Name: Maxine Audrey D. Pulao	Date Performed: October 11, 2022
Course/Section: CPE31S2	Date Submitted:
Instructor: Dr. Jonathan Taylar	Semester and SY: 2022-2023

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.

```
aud@rey:~/ansible$ mkdir files
aud@rey:~/ansible$ ls
ansible.cfg  files  install_apache.yml  inventory  site.yml
aud@rey:~/ansible$ cd files
aud@rey:~/ansible/files$ sudo nano default_site.html
```

```
aud@rey:~/ansible/files$ cat default_site.html
<!DOCTYPE html>
<html>
  <head>
    <title>Managing Enterprise</title>
  </head>
  <body>
    <h1>CPE_232 Managing Enterprise</h1>
    <p>by: Maxine Audrey D. Pulao</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:

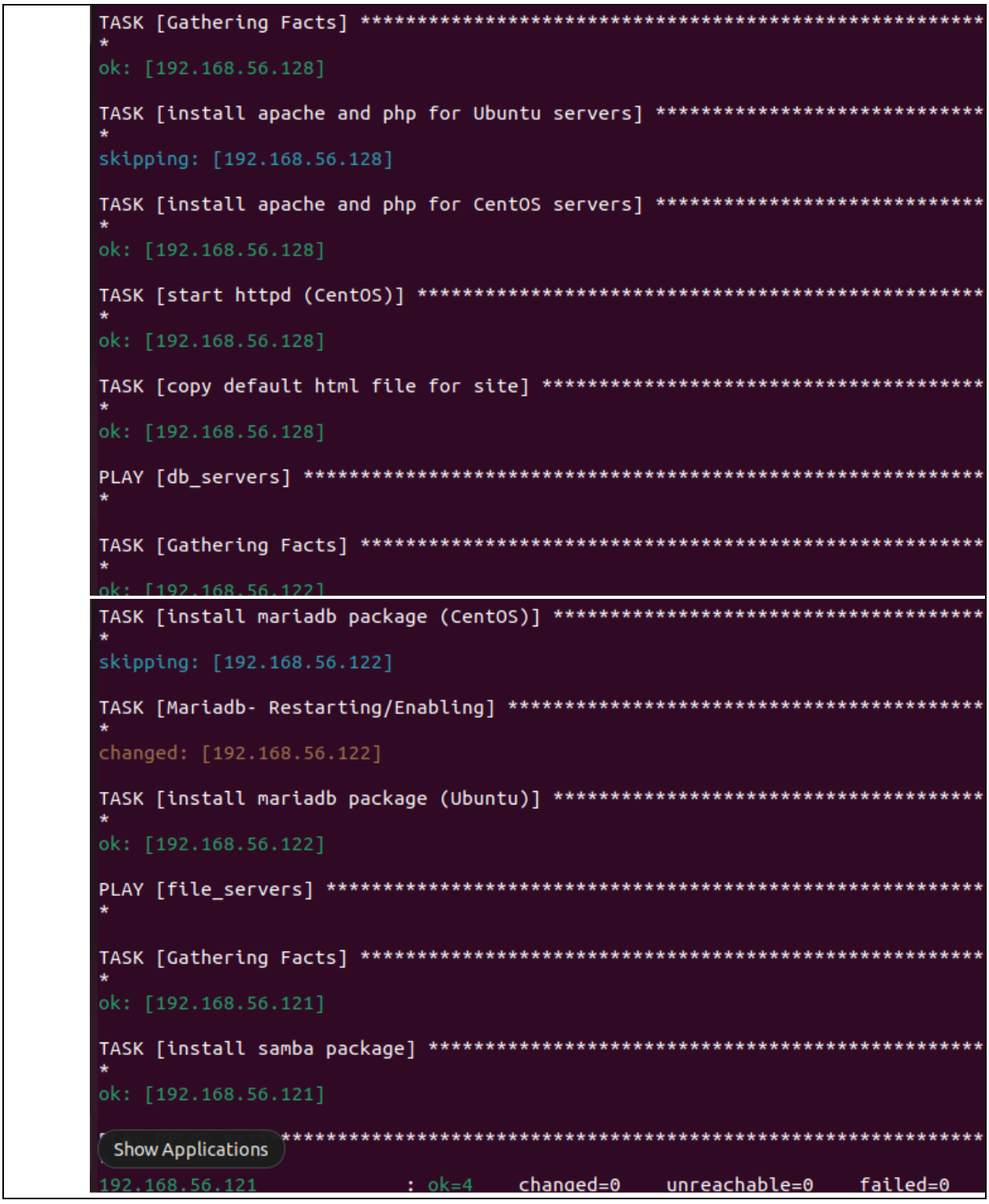
```
- name: copy default html file for site
  tags: apache, apache2, httpd
  copy:
    src: default_site.html
    dest: /var/www/html/index.html
    owner: root
```

group: root
mode: 0644

