

Name: Alexis Neil D. Deniega	Date Performed: October 3rd, 2025
Course/Section: CPE31S4	Date Submitted: October 3rd, 2025
Instructor: Engr. Robin Valenzuela	Semester and SY: SY 2025-26, Sem 1

## 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.

```
Files > default_site.html > html > body
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width,
6          initial-scale=1.0">
7      <title>Default Page</title>
8  </head>
9  <body>
10     <h1>Welcome to the Default Page!</h1>
11     <p>This is a simple HTML Document</p>
12
13     <p> Lorem ipsum dolor sit amet consectetur adipisicing elit.
14         Veniam, accusamus, dicta laborum dolorem iure ab recusandae
15         quis non nesciunt excepturi quas commodi unde at alias
16         cupiditate, consectetur architecto eum placeat.</p>
17
18
19  </body>
20
21  </html>
```

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

```
44     service:
45         name: httpd
46         state: started
47         enabled: true
48         when: ansible_distribution == "CentOS"
49
50     - name: copy default html file for site
51       tags: apache, apache2, httpd
52       copy:
53           src: files/default_site.html
54           dest: /var/www/html/index.html
55           owner: root
56           group: root
57           mode: 0644
58
59   - hosts: db_servers
60     become: true
61     tasks:
62       - name: install mariadb package (centos)
63         tags: centos, db, mariadb
```

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

```
TASK [start httpd (centos)] ****
*****
skipping: [192.168.56.116]
skipping: [192.168.56.117]
ok: [192.168.56.121]

TASK [copy default html file for site] ****
*****
changed: [192.168.56.121]
changed: [192.168.56.116]
changed: [192.168.56.117]

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

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

*It says changed.*

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.

```
programmymain@server1:~$ cd /var
programmymain@server1:/var$ ls
backups  crash  local  log   metrics  run   spool  www
cache    lib    lock   mail   opt     snap   tmp
programmymain@server1:/var$ cd www
programmymain@server1:/var/www$ ls
html
programmymain@server1:/var/www$ cd html
programmymain@server1:/var/www/html$ cat index.html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Default Page</title>
</head>
<body>
    <h1>Welcome to the Default Page!</h1>
    <p>This is a simple HTML Document</p>

    <p> Lorem ipsum dolor sit amet consectetur adipisicing elit. Veniam, accusamus, dicta laborum dolorem iure ab recusandae quis non nesciunt excepturi quas commodi unde at alias cupiditate, consectetur architecto eum placeat.</p>

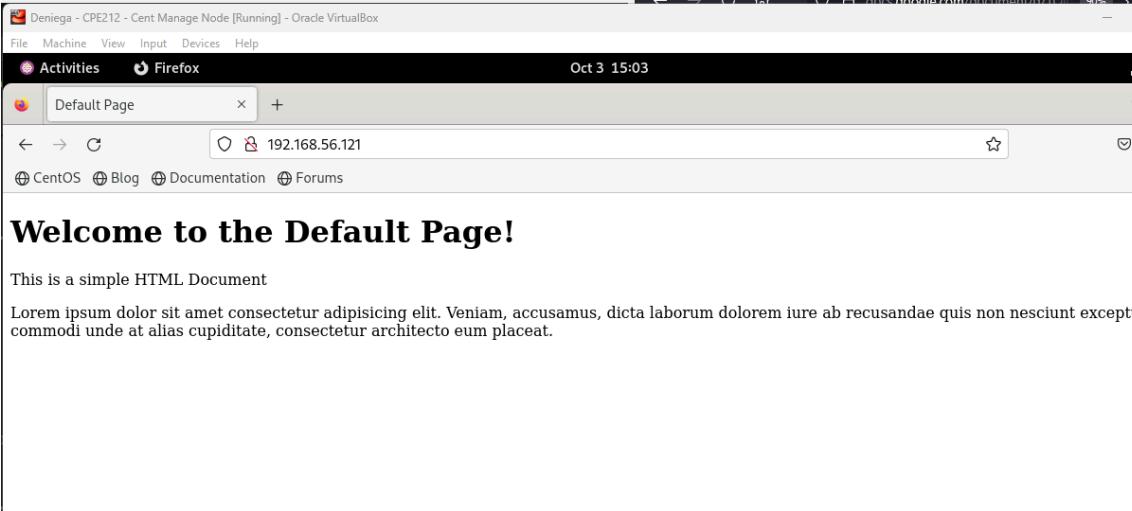
</body>
</html>programmymain@server1:/var/www/html$
```

2.16.3 Lightspeed (Not logged in) Python 3.12.3

