Name: Cruz, Patrick Danielle C.	Date Performed: Oct 6, 2025
Course/Section: CPE212 - CPE31S4	Date Submitted: Oct 7, 2025
Instructor: Engr. Robin Valenzuela	Semester and SY: 1st Semester

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.

```
patrick@Workstation:~/CPE232_patcruz$ cat default_site.html
<!DOCTYPE html>
<html lang="en">
<head>
                <meta charset="UTF-8">
                <title>Default Page</title>
</head>
<body>

                <h1>Welcome to the my Page</h1>
                <p>This is a simple website for HTML.</p>
</body>
</html>
```

```
[pat@localhost ~]$ cat /var/www/html/index.html
<!DOCTYPE html>
<html lang="en">
<head>
                <meta charset="UTF-8">
                <title>Default Page</title>
</head>
<body>
                <h1>Welcome to the my Page</h1>
                <p>This is a simple website for HTML.</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

None

```
- hosts: all
  become: true
  pre_tasks:
    - name: install updates (CentOS)
      dnf:
        update_only: yes
        update_cache: yes
      when: ansible_distribution == "CentOS"

    - name: install updates (Ubuntu)
      apt:
        upgrade: dist
        update_cache: yes
      when: ansible_distribution == "Ubuntu"

- hosts: web_servers
  become: true
  tasks:
```

```
    - name: install apache and php for ubuntu servers
      apt:
        name:
          - apache2
          - libapache2-mod-php
        state: latest
      when: ansible_distribution == "Ubuntu"

    - name: install apache and php for CentOS servers
      dnf:
        name:
          - httpd
          - php
        state: latest
      when: ansible_distribution == "CentOS"

- name: copy default html file for site
  hosts: web_servers
  become: true
  tags: apache, apache2, httpd
  tasks:
    - name: copy default html file for site
      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:
        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)
      apt:
        name: mariadb-server
        state: latest
      when: ansible_distribution == "Ubuntu"
```

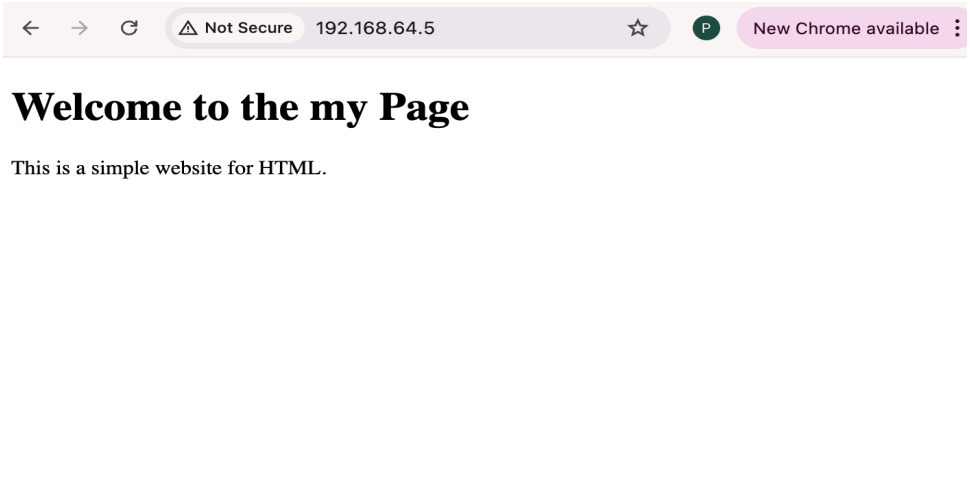
```
- hosts: file_servers
  become: true
  tasks:
    - name: Install samba package
      package:
        name: samba
        state: latest
```

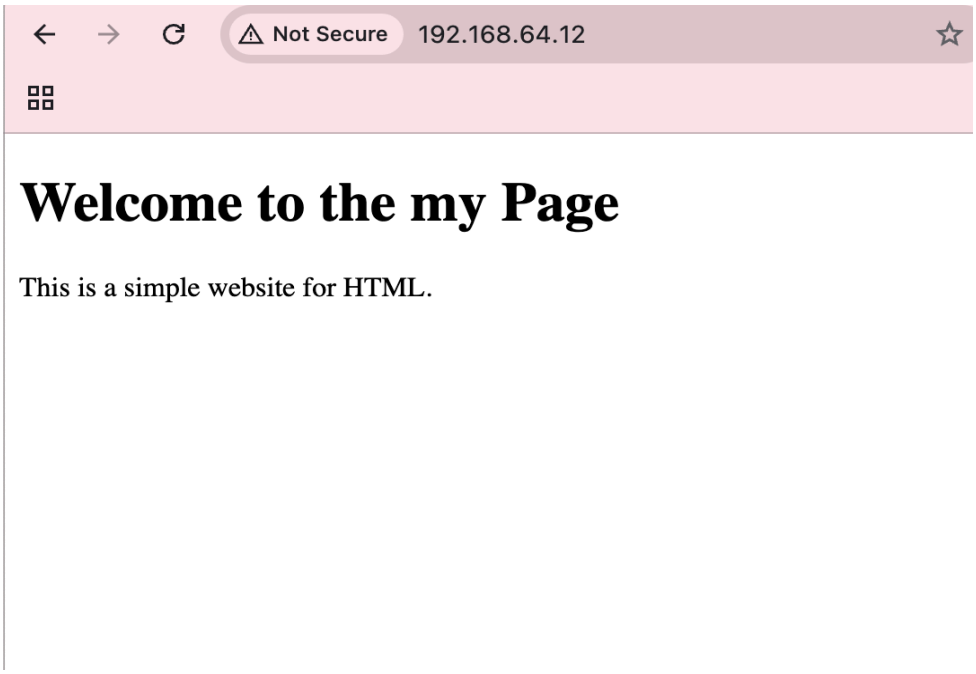
3. Run the playbook *site.yml*. Describe the changes.

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.

- after running the playbook, Ansible copied the `default_site.html` file from the local repository to the remote web servers. The file was placed in `/var/www/html/index.html`, replacing the previous default page. Both

Ubuntu and CentOS servers now display the new HTML content created by the group. The web page shows “Welcome to my Page,” confirming that the playbook successfully deployed the updated website file to all target servers.





5. Sync your local repository with GitHub and describe the changes. https://github.com/Patrickcruz14/Laboratories-Automating/tree/main/lab7

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
```

