a data loss. Overall, it improves efficiency by reducing clutter and avoiding file duplication, ultimately streamlining workflows.