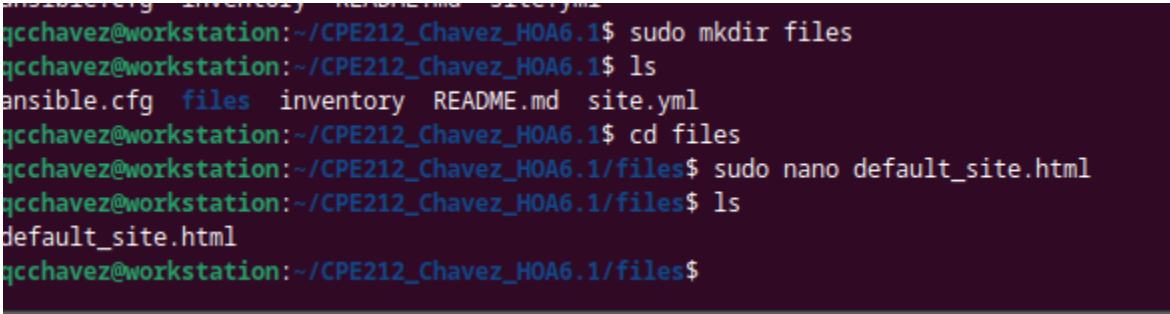
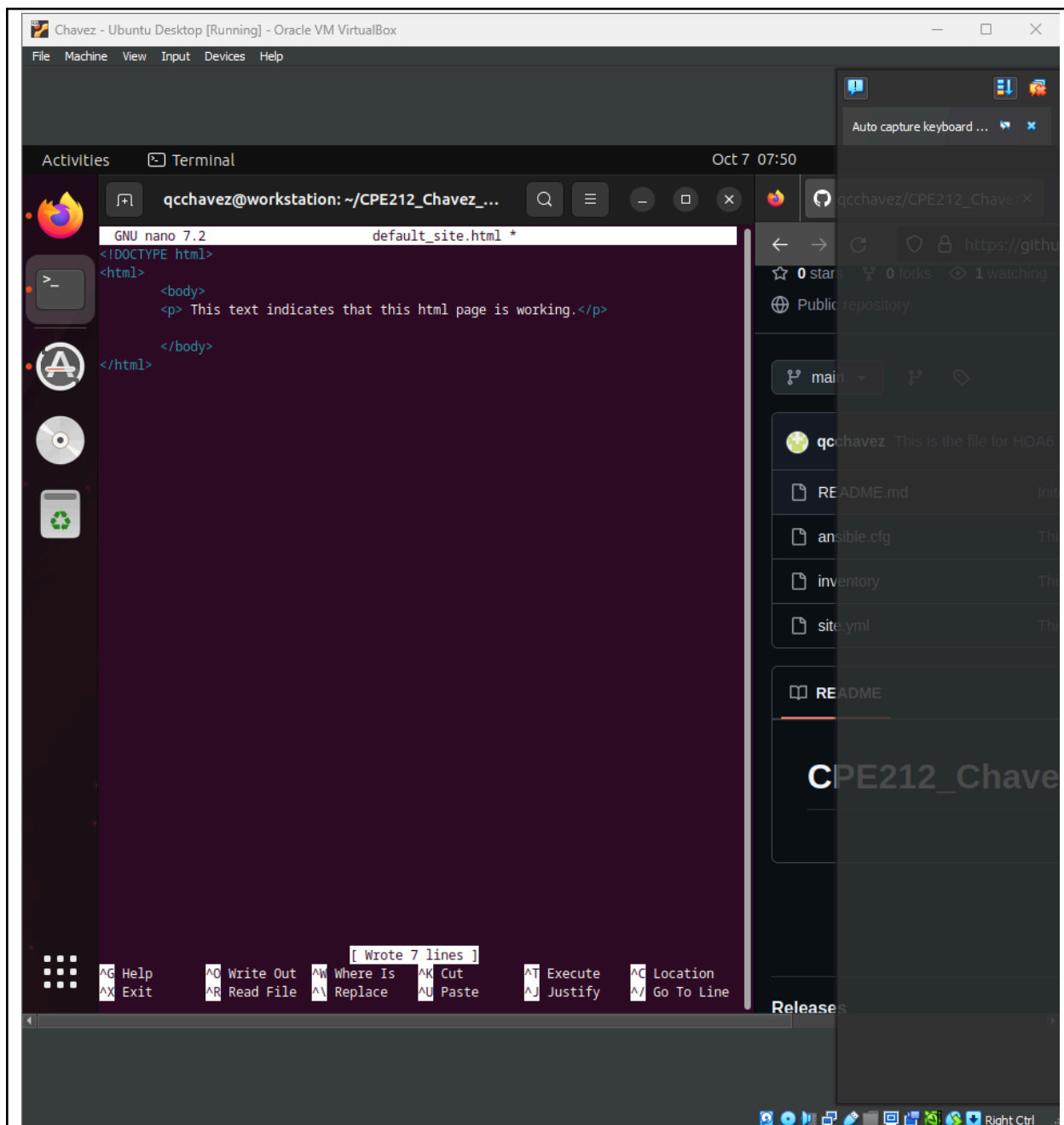


Name: Clarence B. Chavez	Date Performed: Oct. 7, 2024
Course/Section: CPE31S2	Date Submitted: Oct. 9, 2024
Instructor: Engr. Robin Valenzuela	Semester and SY: 1st Sem, 2024-2025
Activity 7: Managing Files and Creating Roles in Ansible	
1. Objectives: 1.1 Manage files in remote servers 1.2 Implement roles 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 <ol style="list-style-type: none"> 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> qcchavez@workstation:~/CPE212_Chavez_HOA6.1\$ sudo mkdir files qcchavez@workstation:~/CPE212_Chavez_HOA6.1\$ ls ansible.cfg files inventory README.md site.yml qcchavez@workstation:~/CPE212_Chavez_HOA6.1\$ cd files qcchavez@workstation:~/CPE212_Chavez_HOA6.1/files\$ sudo nano default_site.html qcchavez@workstation:~/CPE212_Chavez_HOA6.1/files\$ ls default_site.html qcchavez@workstation:~/CPE212_Chavez_HOA6.1/files\$ </pre>	
<ul style="list-style-type: none"> In this screenshot, I’ve created a directory named “files” and inside that directory, I’ve created an html file named “default_site”. 	