None

```
patrick@Workstation:~/CPE232_patcruz$ cat site.yml
- hosts: all
  become: true
  pre_tasks:
    - name: install updates (CentOS)
      dnf:
        update_only: yes
        update_cache: yes
      when: ansible_distribution == "CentOS"

    - name: install updates (Ubuntu)
      apt:
        upgrade: dist
        update_cache: yes
      when: ansible_distribution == "Ubuntu"

- hosts: web_servers
  become: true
  tasks:
    - name: install apache and php for ubuntu servers
      apt:
        name:
          - apache2
```

```
          - libapache2-mod-php
        state: latest
      when: ansible_distribution == "Ubuntu"

    - name: install apache and php for CentOS servers
      dnf:
        name:
          - httpd
          - php
        state: latest
      when: ansible_distribution == "CentOS"

- name: copy default html file for site
  hosts: web_servers
  become: true
  tags: apache, apache2, httpd
  tasks:
    - name: copy default html file for site
      copy:
        src: default_site.html
        dest: /var/www/html/index.html
        owner: root
        group: root
        mode: '0644'

- name: setup workstation
  hosts: workstation
  become: true
  tasks:
    - name: install unzip
      package:
        name: unzip
        state: present

    - name: install terraform
      unarchive:
        src:
https://releases.hashicorp.com/terraform/0.12.28/terraform_0.12.28_linux_amd
64.zip
        dest: /usr/local/bin
        remote_src: yes
        mode: '0755'
        owner: root
        group: root


- hosts: db_servers
  become: true
  tasks:
    - name: Install MariaDB package (CentOS)
```

```
      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)
      apt:
        name: mariadb-server
        state: latest
      when: ansible_distribution == "Ubuntu"

- hosts: file_servers
  become: true
  tasks:
    - name: Install samba package
      package:
        name: samba
        state: latest
```

