Name: Robin E. Valenzuela	Date Performed:10/15/2022
Course/Section:CPE31S2	Date Submitted:
Instructor: Engr. Jonathan Taylar	Semester and SY:

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.

```
valenzuela@workstation:~/CPE232_Valenzuela$ mkdir files
valenzuela@workstation:~/CPE232_Valenzuela$ cd files
valenzuela@workstation:~/CPE232_Valenzuela/files$ nano default_site.html
valenzuela@workstation:~/CPE232_Valenzuela/files$
```

```
valenzuela@workstation:~/CPE232_Valenzuela/files$ cat default_site.html
<html>
        <title> My HTML </title>

        <body>
             <p> This is what Ansible Looks like </p>
        </body>
</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
```

```
- 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.

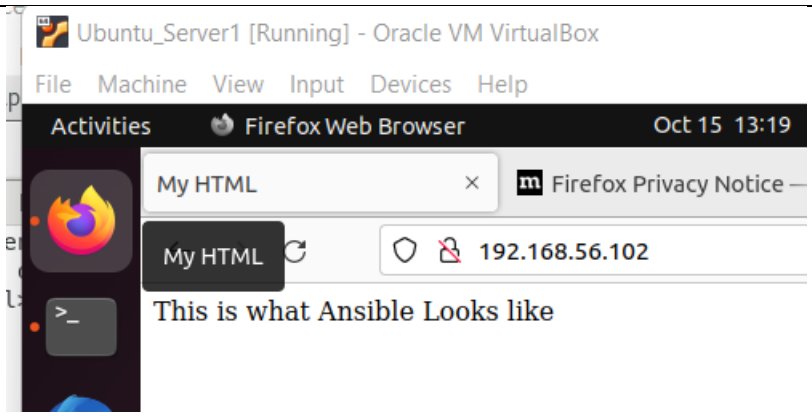
```
TASK [copy default html file for site] **
*
ok: [192.168.56.102]
changed: [192.168.56.105]
```

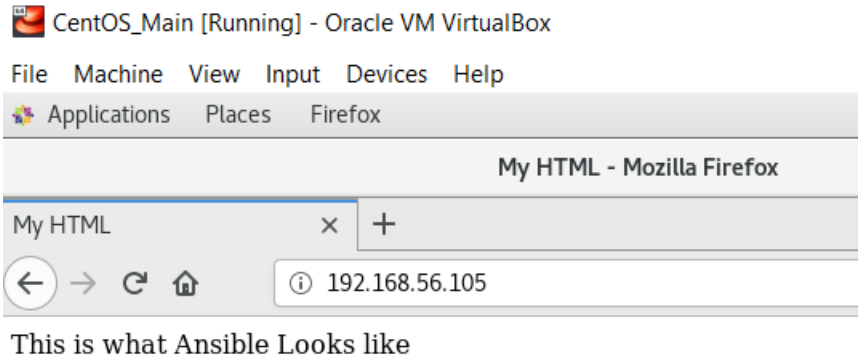
- The changes that happened is that we copied default_site.html

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.

```
valenzuela@server1:~$ cat default_site.html /var/www/html/index.html
cat: default_site.html: No such file or directory
<html>
        <title> My HTML </title>


        <body>
             <p> This is what Ansible Looks like </p>
        </body>
</html>
valenzuela@server1:~$
```

```
[valenzuela@localhost ~]$ cat default_site.html /var/www/html/index.html
cat: default_site.html: No such file or directory
<html>
        <title> My HTML </title>


        <body>
             <p> This is what Ansible Looks like </p>
        </body>
</html>
```





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