- This is the content of my **default_site.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 7.2 site.yml
when: ansible_distribution == "CentOS"

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

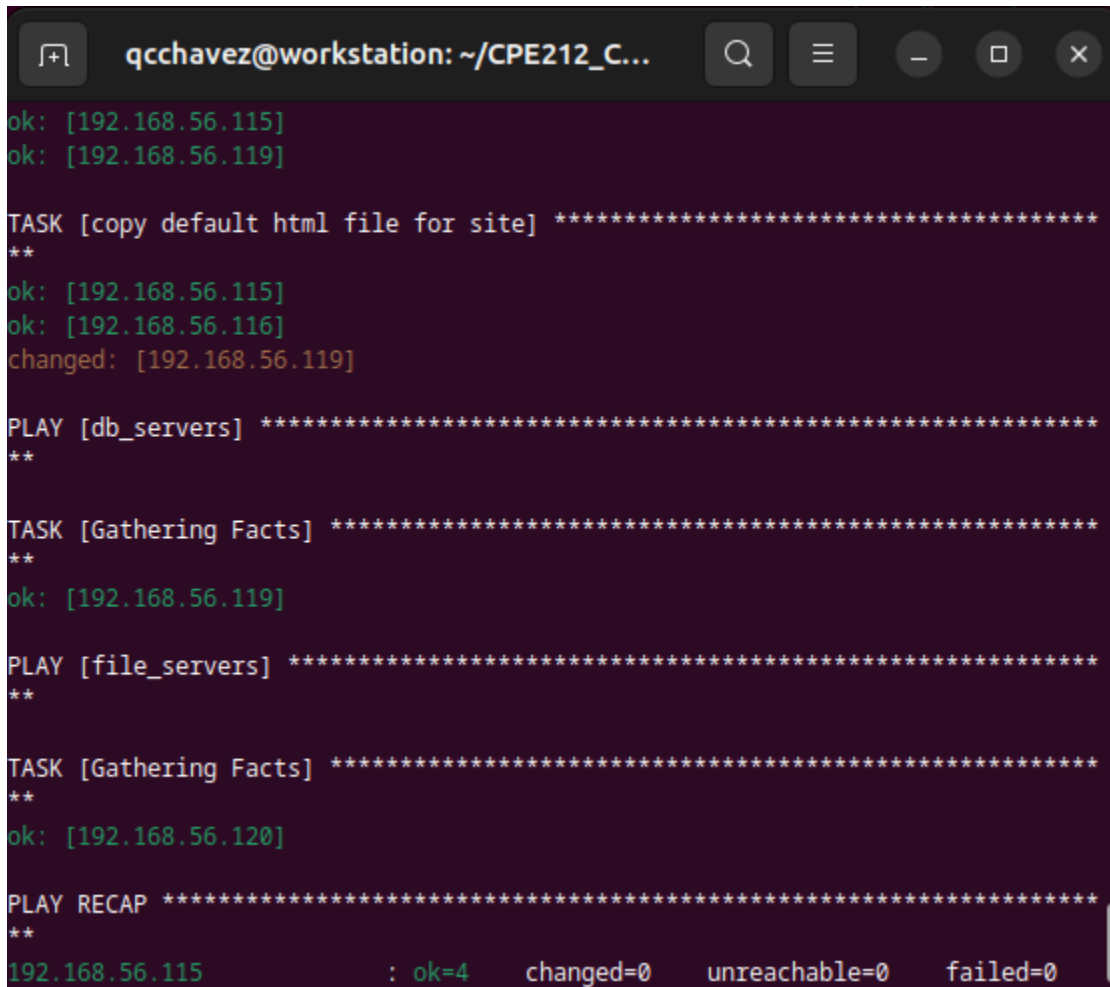
- name: copy default html file for site
  tags: apache,apache2,httpd
  copy:
    src: default_site.html
    dest: /var/www/html/index.html
    owner: root
    group: root
    mode: 0644

#####

- hosts: db_servers
  become: true
```

- In this screenshot, I've added the necessary lines of codes in the `site.yml` playbook based on the required task.

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



The screenshot shows a terminal window with the title bar "qcchavez@workstation: ~/CPE212_C...". The terminal output displays the execution of an Ansible playbook named "site.yml". The output is as follows:

```
ok: [192.168.56.115]
ok: [192.168.56.119]

TASK [copy default html file for site] *****
**
ok: [192.168.56.115]
ok: [192.168.56.116]
changed: [192.168.56.119]

PLAY [db_servers] *****
**

TASK [Gathering Facts] *****
**
ok: [192.168.56.119]

