

Name: Dalena, Justin Miguel S.	Date Performed: Oct. 11, 2024
Course/Section: CPE31S21	Date Submitted: Oct. 11, 2024
Instructor: Engr. Robin Valenzuela	Semester and SY: 2024-25
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. <pre> justin@workstation:~/CPE212_Dalena\$ ls ansible.cfg files install_apache.yml inventory README.md site.retry site.yml justin@workstation:~/CPE212_Dalena\$ cd files justin@workstation:~/CPE212_Dalena/files\$ ls default_site.html justin@workstation:~/CPE212_Dalena/files\$ </pre>	
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: <ul style="list-style-type: none"> - name: copy default html file for site <pre> tags: apache, apache2, httpd copy: src: default_site.html dest: /var/www/html/index.html owner: root group: root mode: 0644 </pre> 	

```
- name: install apache and php for CentOS servers
  yum:
    name:
      - httpd
      - php
    state: latest
  when: ansible_distribution is defined and ansible_distribution == "CentOS"
```

```
- 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: db_servers
  become: true
  tasks:

  - name: install mariadb package (CentOS)
    yum:
```

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

```
[WARNING]: Module invocation had junk after the JSON data: AttributeError("module 'platform' has no attribute 'dist'")
ok: [server2]

TASK [install updates (CentOS)] *****
skipping: [Server1]
skipping: [server2]
skipping: [Server3]

TASK [install updates (Ubuntu)] *****
skipping: [server2]
ok: [Server3]
ok: [Server1]

PLAY [web_servers] *****

TASK [Gathering Facts] *****
ok: [Server1]

TASK [install apache and php for Ubuntu servers] *****
ok: [Server1]

TASK [install apache and php for CentOS servers] *****
skipping: [Server1]

TASK [copy default html file for site] *****
changed: [Server1]

PLAY [db_servers] *****

TASK [Gathering Facts] *****
[WARNING]: Module invocation had junk after the JSON data: AttributeError("module 'platform' has no attribute 'dist'")
ok: [server2]

TASK [install mariadb package (CentOS)] *****
skipping: [server2]

TASK [Mariadb Restarting/Enabling] *****
fatal: [server2]: FAILED! => ['changed': false, 'msg': 'Could not find the requested service mariadb: host']
to retry, use: --limit @/home/justin/CPE212_Dalena/site.retry

PLAY RECAP *****
Server1      : ok=5    changed=1    unreachable=0    failed=0
Server3      : ok=2    changed=0    unreachable=0    failed=0
server2      : ok=2    changed=0    unreachable=0    failed=1
```

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.

```
200 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

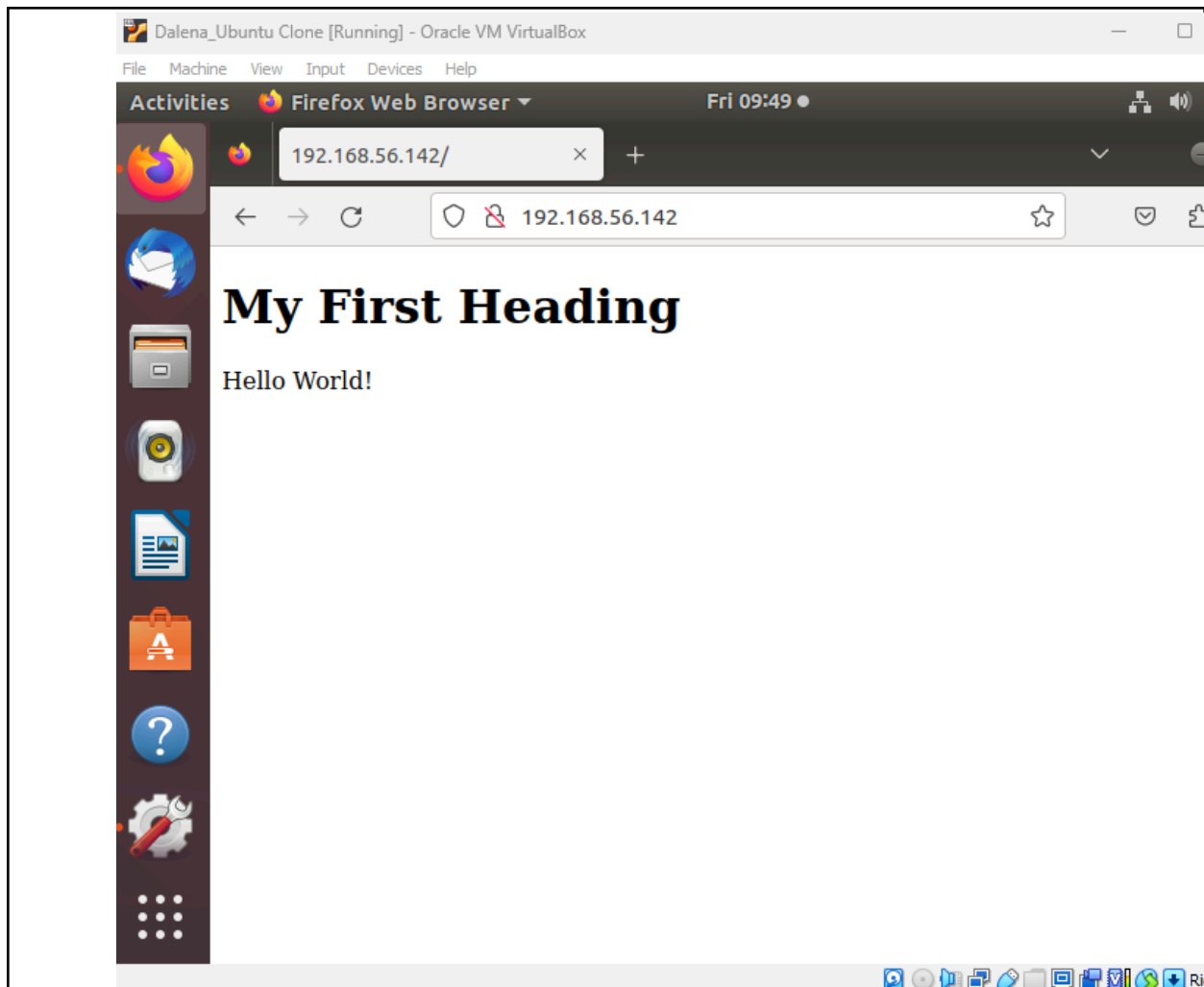
New release '20.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Your Hardware Enablement Stack (HWE) is supported until April 20
Last login: Fri Oct 11 09:46:57 2024 from 192.168.56.141
justin@Server1:~$ cat /var/www/html/index.html
<!DOCTYPE html>
<html>
<body>

<h1>My First Heading </h1>

<p>Hello World!</p>

</body>
</html>
justin@Server1:~$
```