2. Edit the inventory file and add a workstation group. Add any Ubuntu remote server. Make sure to remember the IP address.

```
  GNU nano 8.3                                inventory
[web_servers]
192.168.64.5
192.168.64.12 ansible_user=pat

[workstation]
192.168.64.5

[db_servers]
192.168.64.15

[file_servers]
192.168.64.13
```

3. Run the playbook. Describe the output.

```
patrick@Workstation:~/CPE232_patcruz$ ansible-playbook site.yml -K
BECOME password:

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

TASK [Gathering Facts] *********************************************************
[WARNING]: Platform linux on host 192.168.64.5 is using the discovered Python
interpreter at /usr/bin/python3.13, but future installation of another Python
interpreter could change the meaning of that path. See
https://docs.ansible.com/ansible-
core/2.18/reference_appendices/interpreter_discovery.html for more information.
ok: [192.168.64.5]
[WARNING]: Platform linux on host 192.168.64.15 is using the discovered Python
interpreter at /usr/bin/python3.13, but future installation of another Python
interpreter could change the meaning of that path. See
https://docs.ansible.com/ansible-
core/2.18/reference_appendices/interpreter_discovery.html for more information.
ok: [192.168.64.15]
[WARNING]: Platform linux on host 192.168.64.13 is using the discovered Python
interpreter at /usr/bin/python3.13, but future installation of another Python
interpreter could change the meaning of that path. See
https://docs.ansible.com/ansible-
core/2.18/reference_appendices/interpreter_discovery.html for more information.
ok: [192.168.64.13]
```

```
PLAY [setup workstation] ***************************************************

TASK [Gathering Facts] *****************************************************
ok: [192.168.64.5]

TASK [install unzip] *******************************************************
ok: [192.168.64.5]

TASK [install terraform] ***************************************************
changed: [192.168.64.5]

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

```
PLAY RECAP *********************************************************************
192.168.64.12              : ok=6    changed=0    unreachable=0    failed=0    s
kipped=2    rescued=0    ignored=0
192.168.64.13              : ok=4    changed=0    unreachable=0    failed=0    s
kipped=1    rescued=0    ignored=0
192.168.64.15              : ok=5    changed=1    unreachable=0    failed=0    s
kipped=2    rescued=0    ignored=0
192.168.64.5               : ok=9    changed=1    unreachable=0    failed=0    s
kipped=2    rescued=0    ignored=0
```

- After running the playbook, Ansible successfully installed the unzip package and downloaded the Terraform binary to /usr/local/bin. When the terraform command was executed on the Ubuntu workstation, it displayed the Terraform version information and a list of available commands such as apply, init, plan, and destroy. This confirmed that Terraform was properly installed and ready for use on the workstation.

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

```
patrick@Workstation:~/CPE232_patcruz$ terraform
Usage: terraform [global options] <subcommand> [args]

The available commands for execution are listed below.
The primary workflow commands are given first, followed by
less common or more advanced commands.

Main commands:
  init          Prepare your working directory for other commands
  validate      Check whether the configuration is valid
  plan          Show changes required by the current configuration
  apply         Create or update infrastructure
  destroy       Destroy previously-created infrastructure

All other commands:
  console       Try Terraform expressions at an interactive command prompt
  fmt           Reformat your configuration in the standard style
  force-unlock  Release a stuck lock on the current workspace
  get           Install or upgrade remote Terraform modules
  graph         Generate a Graphviz graph of the steps in an operation
```

```
  graph         Generate a Graphviz graph of the steps in an operation
  import        Associate existing infrastructure with a Terraform resource
  login         Obtain and save credentials for a remote host
  logout        Remove locally-stored credentials for a remote host
  metadata      Metadata related commands
  output        Show output values from your root module
  providers     Show the providers required for this configuration
  refresh       Update the state to match remote systems
  show          Show the current state or a saved plan
  state         Advanced state management
  taint         Mark a resource instance as not fully functional
  test          Experimental support for module integration testing
  untaint       Remove the 'tainted' state from a resource instance
  version       Show the current Terraform version
  workspace     Workspace management

Global options (use these before the subcommand, if any):
  -chdir=DIR    Switch to a different working directory before executing the
                given subcommand.
  -help         Show this help output, or the help for a specified subcommand.
  -version      An alias for the "version" subcommand.
```

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.

None

```
---
- 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.

```
patrick@Workstation:~/CPE232_patcruz/roles$ tree
.
├── base
│   └── tasks
├── db_servers
│   └── tasks
├── file_servers
│   └── tasks
├── web_servers
│   └── tasks
└── workstations
    └── 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.
4. Run the site.yml playbook and describe the output.