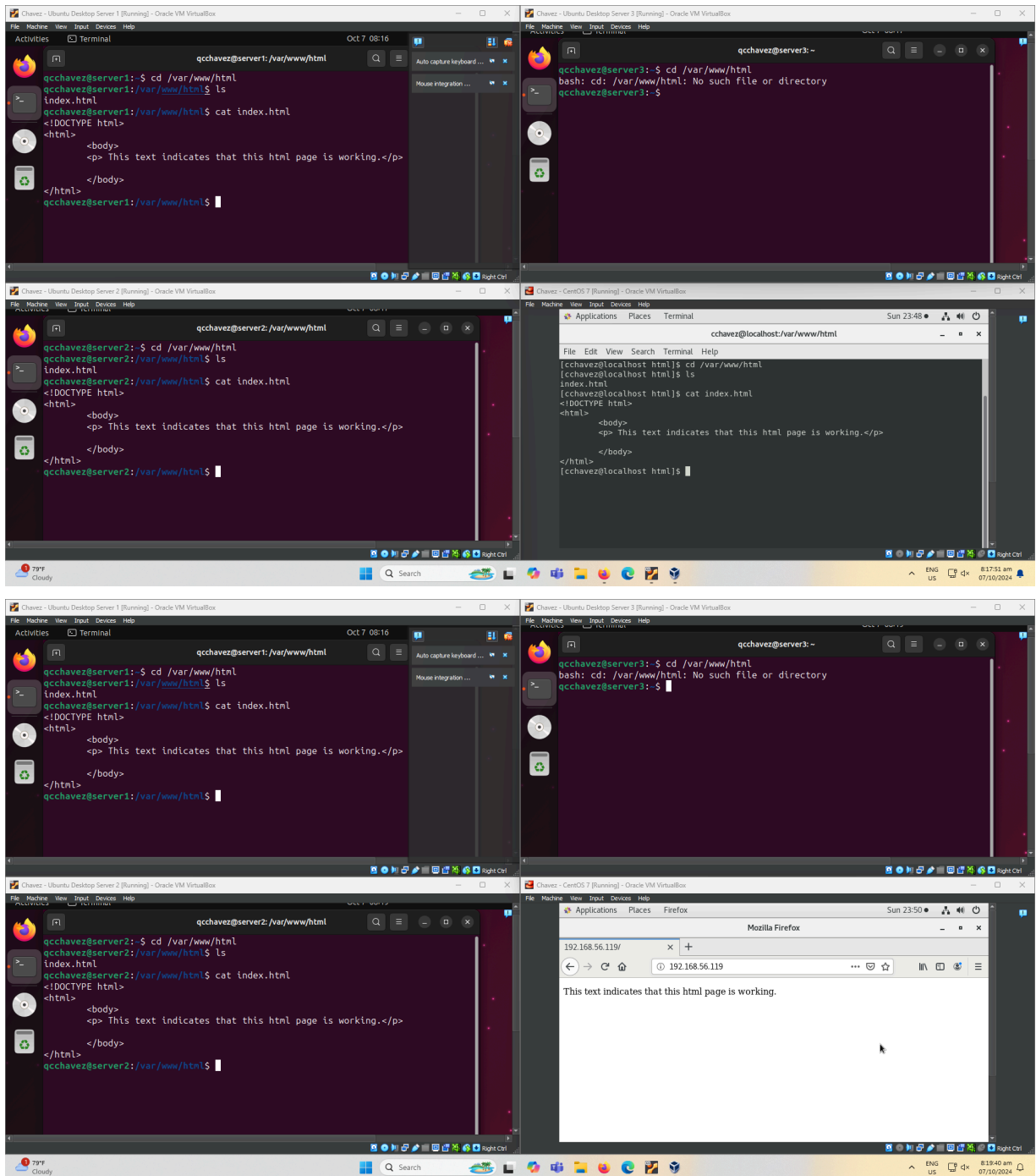
PLAY [file_servers] *****
**

TASK [Gathering Facts] *****
**
ok: [192.168.56.120]

PLAY RECAP *****
**
192.168.56.115      : ok=4    changed=0    unreachable=0    failed=0
```

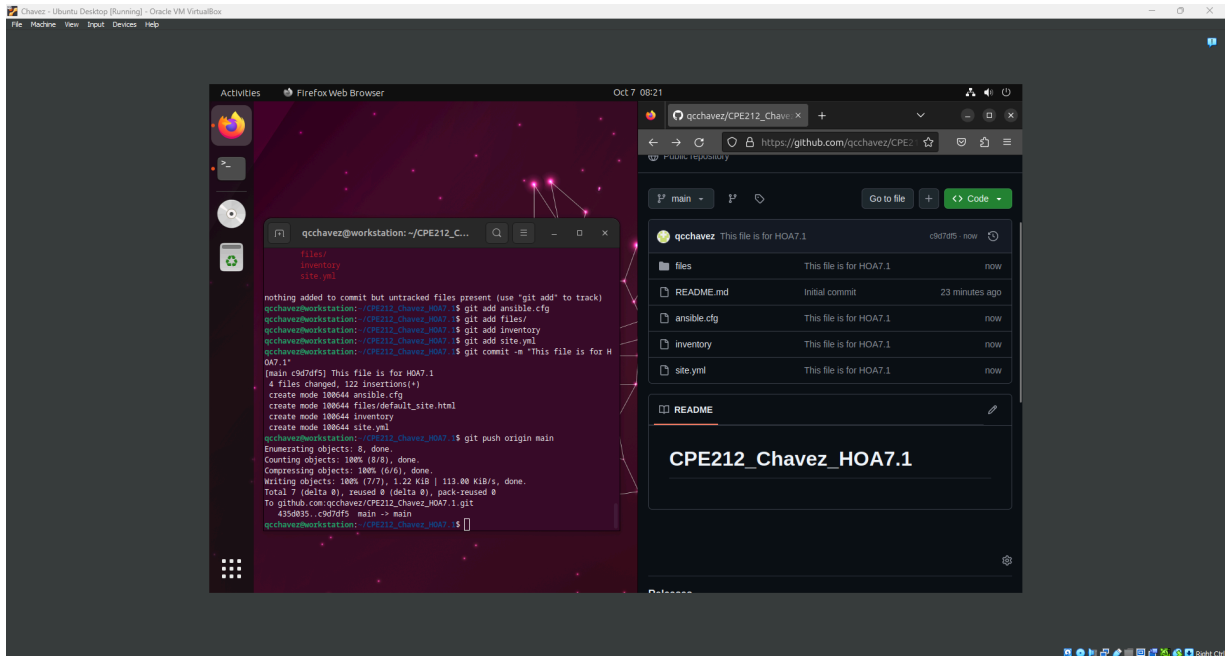
- In this screenshot, it shows that copying the default html file for site to the target remote servers was done successfully.

- 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.



- In this screenshot, I've shown if the `index.html` was applied to the remote servers in the **web_servers** group.

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



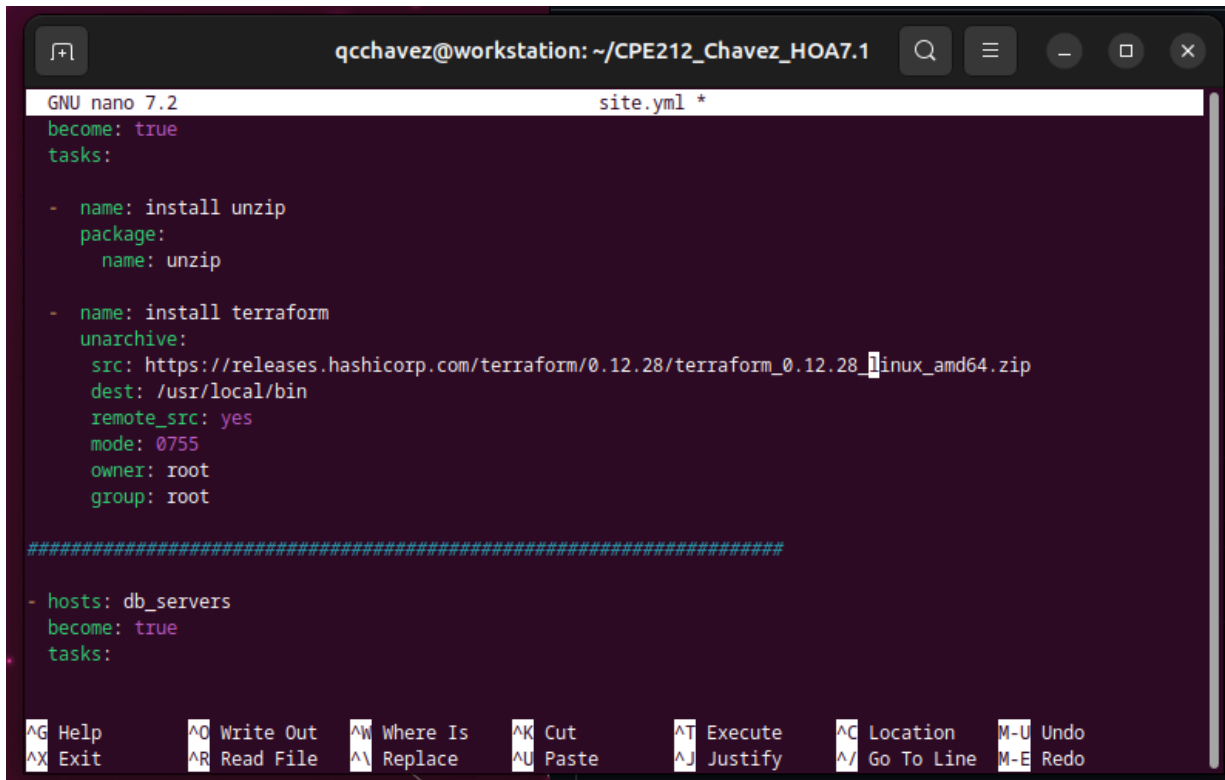
- In this screenshot, I've committed to the GitHub to comply with the task 1.

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