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

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

```

PLAY [workstations] *****

TASK [Gathering Facts] *****
ok: [Server1]

TASK [install unzip] *****
ok: [Server1]

TASK [install terraform] *****
changed: [Server1]

PLAY [web_servers] *****

TASK [Gathering Facts] *****
ok: [Server1]

TASK [install apache and php for Ubuntu servers] *****
ok: [Server1]

TASK [install apache and php for CentOS servers] *****
skipping: [Server1]

TASK [copy default html file for site] *****
ok: [Server1]

PLAY [db_servers] *****

TASK [Gathering Facts] *****
[WARNING]: Module invocation had junk after the JSON data: AttributeError
ok: [server2]

TASK [install mariadb package (CentOS)] *****
skipping: [server2]

TASK [Mariadb Restarting/Enabling] *****
fatal: [server2]: FAILED! => {"changed": false, "msg": "Could not find t
to retry, use: --limit @/home/justin/CPE212_Dalena/site.retry

PLAY RECAP *****
Server1          : ok=8    changed=1    unreachable=0    failed=0
Server3          : ok=2    changed=0    unreachable=0    failed=0
server2          : ok=2    changed=0    unreachable=0    failed=0

justin@workstation:~/CPE212_Dalena$ S

```

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

```
justin@Server1:~$ terraform --version
Terraform v0.12.28

Your version of Terraform is out of date! The latest version
is 1.9.7. You can update by downloading from https://www.terraform.io/down
.html
```

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.

```
justin@workstation:~/CPE212_Dalena$ mkdir roles
justin@workstation:~/CPE212_Dalena$ cd roles
justin@workstation:~/CPE212_Dalena/roles$ mkdir base, web_servers, file_servers, db_s
justin@workstation:~/CPE212_Dalena/roles$ ls
base, db_servers file_servers, web_servers,
justin@workstation:~/CPE212_Dalena/roles$ mkdir worstations
justin@workstation:~/CPE212_Dalena/roles$ rmdir worstations
justin@workstation:~/CPE212_Dalena/roles$ ls
base, db_servers file_servers, web_servers,
justin@workstation:~/CPE212_Dalena/roles$ mkdir workstations
justin@workstation:~/CPE212_Dalena/roles$ ls
base, db_servers file_servers, web_servers, workstations
justin@workstation:~/CPE212_Dalena/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.
4. Run the site.yml playbook and describe the output.

Reflections:

Answer the following:

1. What is the importance of creating roles?
 - provide a structured way to organize tasks, templates, files, and variables.
2. What is the importance of managing files?
 - offers flexibility and control over remote systems among its many features.