```
GNU nano 6.2                                site.yml
- name: start httpd (CentOS)
  tags: apache, centos,httpd
  service:
    name: httpd
    state: started
    enabled: true
  when: 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: 00644

- hosts: db_servers
  become: true
  tasks:
```

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

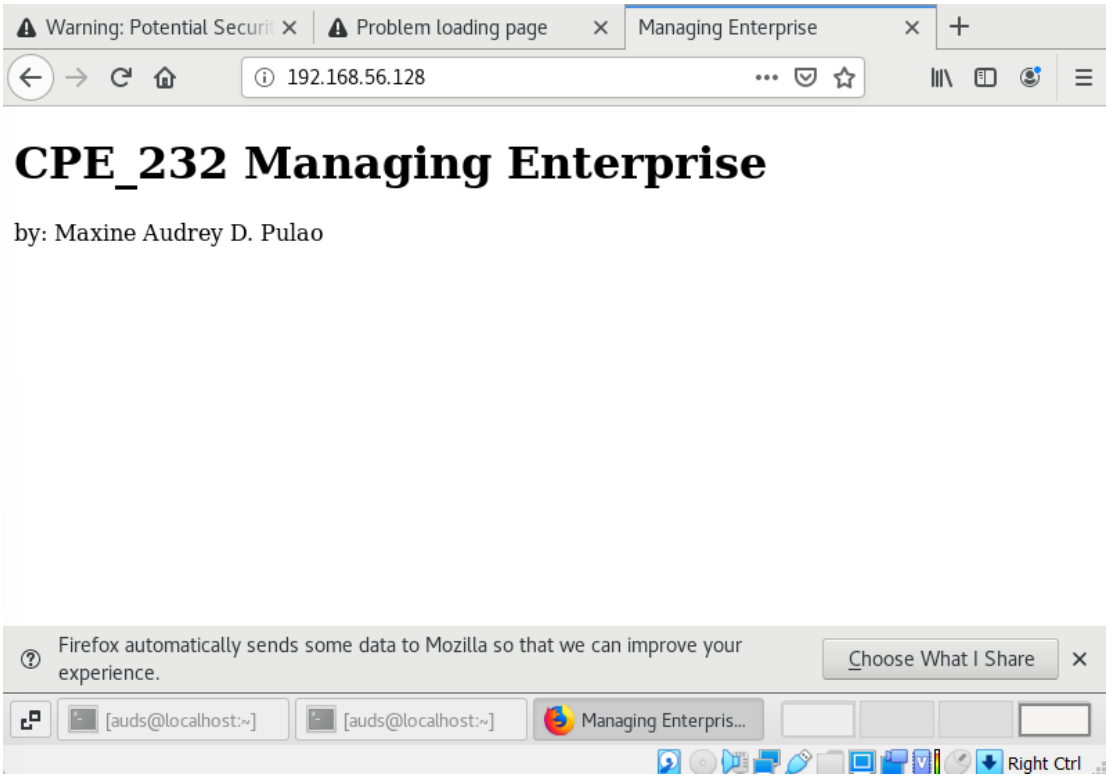
```
aud@rey:~/ansible$ ansible-playbook --ask-become-pass site.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************
*
ok: [192.168.56.122]
ok: [192.168.56.128]
ok: [192.168.56.121]

TASK [install updates (CentOS)] ************************************************
*
skipping: [192.168.56.122]
skipping: [192.168.56.121]
ok: [192.168.56.128]

TASK [install updates (Ubuntu)] ************************************************
*
skipping: [192.168.56.128]
ok: [192.168.56.121]
ok: [192.168.56.122]