```
valenzuela@workstation:~/CPE232_Valenzuela$ git add files
valenzuela@workstation:~/CPE232_Valenzuela$ git commit -m "html"
[main 9862417] html
 1 file changed, 8 insertions(+)
 create mode 100644 files/default_site.html
valenzuela@workstation:~/CPE232_Valenzuela$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 414 bytes | 414.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:qrevalenzuela/CPE232_Valenzuela.git
   cc09610..9862417  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_amd64.zip
    dest: /usr/local/bin
    remote_src: yes
    mode: 0755
    owner: root
    group: root

```
- 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_l
inux_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.

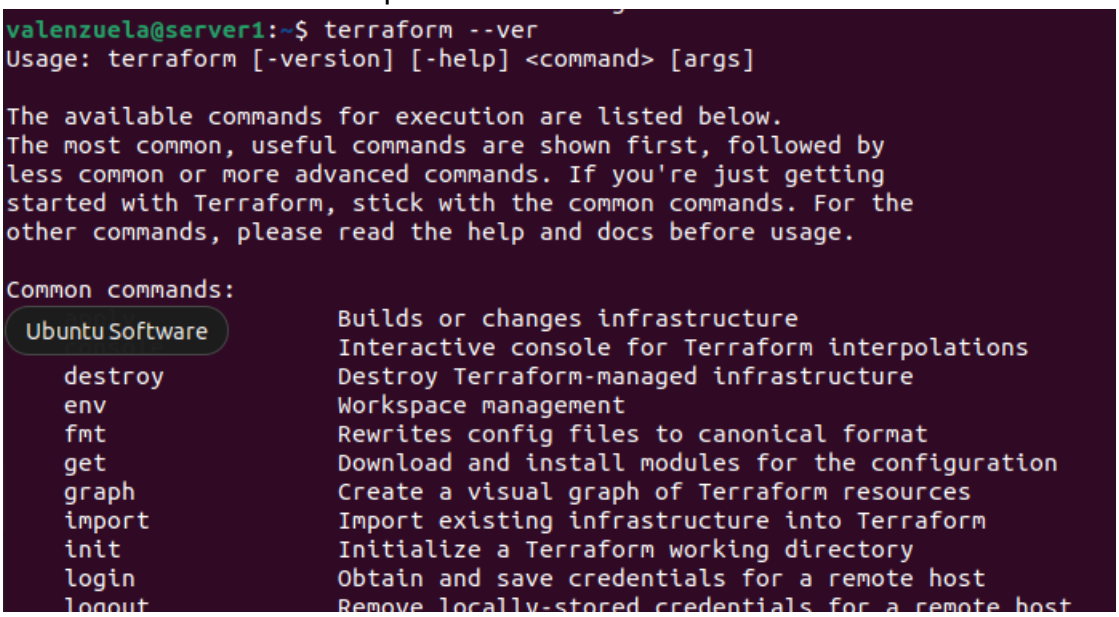
```
  GNU nano 6.2                                inventory *
[web_servers]
192.168.56.102
192.168.56.105

[db_servers]
192.168.56.105

[file_servers]
192.168.56.102

[workstations]
192.168.56.102
```

3. Run the playbook. Describe the output.

```
TASK [install terraform] **
*
changed: [192.168.56.102]
```

- The workstation has successfully installed terraform

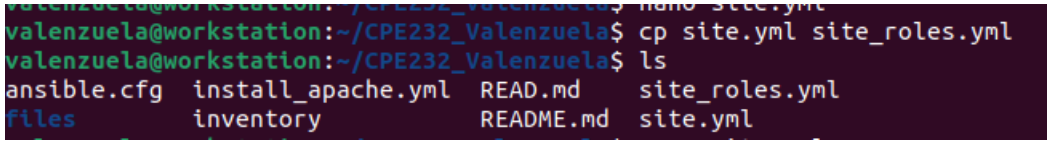
4. On the Ubuntu remote workstation, type terraform to verify installation of terraform. Describe the output.



- The output of the terraform command is how to use the terraform

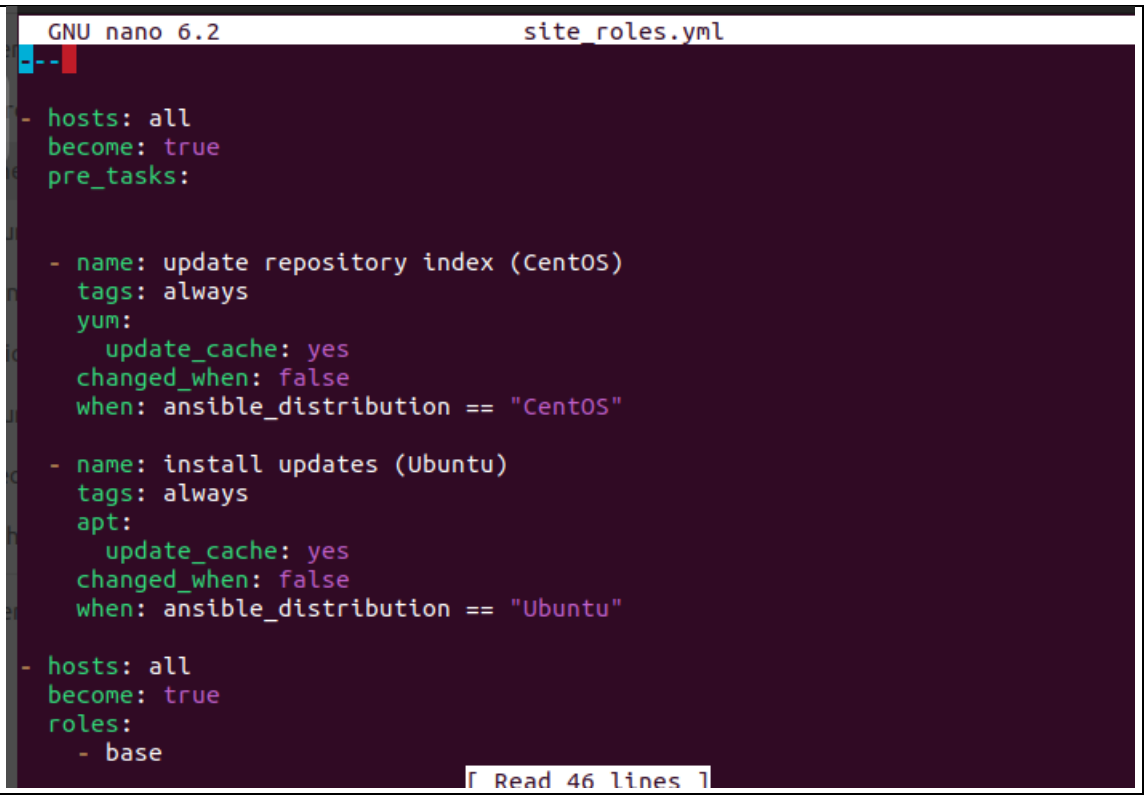
Task 3: Create roles

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)



2.

3.


GNU nano 6.2
site_roles.yml

- hosts: all
become: true
pre_tasks:
- name: update repository index (CentOS)
tags: always
yum:
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
[Read 46 lines]

```
---
- 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.

