Name: Escobilla, Davonn P.	Date Performed: 15/10/2022
Course/Section: CPE31S24	Date Submitted: 15/10/2022
Instructor: Dr. Jonathan Taylar	Semester and SY: 1st Sem, 2022

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.

```
davonn@workstation:~$ cd CPE232_Escobilla
davonn@workstation:~/CPE232_Escobilla$ mkdir files
davonn@workstation:~/CPE232_Escobilla$ ls
ansible.cfg  files  install_apache.yml  inventory  README.md  site.yml
```

```
davonn@workstation: ~/CPE232_Escobilla/files
  GNU nano 4.8                    default_site.html
<html>
  <title>Web Test</title>
  <body>
   Hello World!
  </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

```
davonn@workstation: ~/CPE232_Escobilla

GNU nano 4.8                                site.yml
    dnf:
      name:
        - httpd
        - php
      state: latest
    when: ansible_distribution == "CentOS"

  - name: start httpd (CentOS)
    tags: apache, centos,httpd
    service:
      name: httpd
      state: started
    when: ansible_distribution == "CentOS"

  - name: copy default html for site
    tags: apacache, 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.

```
davonn@workstation: ~/CPE232_Escobilla
davonn@workstation:~$ cd CPE232_Escobilla
davonn@workstation:~/CPE232_Escobilla$ ansible-playbook --ask-become-pass site.
yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************
*
ok: [192.168.56.103]
ok: [192.168.56.105]

TASK [install updates (CentOS)] ************************************************
*
skipping: [192.168.56.103]
ok: [192.168.56.105]

TASK [install updates (Ubuntu)] ************************************************
*
skipping: [192.168.56.105]
ok: [192.168.56.103]

PLAY [web_servers] *************************************************************
*

TASK [Gathering Facts] *********************************************************
*
ok: [192.168.56.103]
```

davonn@workstation: ~/CPE232_Escobilla

```
ok: [192.168.56.105]

TASK [install apache and php for Ubuntu servers] ******************************
*
skipping: [192.168.56.105]
ok: [192.168.56.103]

TASK [install apache and php for CentOS servers] ******************************
*
skipping: [192.168.56.103]
ok: [192.168.56.105]

TASK [start httpd (CentOS)] ***************************************************
*
skipping: [192.168.56.103]
ok: [192.168.56.105]

TASK [copy default html for site] *********************************************
*
ok: [192.168.56.105]
changed: [192.168.56.103]

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

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

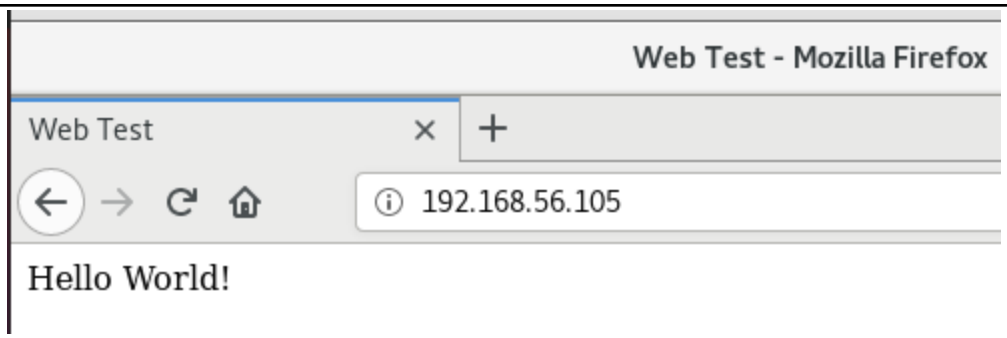
```
davonn@workstation: ~/CPE232_Escobilla
ok: [192.168.56.105]

TASK [install mariadb package (Ubuntu)] ************************************
*
skipping: [192.168.56.105]

TASK [Mariadb- Restarting/Enabling] ****************************************
*
changed: [192.168.56.105]

PLAY [file_servers] ********************************************************
*

TASK [Gathering Facts] *****************************************************
*
ok: [192.168.56.103]

TASK [install samba package] ***********************************************
*
changed: [192.168.56.103]

PLAY RECAP *****************************************************************
*
192.168.56.103             : ok=7    changed=2    unreachable=0    failed=0
skipped=3    rescued=0    ignored=0
192.168.56.105             : ok=9    changed=1    unreachable=0    failed=0
skipped=3    rescued=0    ignored=0
```

In the web_servers both remote servers have changed since we entered a new task about copying the default html for the site.

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.