```
GNU nano 7.2 site.yml *
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

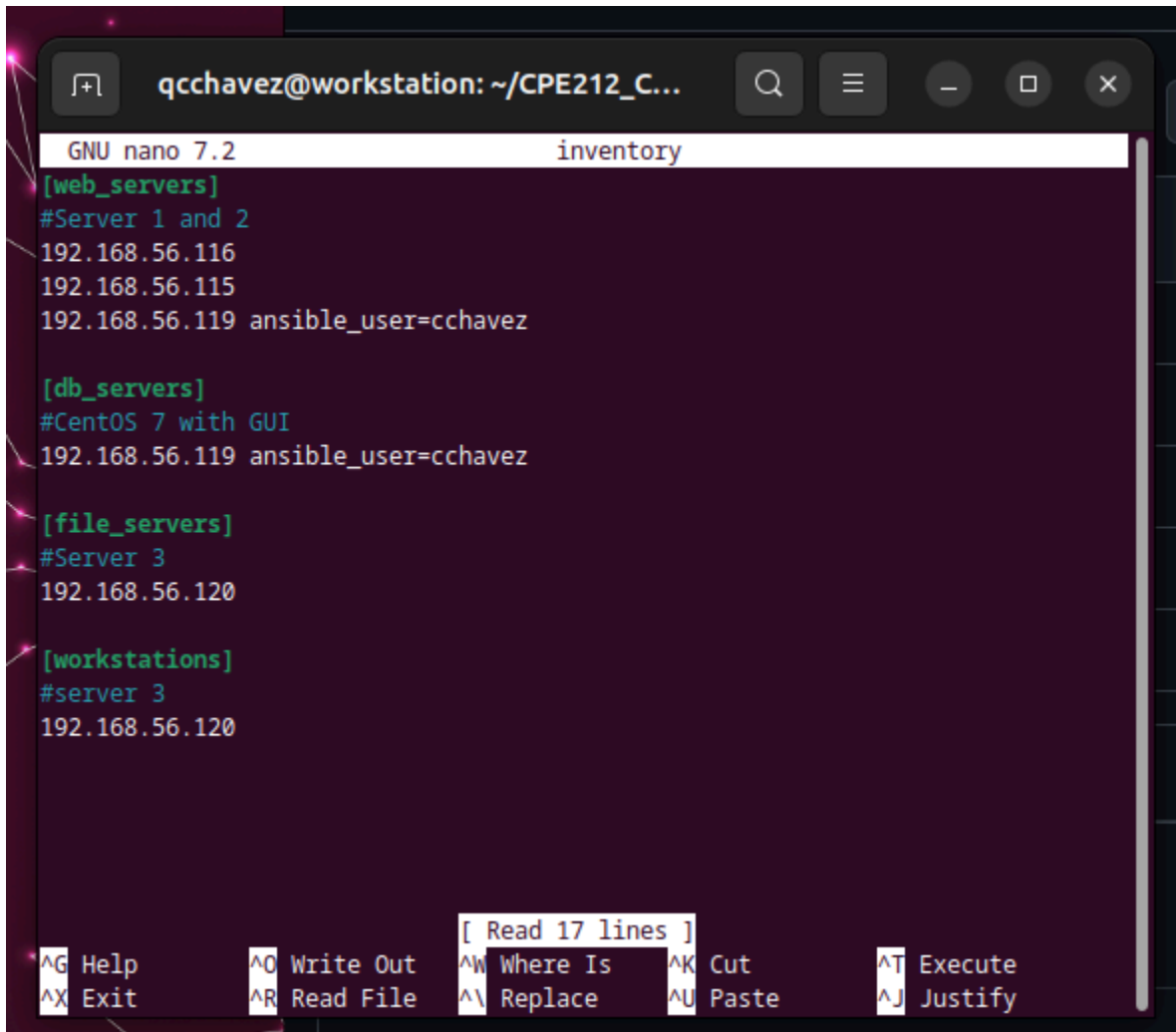
#####

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

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^_ Go To Line M-E Redo

- In this screenshot, I've added the necessary lines of codes for the workstations group.