```
Last login: Fri Oct  3 14:52:43 2025 from 192.168.56.115
programmymain@server2:~$ cat /var/www/html/index.html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Default Page</title>
</head>
<body>
    <h1>Welcome to the Default Page!</h1>
    <p>This is a simple HTML Document</p>

        <p> Lorem ipsum dolor sit amet consectetur adipisicing elit. Veniam, accusamus, dicta laborum dolorem iure ab recusandae quis non nesciunt excepturi quas commodi unde at alias cupiditate, consectetur architecto eum placeat.</p>

    </body>
</html>programmymain@server2:~$
```



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

```

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working
  directory)
    modified:  install_apache.yml
    modified:  site.yml

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .vscode/
    files/
    roles/
    site2.yml

no changes added to commit (use "git add" and/or "git commit
-a")
# programmain@workstation:~/CPE232_Deniega$ git add -A
# programmain@workstation:~/CPE232_Deniega$ git commit -m HOA
7 Part 1
error: pathspec '7' did not match any file(s) known to git
error: pathspec 'Part' did not match any file(s) known to git
error: pathspec '1' did not match any file(s) known to git
# programmain@workstation:~/CPE232_Deniega$ git commit -m "HOA
A 7 Part 1"
[main 8135f3] HOA 7 Part 1
 18 files changed, 31 insertions(+), 6 deletions(-)
 create mode 106644 .vscode/settings.json
 create mode 106644 files/default_site.html
 create mode 106644 roles/base/tasks/main.yml
 create mode 106644 roles/db_servers/tasks/main.yml
 create mode 106644 roles/file_servers/tasks/main.yml
 create mode 106644 roles/web_servers/tasks/main.yml
 create mode 106644 roles/workstation/tasks/main.yml
 create mode 106644 site2.yml
 create mode 106644 site.yml
# programmain@workstation:~/CPE232_Deniega$ git push origin main
Enumerating objects: 16, done.
Counting objects: 100% (16/16), done.
Delta compression using up to 6 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (13/13), 1.44 KiB | 1.44 MiB/s, done.
Total 13 (delta 0), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local
objects.
To github.com:Alexis-acad/CPE232_Deniega.git
 * [new branch] main -> main
# programmain@workstation:~/CPE232_Deniega$ git status
# On branch main
# Your branch is up-to-date with 'origin/main'.
#   nothing to commit, working tree clean
# programmain@workstation:~/CPE232_Deniega$ ls
# programmain@workstation:~/CPE232_Deniega$ python 3.12.3 --version
Python 3.12.3
# programmain@workstation:~/CPE232_Deniega$
```

Alexis-acad HOA 7 Part 1

File	Last Commit	Time Ago
.vscode	HOA 7 Part 1	now
files	HOA 7 Part 1	now
group_vars/all	HOA 6 finished	2 weeks ago
roles	HOA 7 Part 1	now
README.md	HOA 6 finished	2 weeks ago
ansible.cfg	HOA 6 finished	2 weeks ago
install_apache.yml	HOA 7 Part 1	now
install_apache_redhat.yml	HOA 5 final	3 weeks ago
inventory.ini	HOA 6 finished	2 weeks ago
site.yml	HOA 7 Part 1	now
site2.yml	HOA 7 Part 1	now

**CPE232\_Deniega**

This is my first repository for the course CPE 212 - Automating Server Management.

**What is it about?**

## 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](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