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


TASK [Gathering Facts] ***********
*
ok: [192.168.56.128]
TASK [install apache and php for Ubuntu servers] ***********
*
skipping: [192.168.56.128]
TASK [install apache and php for CentOS servers] ***********
*
ok: [192.168.56.128]
TASK [start httpd (CentOS)] ***********
*
ok: [192.168.56.128]
TASK [copy default html file for site] ***********
*
ok: [192.168.56.128]
PLAY [db_servers] ***********
*
TASK [Gathering Facts] ***********
*
ok: [192.168.56.122]
TASK [install mariadb package (CentOS)] ***********
*
skipping: [192.168.56.122]
TASK [Mariadb- Restarting/Enabling] ***********
*
changed: [192.168.56.122]
TASK [install mariadb package (Ubuntu)] ***********
*
ok: [192.168.56.122]
PLAY [file_servers] ***********
*
TASK [Gathering Facts] ***********
*
ok: [192.168.56.121]
TASK [install samba package] ***********
*
ok: [192.168.56.121]
Show Applications
192.168.56.121 : ok=4 changed=0 unreachable=0 failed=0

```
PLAY RECAP *********************************************************************
*
192.168.56.121             : ok=4    changed=0    unreachable=0    failed=0
skipped=1    rescued=0    ignored=0
192.168.56.122             : ok=5    changed=1    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
192.168.56.128             : ok=6    changed=0    unreachable=0    failed=0
skipped=2    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.

```
aud@rey:~/ansible/files$ cat default_site.html
<!DOCTYPE html>
<html>
  <head>
    <title>Managing Enterprise</title>
  </head>
  <body>
    <h1>CPE_232 Managing Enterprise</h1>
    <p>by: Maxine Audrey D. Pulao</p>
  </body>
</html>
```



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
```

aud@rey: ~/ansible × aud@rey: ~/ansible ×

GNU nano 6.2 site.backup.yml

```
---

- hosts: all
  become: true
  pre_tasks:

  - name: install updates (CentOS)
    tags: always
    dnf:
      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"

- 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.
      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

```
[workstations]
192.168.56.122

[web_servers]
192.168.56.128 ansible_user=auds

[db_servers]
192.168.56.122 ansible_user=aud

[file_servers]
192.168.56.121 ansible_user=aud
```

3. Run the playbook. Describe the output.

```
aud@rey:~/ansible$ ansible-playbook --ask-become-pass site.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************
*
ok: [192.168.56.121]
ok: [192.168.56.122]
ok: [192.168.56.128]

TASK [install updates (CentOS)] ************************************
*
skipping: [192.168.56.122]
skipping: [192.168.56.121]
ok: [192.168.56.128]

TASK [install updates (Ubuntu)] ************************************
*
skipping: [192.168.56.128]
ok: [192.168.56.121]
ok: [192.168.56.122]

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

```
TASK [Gathering Facts] *********************************************
*
ok: [192.168.56.122]

TASK [install unzip] ***********************************************
*
ok: [192.168.56.122]

TASK [install terraform] *******************************************
*
changed: [192.168.56.122]

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

TASK [Gathering Facts] *********************************************
*
ok: [192.168.56.128]

TASK [install apache and php for Ubuntu servers] *******************
*
skipping: [192.168.56.128]

TASK [install apache and php for CentOS servers] *******************
*
ok: [192.168.56.128]
```

```
TASK [start httpd (CentOS)] ****************************************************
*
ok: [192.168.56.128]

TASK [copy default html file for site] *****************************************
*
ok: [192.168.56.128]

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

TASK [Gathering Facts] *********************************************************
*
ok: [192.168.56.122]

TASK [install mariadb package (CentOS)] ****************************************
*
skipping: [192.168.56.122]

TASK [Mariadb- Restarting/Enabling] ********************************************
*
changed: [192.168.56.122]

TASK [install mariadb package (Ubuntu)] ****************************************
*
[                                  122]
```

Show Applications

```
TASK [install mariadb package (CentOS)] ****************************************
*
skipping: [192.168.56.122]

TASK [Mariadb- Restarting/Enabling] ********************************************
*
changed: [192.168.56.122]

TASK [install mariadb package (Ubuntu)] ****************************************
*
ok: [192.168.56.122]

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

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

TASK [install samba package] ***************************************************
*
ok: [192.168.56.121]

PLAY RECAP *********************************************************************
*
192.168.56.121             : ok=4    changed=0    unreachable=0    failed=0
skipped=1    rescued=0    ignored=0
```

```
changed: [192.168.56.122]

TASK [install mariadb package (Ubuntu)] ********************************
*
ok: [192.168.56.122]

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

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

TASK [install samba package] *******************************************
*
ok: [192.168.56.121]

PLAY RECAP *************************************************************
*
192.168.56.121             : ok=4    changed=0    unreachable=0    failed=0
skipped=1    rescued=0    ignored=0
192.168.56.122             : ok=8    changed=2    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0
192.168.56.128             : ok=6    changed=0    unreachable=0    failed=0
skipped=2    rescued=0    ignored=0

aud@rey:~/ansible$
```