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



```
qccchavez@workstation: ~/CPE212_C...
GNU nano 7.2 inventory
[web_servers]
#Server 1 and 2
192.168.56.116
192.168.56.115
192.168.56.119 ansible_user=cchavez

[db_servers]
#CentOS 7 with GUI
192.168.56.119 ansible_user=cchavez

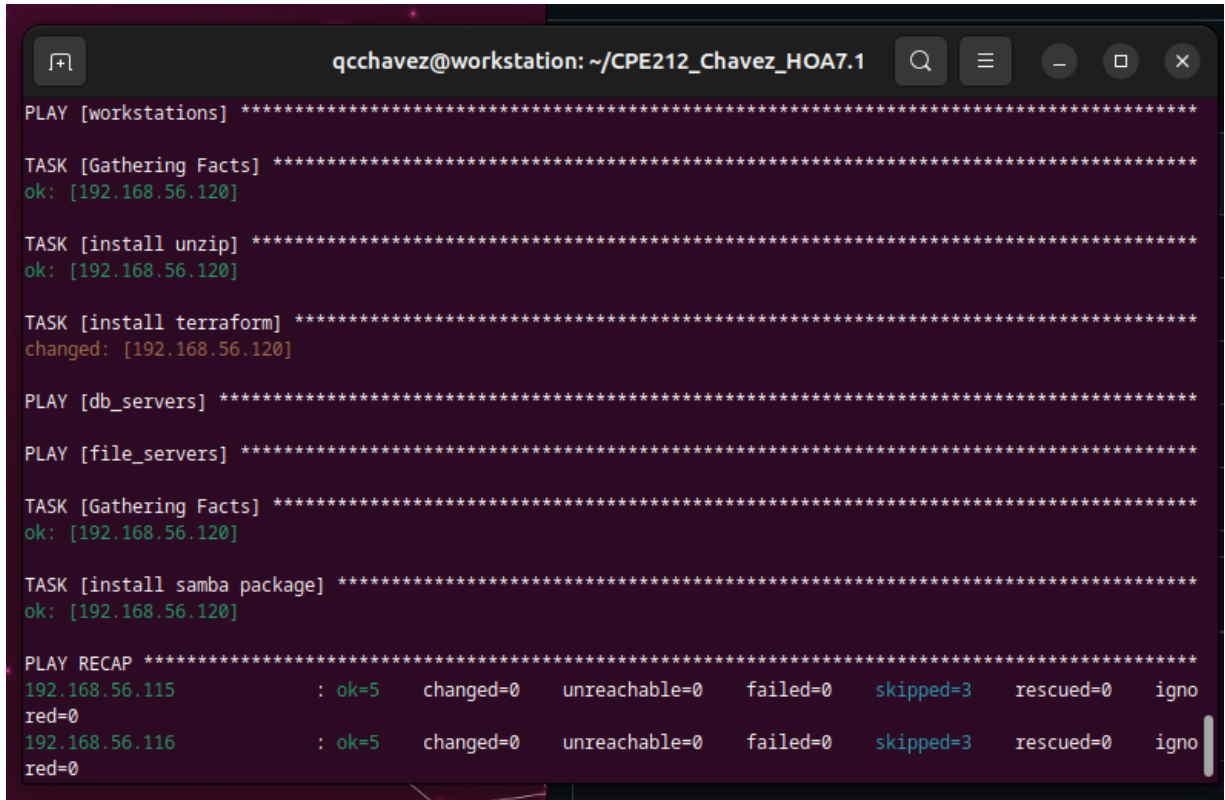
[file_servers]
#Server 3
192.168.56.120

[workstations]
#server 3
192.168.56.120

[ Read 17 lines ]
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify
```

- In this screenshot, I've added the **server3 IP address** inside the **workstations** group.

3. Run the playbook. Describe the output.

A terminal window titled 'qcchavez@workstation: ~/CPE212_Chavez_HOA7.1' displays the output of an Ansible playbook. The output is color-coded: green for success, yellow for changes, and red for errors. The playbook consists of several plays: 'workstations', 'db_servers', and 'file_servers'. The 'workstations' play includes tasks for 'Gathering Facts', 'install unzip', and 'install terraform'. The 'db_servers' and 'file_servers' plays also include a 'Gathering Facts' task. The 'install terraform' task shows a change on host 192.168.56.120. The 'PLAY RECAP' section at the bottom summarizes the results for two hosts: 192.168.56.115 and 192.168.56.116. Both hosts show 5 OK, 0 changed, 0 unreachable, 0 failed, 3 skipped, 0 rescued, and 0 ignored.

```
PLAY [workstations] *****
TASK [Gathering Facts] *****
ok: [192.168.56.120]

TASK [install unzip] *****
ok: [192.168.56.120]

TASK [install terraform] *****
changed: [192.168.56.120]

PLAY [db_servers] *****

PLAY [file_servers] *****

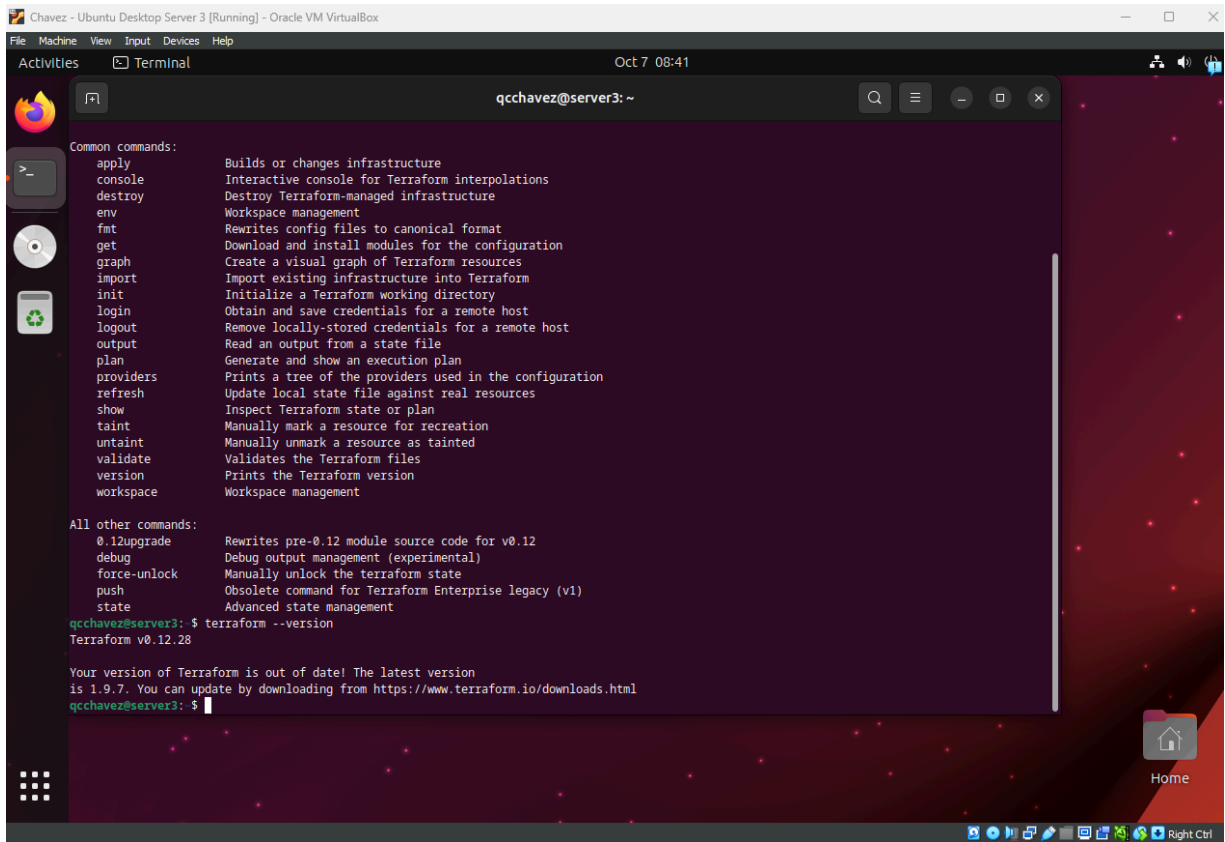
TASK [Gathering Facts] *****
ok: [192.168.56.120]

TASK [install samba package] *****
ok: [192.168.56.120]

PLAY RECAP *****
192.168.56.115      : ok=5    changed=0    unreachable=0    failed=0    skipped=3    rescued=0    igno
red=0
192.168.56.116      : ok=5    changed=0    unreachable=0    failed=0    skipped=3    rescued=0    igno
red=0
```

- After running the playbook, it shows that the **install terraform** task was successfully done and made changes to the target remote server.