```

55     owner: root
56     group: root
57     mode: 0644
58
59     - hosts: workstations
60       become: true
61       tasks:
62         - name: install unzip
63           package:
64             name: unzip
65         - name: install terraform
66           unarchive:
67             src: https://releases.hashicorp.com/terraform/0.12.28/
68             terraform_0.12.28_linux_amd64.zip
69             dest: /usr/local/bin
70             remote_src: yes
71             mode: 0755
72             owner: root
73             group: root
74
75     - hosts: db_servers
76       become: true
77       tasks:

```

[Go to Line/Column](#)

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

```

inventory.ini M ✘ site.yml M ✘ site2.yml ✘ default_site.h
inventory.ini
1 [web_servers]
2 192.168.56.116
3 192.168.56.117
4 192.168.56.121 ansible_user=alexis-cent
5
6 [workstations]
7 192.168.56.116
8 192.168.56.121 ansible_user=alexis-cent
9 192.168.56.123
10
11 [db_servers]
12 192.168.56.121 ansible_user=alexis-cent
13
14 [file_servers]
15 192.168.56.123
16
17 [all:vars]
18 ansible_ssh_common_args=' -o GSSAPIAuthentication=no'
19

```

3. Run the playbook. Describe the output.

```
ok: [192.168.56.115]
ok: [192.168.56.117]

PLAY [workstations] *****
*****
TASK [Gathering Facts] *****
*****
ok: [192.168.56.121]
ok: [192.168.56.116]
ok: [192.168.56.123]

TASK [install unzip] *****
*****
ok: [192.168.56.121]
ok: [192.168.56.116]
ok: [192.168.56.123]

TASK [install terraform] *****
*****
changed: [192.168.56.121]
changed: [192.168.56.123]
changed: [192.168.56.116]

PLAY [db_servers] *****
*****
TASK [Gathering Facts] *****
*****
ok: [192.168.56.121]

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

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

```
programmymain@workstation:~/CPE232_Deniega$ ssh programmymain@192.168.56.123
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-29-generic x8
6_64)

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

Expanded Security Maintenance for Applications is not enabled
.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates
.

See https://ubuntu.com/esm or run: sudo pro status

*** System restart required ***
Last login: Fri Oct  3 15:11:39 2025 from 192.168.56.115
programmymain@server3:~$ 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 t
he
other commands, please read the help and docs before usage.
```

*It's just the help thing, but it works...*

### 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.

```

⑧ site2.yml
1   ---
2   - hosts: all
3     become: true
4     pre_tasks:
5
6     - name: update repo index (centos)
7       tags: always
8       dnf:
9         | update_cache: yes
10        changed_when: false
11        when: ansible_distribution == "CentOS"
12
13     - name: install updates (ubuntu)
14       tags: always
15       apt:
16         | update_cache: yes
17        changed_when: false
18        when: ansible_distribution == "Ubuntu"
19
20     - hosts: all
21       become: true
22       roles:
23         - base
24
25     - hosts: workstations
26       become: true
27       roles:
28         - workstations
29
30     - hosts: web_servers
31       become: true
32       roles:
33         - web_servers
34
35     - hosts: db_servers
36       become: true
37       roles:
38         - db_servers
39
40     - hosts: file_servers
41       become: true
42       roles:
43         - file_servers

```

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.

```

● programmymain@workstation:~/CPE232_Deniega$ mkdir -p roles/{w
orkstation,base,web_servers,file_servers,db_servers}/tasks

```

```
Processing triggers for man-db (2.12.0-4ubuntu0.2) ...
● programmymain@workstation:~/CPE232_Deniega$ tree
.
├── ansible.cfg
├── group_vars
│   └── all
│       └── vault.yaml
├── install_apache_redhat.yml
├── install_apache.yml
├── inventory.ini
└── README.md
    └── roles
        ├── base
        │   └── tasks
        ├── db_servers
        │   └── tasks
        ├── file_servers
        │   └── tasks
        ├── web_servers
        │   └── tasks
        └── workstation
            └── tasks
    └── site.yml
```