4. 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.

```
valenzuela@workstation:~/CPE232_Valenzuela$ ls
ansible.cfg  install_apache.yml  READ.md    roles           site.yml
files        inventory           README.md  site_roles.yml
valenzuela@workstation:~/CPE232_Valenzuela/roles$ ls
base  db_servers  file_servers  web_servers  workstations
valenzuela@workstation:~/CPE232_Valenzuela/roles$ ls base
tasks
valenzuela@workstation:~/CPE232_Valenzuela/roles$ s
valenzuela@workstation:~/CPE232_Valenzuela/roles$ mkdir base/tasks
valenzuela@workstation:~/CPE232_Valenzuela/roles$ mkdir web_servers/tasks
valenzuela@workstation:~/CPE232_Valenzuela/roles$ mkdir file_servers/tasks
valenzuela@workstation:~/CPE232_Valenzuela/roles$ mkdir db_servers/tasks
valenzuela@workstation:~/CPE232_Valenzuela/roles$ mkdir workstations/tasks
```

5. 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.

```
  GNU nano 6.2                          main.yml *
- 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: "Mariadb - Restarting/Enabling"
    service:
      name: mariadb
      state: restarted
      enabled: true

  - name: install mariadb package (Ubuntu)
    tags: db, mariadb,ubuntu
    apt:
      name: mariadb-server
      state: latest
    when: ansible_distribution == "Ubuntu"
```

```
 GNU nano 6.2                                main.yml *
- 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_>
      dest: /usr/local/bin
      remote_src: yes
      mode: 0755
      owner: root
      group: root
```

```
 GNU nano 6.2                                main.yml *
- hosts: file_servers
  become: true
  tasks:

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

```
valenzuela@workstation:~/CPE232_Valenzuela$ cd roles
valenzuela@workstation:~/CPE232_Valenzuela/roles$ cd web_servers
valenzuela@workstation:~/CPE232_Valenzuela/roles/web_servers$ cd tasks
valenzuela@workstation:~/CPE232_Valenzuela/roles/web_servers/tasks$ nano main.y
ml
valenzuela@workstation:~/CPE232_Valenzuela/roles/web_servers/tasks$ cd ..
valenzuela@workstation:~/CPE232_Valenzuela/roles/web_servers$ cd ..
valenzuela@workstation:~/CPE232_Valenzuela/roles$ cd db_servers
valenzuela@workstation:~/CPE232_Valenzuela/roles/db_servers$ cd tasks
valenzuela@workstation:~/CPE232_Valenzuela/roles/db_servers/tasks$ nano main.ym
l
valenzuela@workstation:~/CPE232_Valenzuela/roles/db_servers/tasks$ cd
valenzuela@workstation:~$ cd CPE232_Valenzuela/roles
valenzuela@workstation:~/CPE232_Valenzuela/roles$ cd tasks
bash: cd: tasks: No such file or directory
valenzuela@workstation:~/CPE232_Valenzuela/roles$ ls
base  db_servers  file_servers  web_servers  workstations
valenzuela@workstation:~/CPE232_Valenzuela/roles$ cd workstations
valenzuela@workstation:~/CPE232_Valenzuela/roles/workstations$ ls
tasks
valenzuela@workstation:~/CPE232_Valenzuela/roles/workstations$ cd tasks
valenzuela@workstation:~/CPE232_Valenzuela/roles/workstations/tasks$ nano main.
yml
```

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

```
TASK [install updates (Ubuntu)] ************************************************
*
skipping: [192.168.56.105]
ok: [192.168.56.102]

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

TASK [Gathering Facts] *********************************************************
*
ok: [192.168.56.105]
ok: [192.168.56.102]

TASK [base : adding of ssh] ****************************************************
*
changed: [192.168.56.105]
changed: [192.168.56.102]

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

TASK [Gathering Facts] *********************************************************
*
ok: [192.168.56.102]

TASK [workstations : install unzip] ********************************************
*
ok: [192.168.56.102]
```

- The output of this is that the playbook ran because of the roles.

Reflections:

Answer the following:

1. What is the importance of creating roles?
    - Creating roles will create reusable automatic components so that we can use it over and over again without ever having to create the whole command again, the system administrator just needs to call the role that is needed.
2. What is the importance of managing files?
    - The importance of managing files is make things easier and makes it minimal for the system administrator to locate files.