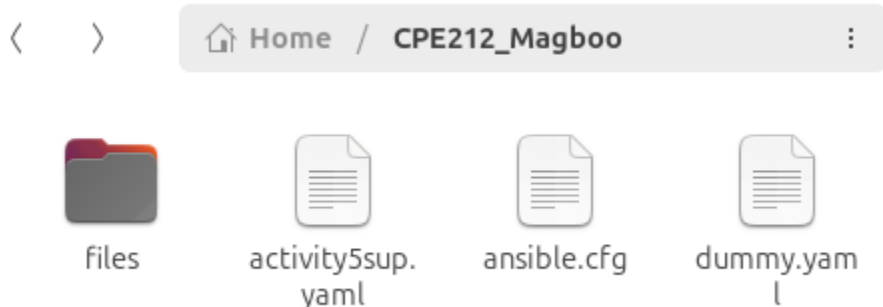
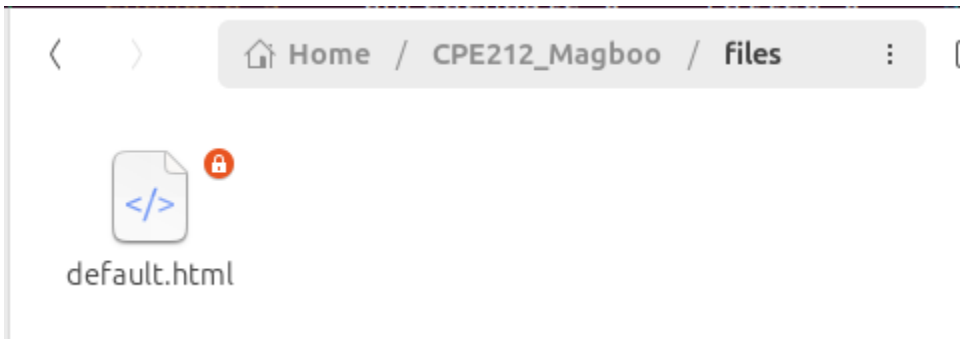


Name: Magboo, Matt Clemence C.	Date Performed: 10/03/2025
Course/Section: CPE212 - CPE31S2	Date Submitted: 10/03/2025
Instructor: Engr. Robin Valenzuela	Semester and SY: 2025 - 2026
Activity 7: Managing Files and Creating Roles in Ansible	
1. Objectives: 1.1 Manage files in remote servers 1.2 Implement roles in ansible	
2. Discussion: <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 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.</p>	
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.	
	
	
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

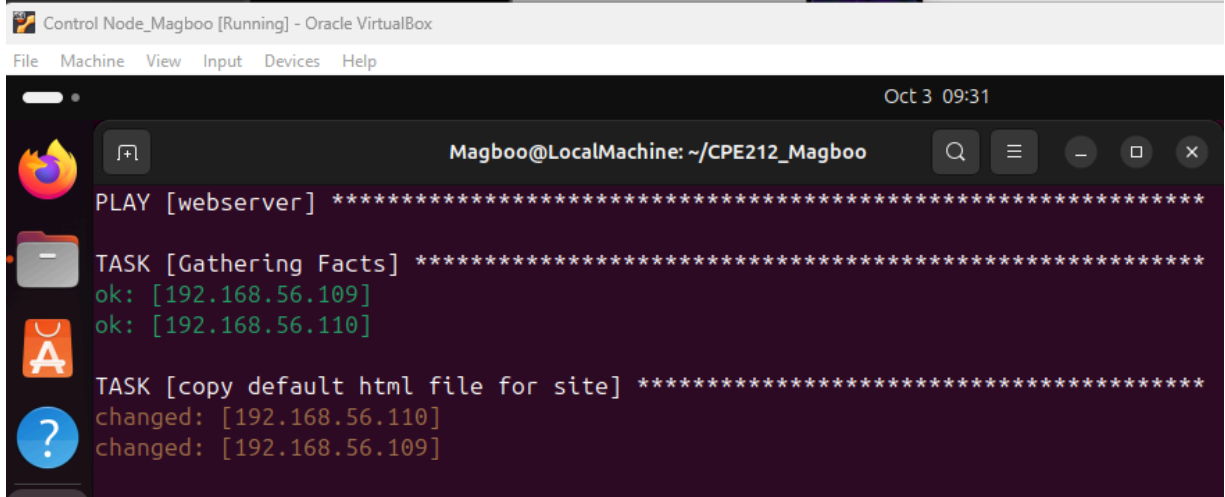
owner: root

group: root

mode: 0644

```
- hosts: webserver
  become: true
  tasks:
  - name: copy default html file for site
    tags: apache, apache2, httpd
    copy:
      src: /home/Magboo/CPE212_Magboo/files/default.html
      dest: /var/www/html/index.html
      owner: root
      group: root
      mode: 0644
    become: True
```