```
davonn@server1:~$ cat /var/www/html/index.html
<html>
  <title>Web Test</title>
  <body>
   Hello World!
  </body>
</html>
```

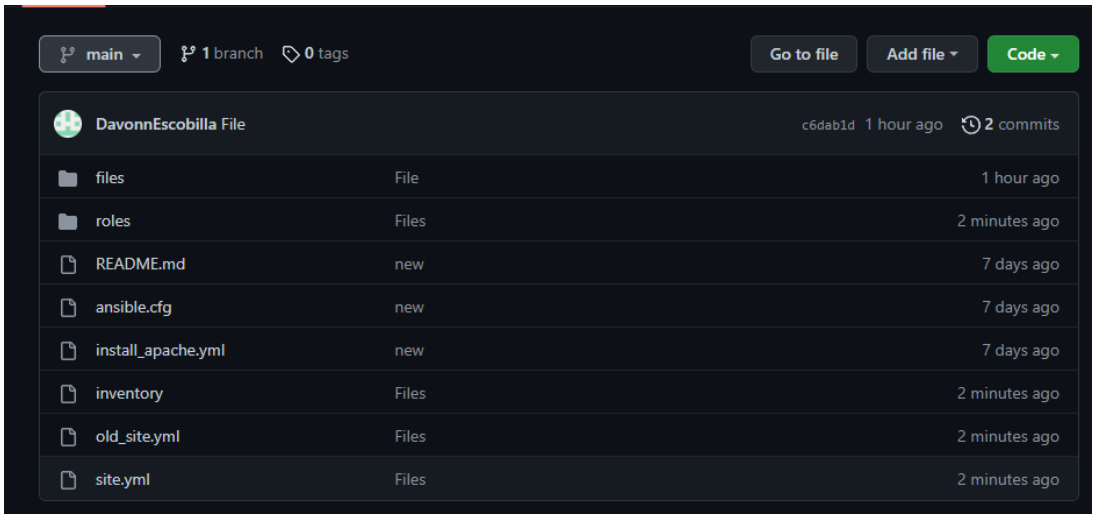
```
[davonn@localhost ~]$ cat /var/www/html/index.html
<html>
  <title>Web Test</title>
  <body>
   Hello World!
  </body>
</html>
```



The inputs from the default_site.html in the playbook applied to the remote servers, in the CentOS upon typing the ip address the output is Hello World! as it is the input in the html.

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