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.

```
● programmymain@workstation:~/CPE232_Deniega$ touch roles/{work
station,base,web_servers,file_servers,db_servers}/tasks/main.
yml
● programmymain@workstation:~/CPE232_Deniega$ tree
.
├── ansible.cfg
├── files
│   └── default_site.html
├── group_vars
│   └── all
│       └── vault.yaml
├── install_apache_redhat.yaml
├── install_apache.yaml
├── inventory.ini
└── README.md

    ├── roles
    │   ├── base
    │   │   └── tasks
    │   │       └── main.yaml
    │   ├── db_servers
    │   │   └── tasks
    │   │       └── main.yaml
    │   ├── file_servers
    │   │   └── tasks
    │   │       └── main.yaml
    │   ├── web_servers
    │   │   └── tasks
    │   │       └── main.yaml
    │   └── workstation
    │       └── tasks
    │           └── main.yaml
    └── site.yaml

15 directories, 13 files
```

The screenshot shows a terminal window with the following content:

```
● programmymain@workstation:~/CPE232_Deniega$ tree
.
├── ansible.cfg
├── files
│   └── default_site.html
├── group_vars
│   └── all
│       └── vault.yaml
├── install_apache_redhat.yaml
├── install_apache.yaml
├── inventory.ini
└── README.md

    ├── roles
    │   ├── base
    │   │   └── tasks
    │   │       └── main.yaml
    │   ├── db_servers
    │   │   └── tasks
    │   │       └── main.yaml
    │   ├── file_servers
    │   │   └── tasks
    │   │       └── main.yaml
    │   ├── web_servers
    │   │   └── tasks
    │   │       └── main.yaml
    │   └── workstation
    │       └── tasks
    │           └── main.yaml
    └── site.yaml

15 directories, 13 files
```

The terminal is running on a system named "programmymain" with the IP address 192.168.1.11. The current working directory is "/home/programmymain/CPE232\_Deniega". The user has run the command "tree" to display the directory structure. The structure includes an "ansible.cfg" file, a "files" directory containing "default\_site.html", a "group\_vars" directory with a "all" entry containing "vault.yaml", and four "install" files ("install\_apache\_redhat.yaml", "install\_apache.yaml"). There is also an "inventory.ini" file and a "README.md" file. The "roles" directory contains five sub-directories: "base", "db\_servers", "file\_servers", "web\_servers", and "workstation", each with its own "tasks" sub-directory containing a "main.yaml" file. Finally, there is a "site.yaml" file at the root level. The message "15 directories, 13 files" is displayed at the bottom of the terminal output.

```
roles > db_servers > tasks > main.yml
1  ---
2
3 - name: install mariadb package (centos)
4   tags: centos, db, mariadb
5   yum:
6     name: mariadb-server
7     state: latest
8     when: ansible_distribution == "CentOS"
9
10 - name: "mariadb -- restarting/enabling"
11   service:
12     name: mariadb
13     state: restarted
14     enabled: true
15
16 - name: install mariadb package (ubuntu)
17   tags: db, mariadb, ubuntu
18   apt:
19     name: mariadb-server
20     state: latest
21     when: ansible_distribution == "Ubuntu"
```

```
roles > file_servers > tasks > main.yml
1  ---
2
3 - name: install samba package
4   tags: samba
5   package:
6     name: samba
7     state: latest
8
```

```
roles > web_servers > tasks > main.yml
1 ...
2 ...
3 - name: install apache and php for Ubuntu servers
4   tags: apache, apache2, ubuntu
5   apt:
6     name:
7       - apache2
8       - libapache2-mod-php
9     state: latest
10    update_cache: yes
11    when: ansible_distribution == "Ubuntu"
12
13 - name: install apache and php for CentOS servers
14   tags: apache,centos,httpd
15   dnf:
16     name:
17       - httpd
18       - php
19     state: latest
20   when: ansible_distribution == "CentOS"
21
22 - name: start httpd (centos)
23   tags: apache, centos, httpd
24   service:
25     name: httpd
26     state: started
27     enabled: true
28   when: ansible_distribution == "CentOS"
29
30 - name: copy default html file for site
31   tags: apache, apache2, httpd
32   copy:
33     src: files/default_site.html
34     dest: /var/www/html/index.html
35     owner: root
36     group: root
37     mode: 0644
```

```
roles > workstations > tasks > main.yml
1  ---
2
3  - name: install_unzip
4    package:
5      name: unzip
6
7  - name: install_terraform
8    unarchive:
9      src: https://releases.hashicorp.com/terraform/0.12.28/
10     terraform_0.12.28_linux_amd64.zip
11     dest: /usr/local/bin
12     remote_src: yes
13     mode: 0755
14     owner: root
15     group: root
```