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



The screenshot shows a terminal window titled "Control Node_Magboo [Running] - Oracle VirtualBox". The terminal output displays the execution of an Ansible playbook named "site.yml". The output is as follows:

```
PLAY [webserver] *****
TASK [Gathering Facts] *****
ok: [192.168.56.109]
ok: [192.168.56.110]
TASK [copy default html file for site] *****
changed: [192.168.56.110]
changed: [192.168.56.109]
```

it copied the file default.html in manage node for the web_servers.

```

Magboo@LocalMachine: ~/CPE212_Magboo/files
ok: [192.168.56.110]

PLAY [file_servers] *****

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

TASK [install samba package] *****
changed: [192.168.56.105]

PLAY RECAP *****
192.168.56.105      : ok=7    changed=3    unreachable=0    failed=0    s
skipped=2    rescued=0    ignored=0
192.168.56.109      : ok=8    changed=2    unreachable=0    failed=0    s
skipped=4    rescued=0    ignored=0
192.168.56.110      : ok=8    changed=2    unreachable=0    failed=0    s
skipped=4    rescued=0    ignored=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.

Magboo@Server1: /var/www/html

Magboo@Server1:~\$ ls

Desktop Documents Downloads Music Pictures Public snap Templates Videos

Magboo@Server1:~\$ cd var

bash: cd: var: No such file or directory

Magboo@Server1:~\$ cd /var

Magboo@Server1:/var\$ cd /www

bash: cd: /www: No such file or directory

Magboo@Server1:/var\$ cd www

Magboo@Server1:/var/www\$ cd html

Magboo@Server1:/var/www/html\$ cat index.html

<!DOCTYPE html>
<html lang="eng">
<head>
 <meta charset="utf-8">
 <title>Default Page</title>
</head>
<body>
 <h1>Welcome to the Default Page</h1>
 <p>This is a simple HTML document.</p>
</body>
</html>

Magboo@Server1:/var/www/html\$ S

Default Page

← → ↻ 192.168.56.105

CentOS Blog Documentation Forums

Welcome to the Default Page

This is a simple HTML document.

it what's inside of the default.html in the manage node then created its own file as index.html

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

```
Magboo@LocalMachine:~/CPE212_Magboo$ git commit -m "act 7"
[main b98a40d] act 7
 2 files changed, 20 insertions(+)
 create mode 100644 files/default.html
Magboo@LocalMachine:~/CPE212_Magboo$ git push origin main
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 6 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 713 bytes | 713.00 KiB/s, done.
Total 5 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To github.com:MagbooMattClemence/CPE212_Magboo.git
 e71b14c..b98a40d  main -> main
```

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_a
md64.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.

```
Magboo@LocalMachine: ~/CPE212_Magboo
GNU nano 7.2 inventory.yaml
[workstations]
192.168.56.109
192.168.56.110
192.168.56.105
[webserver]
192.168.56.109
192.168.56.110
192.168.56.105 ansible_user=matt ansible_ssh_private_keyfile=~/.ssh/id_rsa
[file_servers]
192.168.56.105 ansible_user=matt ansible_ssh_private_key_file=~/.ssh/id_rsa
```

3. Run the playbook. Describe the output.

```
TASK [install terraform] *****
changed: [192.168.56.109]
changed: [192.168.56.105]
changed: [192.168.56.110]
```

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

```
Magboo@Server2: ~  
Magboo@Server2:~$ 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  
  login          Obtain and save credentials for a remote host  
  logout         Remove locally-stored credentials for a remote host  
  output         Read an output from a state file  
  plan           Generate and show an execution plan  
  providers      Prints a tree of the providers used in the configuration  
  refresh        Update local state file against real resources  
  show           Inspect Terraform state or plan  
  taint          Manually mark a resource for recreation  
  untaint        Manually unmark a resource as tainted  
  validate       Validates the Terraform files  
  version        Prints the Terraform version  
  workspace      Workspace management
```

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.

```
Magboo@LocalMachine:~/CPE212_Magboo/roles$ mkdir -p {base,webservers,file_servers,db_servers,workstations}/tasks
Magboo@LocalMachine:~/CPE212_Magboo/roles$ ls base
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.

```
Magboo@LocalMachine:~/CPE212_Magboo/roles/base/tasks$ ls
main.yml
```

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

```
Magboo@LocalMachine: ~/CPE212_Magboo
PLAY [webservers] *****
skipping: no hosts matched
[WARNING]: Could not match supplied host pattern, ignoring: db_servers

PLAY [db_servers] *****
skipping: no hosts matched

PLAY [file_servers] *****

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

TASK [file_servers : install samba package] *****
ok: [192.168.56.105]

PLAY RECAP *****
192.168.56.105      : ok=9    changed=0    unreachable=0    failed=0    s
kipped=2    rescued=0    ignored=0
192.168.56.109      : ok=7    changed=0    unreachable=0    failed=0    s
kipped=2    rescued=0    ignored=0
192.168.56.110      : ok=7    changed=0    unreachable=0    failed=0    s
kipped=2    rescued=0    ignored=0
```

Reflections:

Answer the following:

1. What is the importance of creating roles?
it automates tasks that is specified for them
2. What is the importance of managing files?

for you to find the files fast and fix problems quickly you can also share the work with others but organized