```
davonn@workstation:~/CPE232_Escobilla$ git add -A
davonn@workstation:~/CPE232_Escobilla$ git commit -m "File"
[main c6dab1d] File
 3 files changed, 22 insertions(+), 7 deletions(-)
 create mode 100644 files/default_site.html
davonn@workstation:~/CPE232_Escobilla$ git push
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Compressing objects: 100% (5/5), done.
Writing objects: 100% (6/6), 782 bytes | 782.00 KiB/s, done.
Total 6 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:DavonnEscobilla/CPE232_Escobilla.git
   64a145b..c6dab1d  main -> main
```



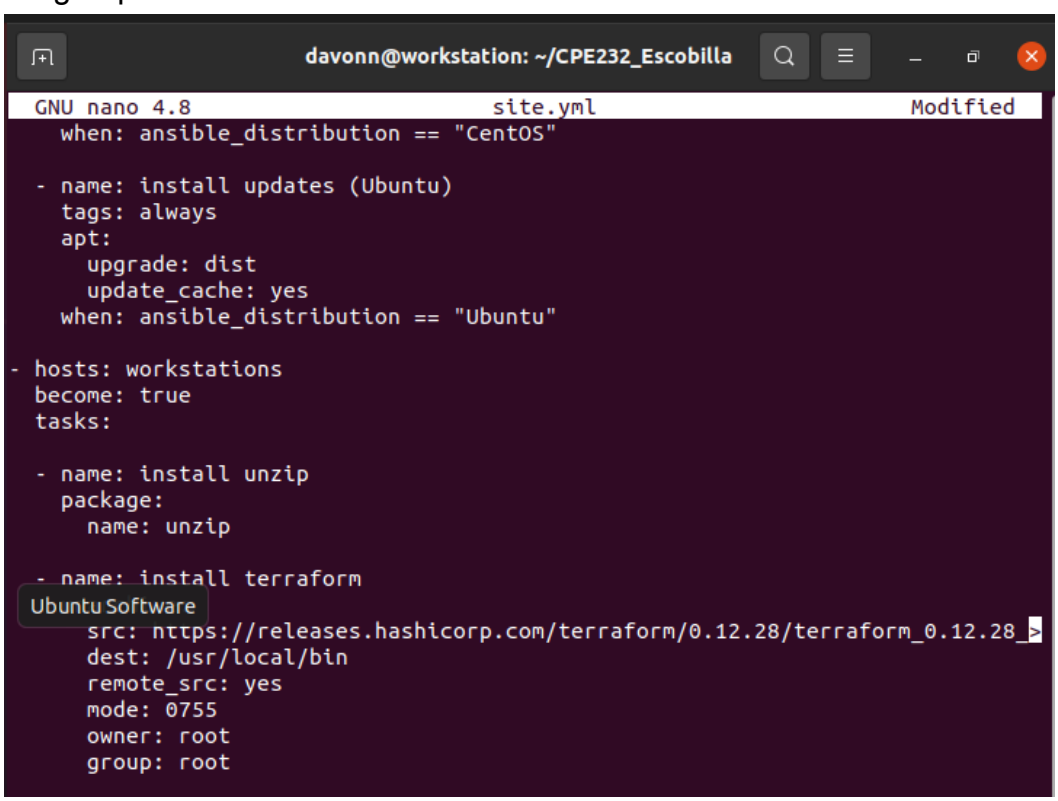
The added files are present on the github interface with updated time and content.

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.

```
davonn@workstation: ~/CPE232_Escobilla

TASK [install updates (CentOS)] *************************
*
skipping: [192.168.56.103]
ok: [192.168.56.105]

TASK [install updates (Ubuntu)] *************************
*
skipping: [192.168.56.105]
ok: [192.168.56.103]

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

TASK [Gathering Facts] **********************************
*
ok: [192.168.56.103]

TASK [install unzip] ************************************
*
ok: [192.168.56.103]

TASK [install terraform] ********************************
*
changed: [192.168.56.103]
```

The task created on a new play on site.yml is successful since the status becomes changed prior to the address inside the workstation.

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

```
davonn@server1: ~
  </body>
</html>
davonn@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
    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
```

Terraform is installed since the common commands appeared for guidance.

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)

```yaml
---
- 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
```

```
davonn@workstation: ~/CPE232_Escobilla
  GNU nano 4.8                    site.yml                    Modified
---

- hosts: all
  become: true
  tasks:
  pre_tasks:

  - name: update repository index (CentOS)
    tags: always
    yum:
      update_only: yes
    changed_when: false
    when: ansible_distribution == "CentOS"

  - name: install update (Ubuntu)
    tags: always
    apt:
      update_cache: yes
    changed_when: false
    when: ansible_distribution == "Ubuntu"