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



The screenshot shows a terminal window titled "Chavez - Ubuntu Desktop Server 3 [Running] - Oracle VM VirtualBox". The terminal displays a list of common Terraform commands and their descriptions, followed by the output of the command `terraform --version`. The output indicates that the installed version is 0.12.28, which is out of date compared to the latest version 1.9.7. A message suggests updating from <https://www.terraform.io/downloads.html>.

```
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
  untaint    Manually unmark a resource as tainted
  validate   Validates the Terraform files
  version    Prints the Terraform version
  workspace  Workspace management

All other commands:
  0.12upgrade  Rewrites pre-0.12 module source code for v0.12
  debug        Debug output management (experimental)
  force-unlock Manually unlock the terraform state
  push         Obsolete command for Terraform Enterprise legacy (v1)
  state        Advanced state management

qcchavez@server3: $ 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/downloads.html
qcchavez@server3: $
```

- After the successful changes, it was confirmed that the terraform was successfully installed in the target remote server which is the **Server3**.

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

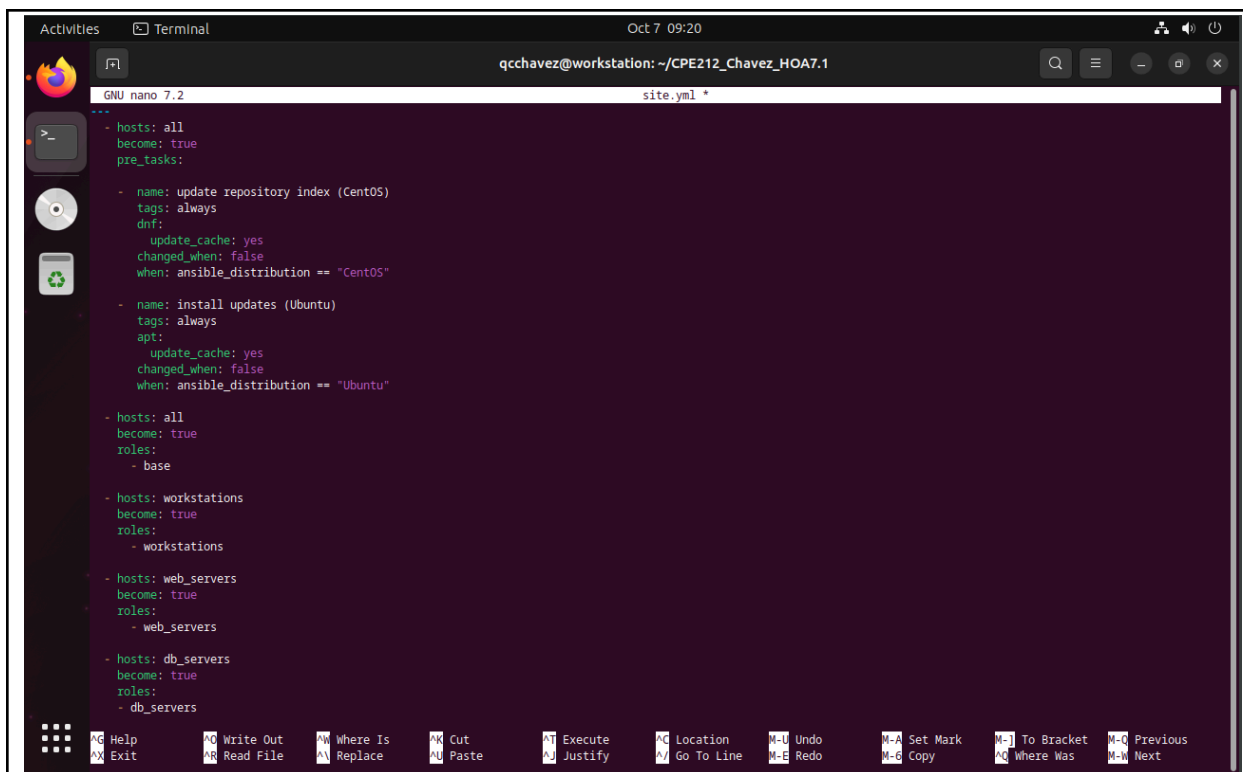
- hosts: workstations
  become: true
  roles:
    - workstations

- 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.



```
GNU nano 7.2 site.yml *
---
- hosts: all
  become: true
  pre_tasks:

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

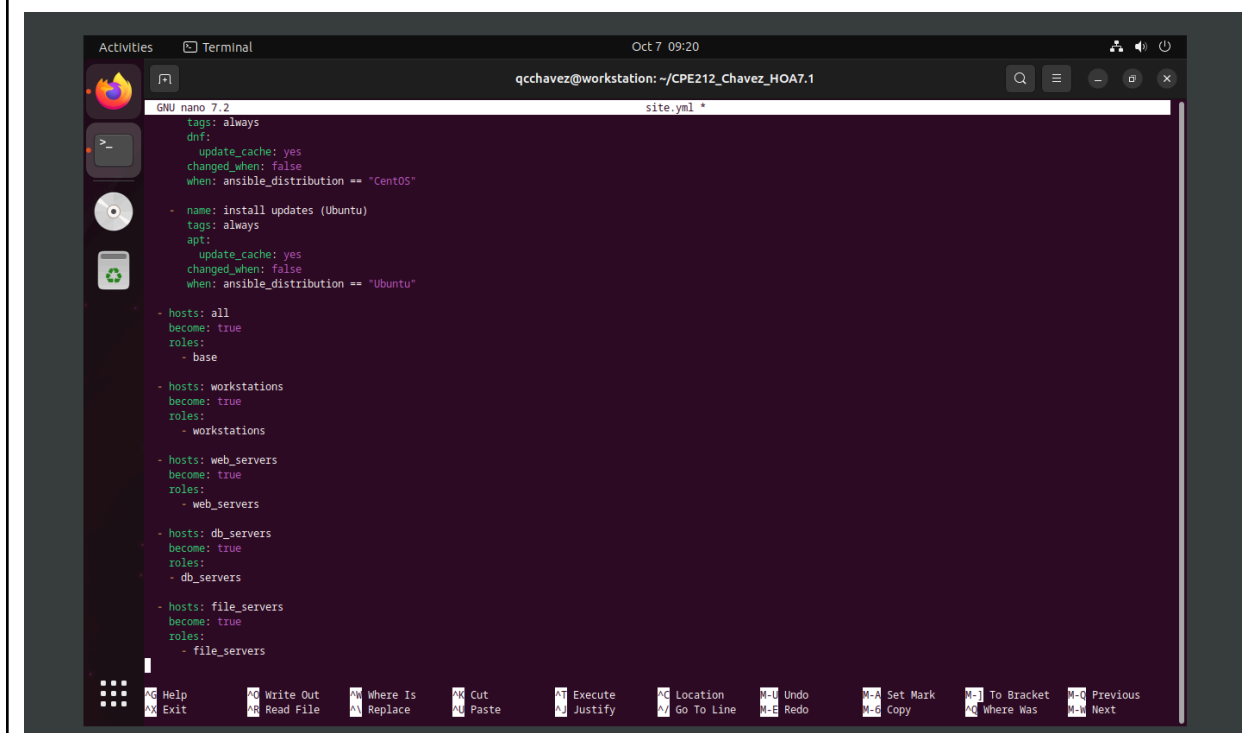
- name: install updates (Ubuntu)
  tags: always
  apt:
    update_cache: yes
    changed_when: false
    when: ansible_distribution == "Ubuntu"

- 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
```



```
GNU nano 7.2 site.yml *
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
```