**NOTE: I know that there are a lot of squiggly lines here but trust me this is working!**

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

```
programmymain@workstation:~/CPE232_Deniega$ ansible-playbook site2.yml -K
ok: [192.168.56.121]
ok: [192.168.56.117]
ok: [192.168.56.116]

PLAY [db_servers] ****
*****
TASK [Gathering Facts] ****
*****
ok: [192.168.56.121]

TASK [db_servers : install mariadb package (centos)] ****
*****
ok: [192.168.56.121]

TASK [db_servers : mariadb -- restarting/enabling] ****
*****
changed: [192.168.56.121]

TASK [db_servers : install mariadb package (ubuntu)] ****
*****
skipping: [192.168.56.121]

PLAY [file_servers] ****
*****
TASK [Gathering Facts] ****
*****
ok: [192.168.56.123]

TASK [file_servers : install samba package] ****
*****
ok: [192.168.56.123]

PLAY RECAP ****
*****
192.168.56.116      : ok=9    changed=0    unreachable
                      =0    failed=0   skipped=3   rescued=0   ignored=0
192.168.56.117      : ok=6    changed=0    unreachable
                      =0    failed=0   skipped=3   rescued=0   ignored=0
192.168.56.121      : ok=13   changed=1    unreachable
                      =0    failed=0   skipped=3   rescued=0   ignored=0
192.168.56.123      : ok=8    changed=0    unreachable
                      =0    failed=0   skipped=1   rescued=0   ignored=0

○ programmymain@workstation:~/CPE232_Deniega$
```

As you can see here, each play called the role's .yml file instead of putting it in one big file.

	Alexis-acad	HOA 7 final	1f7fd40 · now
	.vscode	HOA 7 final	n
	files	HOA 7 Part 1	48 minutes a
	group_vars/all	HOA 6 finished	2 weeks a
	roles	<a href="#">HOA 7 final</a>	n
	README.md	HOA 6 finished	2 weeks a
	ansible.cfg	HOA 6 finished	2 weeks a
	install_apache.yml	HOA 7 Part 1	48 minutes a
	install_apache_redhat.yml	HOA 5 final	3 weeks a
	inventory.ini	HOA 7 final	n
	site.yml	HOA 7 final	n
<a href="#">View all files</a>			

 README

*also just pushed it just in case*

### Reflections:

Answer the following:

1. What is the importance of creating roles?

*Roles are ultimately a set of goals assigned to a person or a group of people. It is essential to a great team. This is the same for a group of remote servers. They only need certain files for certain needs. Putting all plays into one big file ultimately causes problems like having to scroll endlessly trying to find that one error (true story) but splitting it into multiple directories makes it more manageable and makes it more efficient in finding errors.*

2. What is the importance of managing files?

*An operating system's greatest supply are files, they are the ones that keep the hardware and software going. Managing files remotely not only allows for better control of the server, but also just makes it easier if done correctly (unlike putting it all in one folder called "to be sorted later")*