```
patrick@Workstation:~/CPE232_patcruz$ ansible-playbook site.yml -K
BECOME password:

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

TASK [Gathering Facts] *********************************************************
[WARNING]: Platform linux on host 192.168.64.5 is using the discovered Python
interpreter at /usr/bin/python3.13, but future installation of another Python
interpreter could change the meaning of that path. See
https://docs.ansible.com/ansible-
core/2.18/reference_appendices/interpreter_discovery.html for more information.
ok: [192.168.64.5]
[WARNING]: Platform linux on host 192.168.64.13 is using the discovered Python
interpreter at /usr/bin/python3.13, but future installation of another Python
interpreter could change the meaning of that path. See
https://docs.ansible.com/ansible-
```

```
TASK [update repository index (CentOS)] ********************************
skipping: [192.168.64.5]
skipping: [192.168.64.15]
skipping: [192.168.64.13]
ok: [192.168.64.12]

TASK [install updates (Ubuntu)] ****************************************
skipping: [192.168.64.12]
ok: [192.168.64.5]
ok: [192.168.64.15]
ok: [192.168.64.13]

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

TASK [Gathering Facts] *************************************************
ok: [192.168.64.5]
ok: [192.168.64.15]
ok: [192.168.64.13]
ok: [192.168.64.12]
```

```
TASK [base : install updates (Ubuntu)] *********************************
skipping: [192.168.64.12]
ok: [192.168.64.5]
ok: [192.168.64.15]
ok: [192.168.64.13]

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

TASK [Gathering Facts] *************************************************
ok: [192.168.64.5]

TASK [workstations : install unzip] ************************************
ok: [192.168.64.5]

TASK [workstations : install terraform] ********************************
ok: [192.168.64.5]

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

TASK [Gathering Facts] *************************************************
ok: [192.168.64.5]
ok: [192.168.64.12]
```

```
TASK [web_servers : install apache and php for ubuntu servers] *****************
skipping: [192.168.64.12]
ok: [192.168.64.5]

TASK [web_servers : install apache and php for CentOS servers] *****************
skipping: [192.168.64.5]
ok: [192.168.64.12]

TASK [web_servers : copy default html file for site] ***************************
ok: [192.168.64.5]
ok: [192.168.64.12]

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.64.15]

TASK [db_servers : Install MariaDB package (CentOS)] ***************************
skipping: [192.168.64.15]

TASK [db_servers : Install MariaDB package (Ubuntu)] ***************************
ok: [192.168.64.15]
```

```
TASK [file_servers : Install samba package] ************************************
ok: [192.168.64.13]

PLAY RECAP *********************************************************************
192.168.64.12              : ok=7    changed=0    unreachable=0    failed=0    ski
pped=3    rescued=0    ignored=0
192.168.64.13              : ok=6    changed=0    unreachable=0    failed=0    ski
pped=2    rescued=0    ignored=0
192.168.64.15              : ok=7    changed=1    unreachable=0    failed=0    ski
pped=3    rescued=0    ignored=0
192.168.64.5               : ok=10   changed=0    unreachable=0    failed=0    ski
pped=3    rescued=0    ignored=0
```

https://github.com/Patrickcruz14/Laboratories-Automating/tree/main/lab7_complete

Reflections:

Answer the following:

1. What is the importance of creating roles?

- Creating roles is important because it helps organize Ansible tasks into smaller reusable parts. This makes the playbook cleaner, easier to understand, and simpler to maintain.

2. What is the importance of managing files?

- Managing files is important because it keeps configurations and scripts organized and consistent. It helps prevent errors and makes updating or reusing files easier in future setups.