- In these screenshots, I've made a whole change in the **site.yml** file and encoded the necessary lines of codes based on the screenshots given above.

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.

```
qcchavez@workstation:~/CPE212_Chavez_H0A7.1$ ls
ansible.cfg  backups  files  inventory  README.md  site.yml
qcchavez@workstation:~/CPE212_Chavez_H0A7.1$ mkdir roles
qcchavez@workstation:~/CPE212_Chavez_H0A7.1$ cd roles
qcchavez@workstation:~/CPE212_Chavez_H0A7.1/roles$ mkdir base
qcchavez@workstation:~/CPE212_Chavez_H0A7.1/roles$ mkdir web_servers
qcchavez@workstation:~/CPE212_Chavez_H0A7.1/roles$ mkdir file_servers
qcchavez@workstation:~/CPE212_Chavez_H0A7.1/roles$ mkdir db_servers
qcchavez@workstation:~/CPE212_Chavez_H0A7.1/roles$ mkdir workstations
qcchavez@workstation:~/CPE212_Chavez_H0A7.1/roles$ cd base
qcchavez@workstation:~/CPE212_Chavez_H0A7.1/roles/base$ mkdir tasks
qcchavez@workstation:~/CPE212_Chavez_H0A7.1/roles/base$ cd ..
qcchavez@workstation:~/CPE212_Chavez_H0A7.1/roles$ mkdir web_servers
mkdir: cannot create directory 'web_servers': File exists
qcchavez@workstation:~/CPE212_Chavez_H0A7.1/roles$ cd web_servers
qcchavez@workstation:~/CPE212_Chavez_H0A7.1/roles/web_servers$ mkdir tasks
qcchavez@workstation:~/CPE212_Chavez_H0A7.1/roles/web_servers$ cd ..
qcchavez@workstation:~/CPE212_Chavez_H0A7.1/roles$ cd file_servers
qcchavez@workstation:~/CPE212_Chavez_H0A7.1/roles/file_servers$ mkdir tasks
qcchavez@workstation:~/CPE212_Chavez_H0A7.1/roles/file_servers$ cd ..
qcchavez@workstation:~/CPE212_Chavez_H0A7.1/roles$ cd db_servers
qcchavez@workstation:~/CPE212_Chavez_H0A7.1/roles/db_servers$ mkdir tasks
qcchavez@workstation:~/CPE212_Chavez_H0A7.1/roles/db_servers$ cd ..
qcchavez@workstation:~/CPE212_Chavez_H0A7.1/roles$ cd workstations
qcchavez@workstation:~/CPE212_Chavez_H0A7.1/roles/workstations$ mkdir tasks
qcchavez@workstation:~/CPE212_Chavez_H0A7.1/roles/workstations$
```

- In this screenshot, I've created the necessary directories and created **tasks** directories in each one of them.

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.

base

The screenshot shows a terminal window titled "Chavez - Ubuntu Desktop [Running] - Oracle VM VirtualBox". The terminal output shows the following:

```
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

qcchavez@workstation: ~/CPE212_Chavez_HOA7.1$ cd roles/base/tasks
qcchavez@workstation: ~/CPE212_Chavez_HOA7.1/roles/base/tasks$ cat main.yml
---
```

The terminal output then shows the contents of the main.yml file:

```
- 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"
qcchavez@workstation: ~/CPE212_Chavez_HOA7.1/roles/base/tasks$
```

In the background, a web browser window shows the GitHub repository "qcchavez/CPE212_Chavez_HOA7.1". The repository has 0 stars, 0 forks, and 1 watching. The file list includes: backups, files, roles, README.md, ansible.cfg, inventory, old_site.yml, and site.yml.

- In this screenshot, I've created a **main.yml** file and copy-pasted the necessary codes for the **base** role.

workstations

```
qcchavez@workstation:~/CPE212_Chavez_H0A7.1/roles/workstations/tasks$ cat main.yml
#####

- 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
qcchavez@workstation:~/CPE212_Chavez_H0A7.1/roles/workstations/tasks$
```

- In this screenshot, I've created a **main.yml** file and copy-pasted the necessary codes for the **workstations** role.

web_servers

```
qcchavez@workstation:~/CPE212_Chavez_H0A7.1/roles$ cd web_servers/tasks
qcchavez@workstation:~/CPE212_Chavez_H0A7.1/roles/web_servers/tasks$ cat main.yml
#####

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

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

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

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

qcchavez@workstation:~/CPE212_Chavez_H0A7.1/roles/web_servers/tasks$
```

- In this screenshot, I've created a **main.yml** file and copy-pasted the necessary codes for the **web_servers** role.

db_servers

```
qcchavez@workstation:~/CPE212_Chavez_H0A7.1/roles/db_servers/tasks$ cat main.yml
#####

- name: install mariadb package (CentOS)
  tags: centos,db,mariadb
  dnf:
    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"

qcchavez@workstation:~/CPE212_Chavez_H0A7.1/roles/db_servers/tasks$
```

- In this screenshot, I've created a **main.yml** file and copy-pasted the necessary codes for the **db_servers** role.