- hosts: all
  become: true
  roles:
    - base
```

```
davonn@workstation: ~/CPE232_Escobilla
  GNU nano 4.8                    site.yml                    Modified
- 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.

```
davonn@workstation: ~/CPE232_Escobilla/roles
davonn@workstation:~/CPE232_Escobilla$ mkdir roles
davonn@workstation:~/CPE232_Escobilla$ cd roles
davonn@workstation:~/CPE232_Escobilla/roles$ mkdir base
davonn@workstation:~/CPE232_Escobilla/roles$ mkdir web_servers
davonn@workstation:~/CPE232_Escobilla/roles$ mkdir file_servers
davonn@workstation:~/CPE232_Escobilla/roles$ mkdir db_servers
davonn@workstation:~/CPE232_Escobilla/roles$ mkdir workstations
davonn@workstation:~/CPE232_Escobilla/roles$ cd base
davonn@workstation:~/CPE232_Escobilla/roles/base$ mkdir tasks
davonn@workstation:~/CPE232_Escobilla/roles/base$ cd ..
davonn@workstation:~/CPE232_Escobilla/roles$ cd web_servers
davonn@workstation:~/CPE232_Escobilla/roles/web_servers$ mkdir tasks
davonn@workstation:~/CPE232_Escobilla/roles/web_servers$ cd ..
davonn@workstation:~/CPE232_Escobilla/roles$ cd file_servers
davonn@workstation:~/CPE232_Escobilla/roles/file_servers$ mkdir tasks
davonn@workstation:~/CPE232_Escobilla/roles/file_servers$ cd ..
davonn@workstation:~/CPE232_Escobilla/roles$ cd db_servers
davonn@workstation:~/CPE232_Escobilla/roles/db_servers$ mkdir tasks
davonn@workstation:~/CPE232_Escobilla/roles/db_servers$ cd ..
davonn@workstation:~/CPE232_Escobilla/roles$ cd mkdir
bash: cd: mkdir: No such file or directory
davonn@workstation:~/CPE232_Escobilla/roles$ cd workstations
davonn@workstation:~/CPE232_Escobilla/roles/workstations$ mkdir tasks
davonn@workstation:~/CPE232_Escobilla/roles/workstations$ cd
davonn@workstation:~$ cd CPE232_Escobilla
davonn@workstation:~/CPE232_Escobilla$ cd roles
davonn@workstation:~/CPE232_Escobilla/roles$ ls
base  db_servers  file_servers  web_servers  workstations
```

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.

davonn@workstation: ~/CPE232_Escobilla/roles/base

```
  GNU nano 4.8                          main.yml
---

- hosts: all
  become: true
  tasks:
  pre_tasks:

  - name: install updates (CentOS)
    tags: always
    yum:
      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"
```

davonn@workstation: ~/CPE232_Escobilla/roles/workstations

```
  GNU nano 4.8                    main.yml                        Modified
---

- 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
```

davonn@workstation: ~/CPE232_Escobilla/roles/web_servers

```
  GNU nano 4.8                    main.yml                        Modified
---

- hosts: web_servers
  become: true
  tasks:

  - 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
    yum:
      name:
        - httpd
        - php
      state: latest
    when: ansible_distribution == "CentOS"
```

davonn@workstation: ~/CPE232_Escobilla/roles/db_servers

```
  GNU nano 4.8                          main.yml
---

- hosts: db_servers
  become: true
  tasks:

  - name: install mariadb package (CentOS)
    tags: centos, db,maria,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
```

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

```
davonn@workstation: ~/CPE232_Escobilla
*

TASK [Gathering Facts] ***********************************************************
*
ok: [192.168.56.103]
ok: [192.168.56.105]

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

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

PLAY [file_servers] **************************************************************
*

TASK [Gathering Facts] ***********************************************************
*
ok: [192.168.56.103]

PLAY RECAP ***********************************************************************
*
192.168.56.103             : ok=6    changed=0    unreachable=0    failed=0
skipped=1    rescued=0    ignored=0
192.168.56.105             : ok=5    changed=0    unreachable=0    failed=0
skipped=1    rescued=0    ignored=0

davonn@workstation:~/CPE232_Escobilla$
```

The tasks assigned for each play in each server from the inventory is successfully executed with no errors.

Reflections:

Answer the following:

1. What is the importance of creating roles?
   Creating roles are for convenient management of each task in every play. It creates minimal configuration hassle when it comes to future problems that might occur for each task in the play.
2. What is the importance of managing files?
   Managing files helps you to create a healthy working environment since you do not have to scramble and find the specific file you are looking for, especially in important situations. In this activity, it is easier to find and navigate your target file if it is managed neatly.