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

```
aud@rey:~$ 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 configuratio
    refresh            Update local state file against real resources
    show               Inspect Terraform state or plan
    taint              Manually mark a resource for recreation
```

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.

```
---

- 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
```

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.

```
aud@rey:~/ansible/roles$ mkdir {base,web_servers,file_servers,db_servers,works
}
aud@rey:~/ansible/roles$ ls -al
total 28
drwxrwxr-x 7 aud aud 4096 Oct 11 11:14 .
drwxrwxr-x 5 aud aud 4096 Oct 11 11:12 ..
drwxrwxr-x 2 aud aud 4096 Oct 11 11:14 base
drwxrwxr-x 2 aud aud 4096 Oct 11 11:14 db_servers
drwxrwxr-x 2 aud aud 4096 Oct 11 11:14 file_servers
drwxrwxr-x 2 aud aud 4096 Oct 11 11:14 web_servers
drwxrwxr-x 2 aud aud 4096 Oct 11 11:14 workstations
aud@rey:~/ansible/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.

```
aud@rey:~/ansible/roles$ tree
.
├── base
│   └── task
│       └── main.yml
├── db_servers
│   └── task
│       └── main.yml
├── file_servers
│   └── task
│       └── main.yml
├── web_servers
│   └── task
│       └── main.yml
└── workstations
    └── task
        └── main.yml
```

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

```
BECOME password:

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.128]
ok: [192.168.56.121]
ok: [192.168.56.122]

TASK [update repository index (CentOS)] ****************************************
skipping: [192.168.56.122]
skipping: [192.168.56.121]
ok: [192.168.56.128]

TASK [install updates (Ubuntu)] ************************************************
skipping: [192.168.56.128]
ok: [192.168.56.121]
ok: [192.168.56.122]

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.128]
ok: [192.168.56.122]
ok: [192.168.56.121]

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.122]

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.128]

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.122]

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

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

PLAY RECAP *********************************************************************
192.168.56.121             : ok=4    changed=0    unreachable=0    failed=0
d=1    rescued=0    ignored=0
192.168.56.122             : ok=5    changed=0    unreachable=0    failed=0
```

```
PLAY RECAP *********************************************************************
192.168.56.121             : ok=4    changed=0    unreachable=0    failed=0
d=1    rescued=0    ignored=0
192.168.56.122             : ok=5    changed=0    unreachable=0    failed=0
d=1    rescued=0    ignored=0
192.168.56.128             : ok=4    changed=0    unreachable=0    failed=0
d=1    rescued=0    ignored=0
```

```
- hosts: all
  become: true
  pre_tasks:

  - name: install updates (CentOS)
    tags: always
    dnf:
      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"

- 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_>
      dest: /usr/local/bin
      remote_src: yes
      mode: 0755
      owner: root
      group: root

---

- hosts: all
  become: true
  pre_tasks:

  - name: install updates (CentOS)
    tags: always
    dnf:
      update_only: yes
      update_cache: yes
    when: ansible_distribution == "CentOS"

      group: root yesl/bins.hashicorp.com/terraform/0.12.28/terraform_>
```

Reflections:

Answer the following:

1. What is the importance of creating roles?

   The capability of the Linux server will depend on the particular services that are installed and made available on that server. In a nutshell, this is how the idea of

Linux server roles works. Based on the services that have been installed on the server, server roles specify how and for what purposes a particular server is used.

2. What is the importance of managing files?

   In Linux, most of the operations are performed on files. And to handle these files Linux has directories also known as folders which are maintained in a tree-like structure. Though, these directories are also a type of file themselves. By making an orderly file management we can increase efficiency and and avoid bugs and errors.

CONCLUSION:

After performing this activity, I can now manage my files in remote servers using ansible and ssh. I also implemented roles in ansible where I also used the ansible and ssh. This activity helped me master codes relating to ansible and ssh which can be useful in my future endeavors as a computer engineer.

I have learned a lot in this activity and successfully implemented the codes needed. Through this, I can complete other activities with the help of this new knowledge.

HONOR PLEDGE:

I affirm that I will not give or receive any unauthorized help on this activity, and that all work will be my own