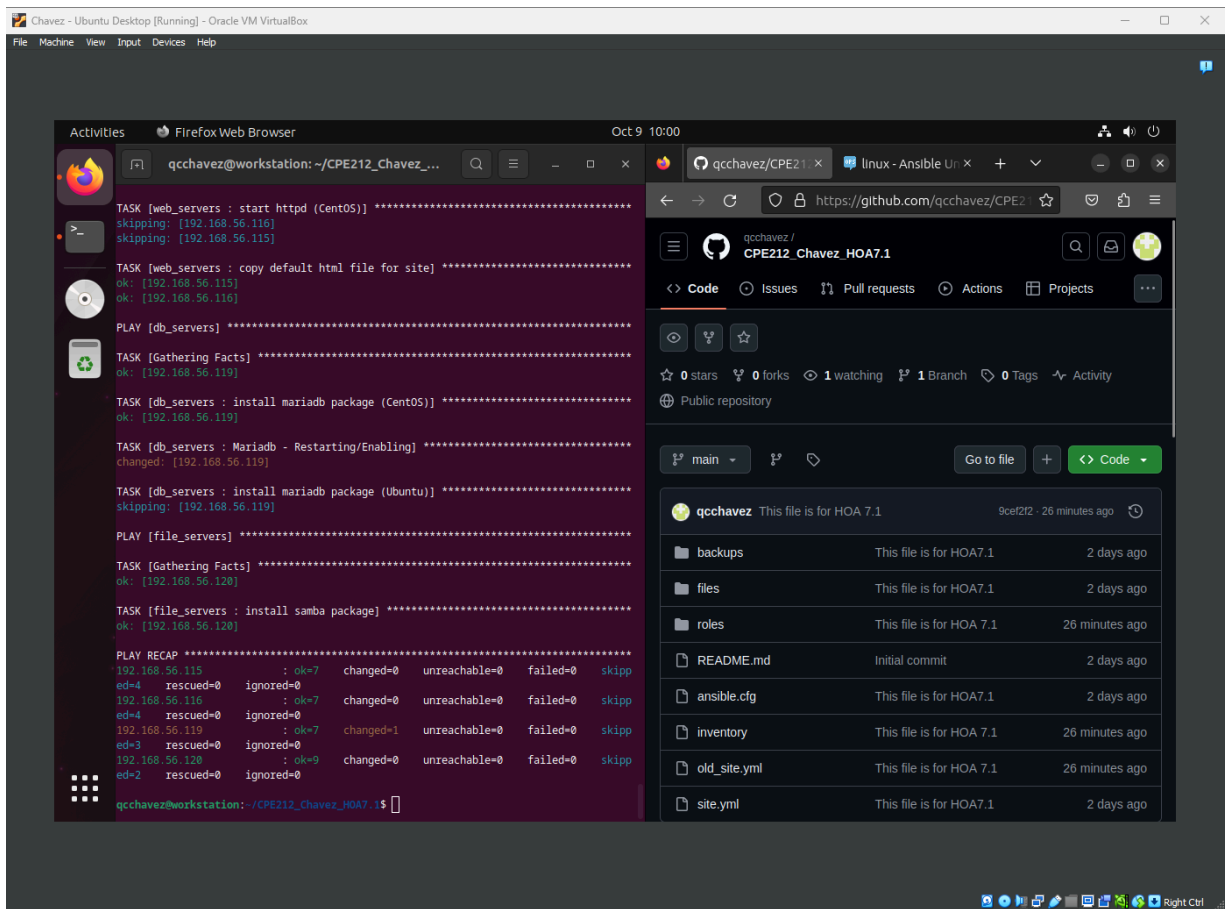
file_servers

```
qcchavez@workstation:~/CPE212_Chavez_HOA7.1/roles$ cd file_servers/tasks
qcchavez@workstation:~/CPE212_Chavez_HOA7.1/roles/file_servers/tasks$ cat main.yml
#####

- name: install samba package
  tags: samba
  package:
    name: samba
    state: latest
qcchavez@workstation:~/CPE212_Chavez_HOA7.1/roles/file_servers/tasks$ ss
```

- In this screenshot, I've created a **main.yml** file and copy-pasted the necessary codes for the **file_servers** role.

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



- In this screenshot, I've ran the **site.yml** playbook and it shows that it worked successfully.

Committing to GitHub

The screenshot shows a terminal window with the following commands and output:

```
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   inventory
        modified:   roles/db_servers/tasks/main.yml
        modified:   roles/file_servers/tasks/main.yml
        modified:   roles/web_servers/tasks/main.yml
        modified:   roles/workstations/tasks/main.yml
        modified:   site.yml

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        old_site.yml

no changes added to commit (use "git add" and/or "git commit -a")
qcchavez@workstation: ~/CPE212_Chavez_HOAT7.1 $ git add inventory
qcchavez@workstation: ~/CPE212_Chavez_HOAT7.1 $ git add roles
qcchavez@workstation: ~/CPE212_Chavez_HOAT7.1 $ git add old_site.yml
qcchavez@workstation: ~/CPE212_Chavez_HOAT7.1 $ git push origin main
Everything up-to-date
qcchavez@workstation: ~/CPE212_Chavez_HOAT7.1 $ git commit -m "This file is for HOA 7.1"
[main 8cef2f2] This file is for HOA 7.1
 7 files changed, 140 insertions(+), 419 deletions(-)
 create mode 100644 old_site.yml
qcchavez@workstation: ~/CPE212_Chavez_HOAT7.1 $ git push origin main
Enumerating objects: 31, done.
Counting objects: 100% (26/26), done.
Compressing objects: 100% (18/18), done.
Writing objects: 100% (20/20), 1.43 KiB | 937.00 KiB/s, done.
Total 20 (delta 5), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (5/5), completed with 2 local objects.
To github.com:qcchavez/CPE212_Chavez_HOAT7.1.git
   c18efad..8cef2f2  main -> main
qcchavez@workstation: ~/CPE212_Chavez_HOAT7.1 $
```

The GitHub repository page shows the repository name 'qcchavez/CPE212_Chavez_HOAT7.1' and a list of files:

File	Commit	Time
backups	This file is for HOA7.1	2 days ago
files	This file is for HOA7.1	2 days ago
roles	This file is for HOA 7.1	now
README.md	Initial commit	2 days ago
ansible.cfg	This file is for HOA7.1	2 days ago
inventory	This file is for HOA 7.1	now
old_site.yml	This file is for HOA 7.1	now
site.yml	This file is for HOA7.1	2 days ago

- In this screenshot, I've committed my repository to GitHub

Reflections:

Answer the following:

1. What is the importance of creating roles?
 - The importance of creating roles is that it can be maintained properly, can be reused over and over, and also modular since every role contains a .yml file that is specifically made for that exact role.
2. What is the importance of managing files?
 - The importance of managing files is that it is more neat, organized, and less room for making mistakes. Properly managed files also means that it is easier to detect the cause of the errors in case some of them occur in the future.