| | |
|---|---|
| **Name:** Jaira Biane Maculada | **Date Performed:** 10/02/23 |
| **Course/Section:**CPE232/CPE31S6 | **Date Submitted:** 10/03/23 |
| **Instructor:** Dr. Jonathan V. Taylar | **Semester SY:** 1st Sem(2023-2024) |

## Activity 7: Managing Files and Creating Roles in Ansible

1. **Objectives:**

1.1 Manage files in remote servers

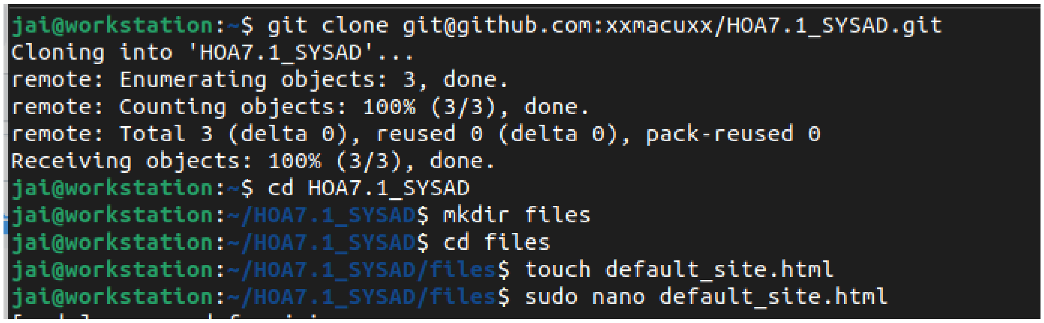
1.2 Implement roles in ansible

2. **Discussion**:

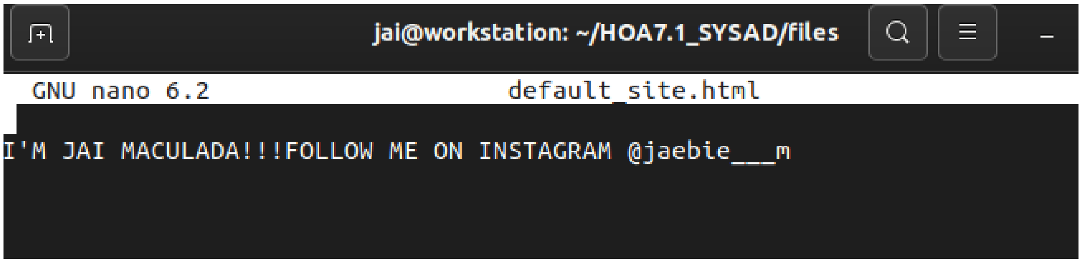
In this activity, we look at the concept of copying a file to a server. We are going to create a file into our git repository and use Ansible to grab that file and put it into a particular place so that we could do things like customize a default website, or maybe install a default configuration file. 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.



```
jai@workstation:~$ git clone git@github.com:xxmacuxx/HOA7.1_SYSAD.git
Cloning into 'HOA7.1_SYSAD'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
jai@workstation:~$ cd HOA7.1_SYSAD
jai@workstation:~/HOA7.1_SYSAD$ mkdir files
jai@workstation:~/HOA7.1_SYSAD$ cd files
jai@workstation:~/HOA7.1_SYSAD/files$ touch default_site.html
jai@workstation:~/HOA7.1_SYSAD/files$ sudo nano default_site.html
```

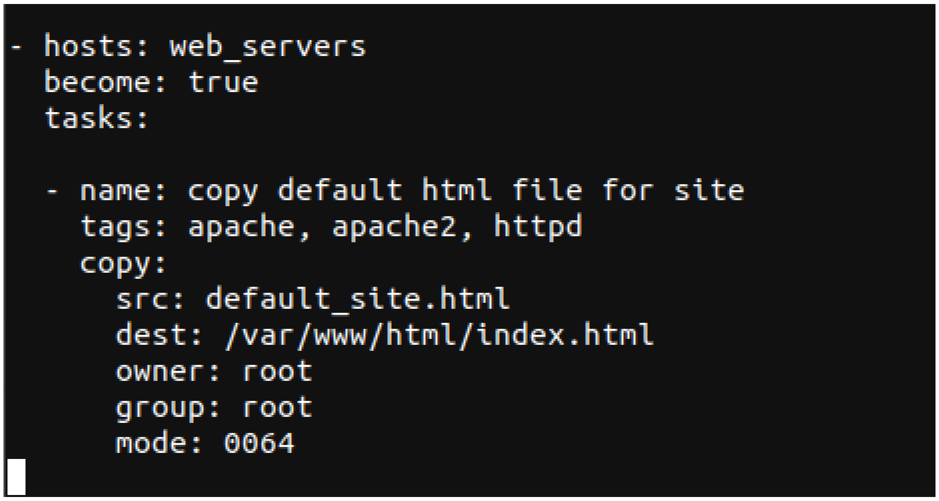


```
  GNU nano 6.2                       default_site.html

I'M JAI MACULADA!!!FOLLOW ME ON INSTAGRAM @jaebie___m
```

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

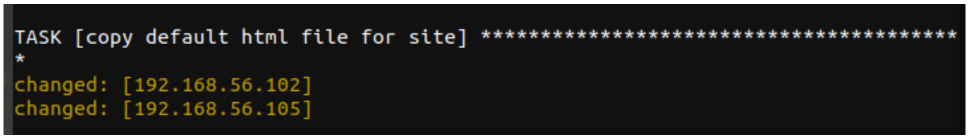
     tags: apache, apache2, httpd

```
    copy:
      src: default_site.html
      dest: /var/www/html/index.html
      owner: root
      group: root
      mode: 0644
```



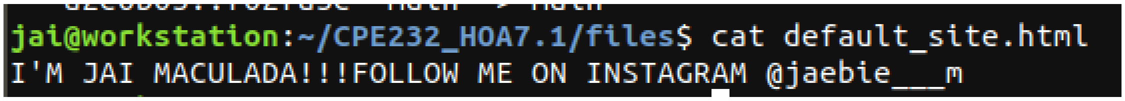
```
- hosts: web_servers
  become: true
  tasks:

  - 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: 0064
```

3. Run the playbook *site.yml*. Describe the changes. **I was able to create a web page wherein it will output the content of the file 'default_site.html.'**



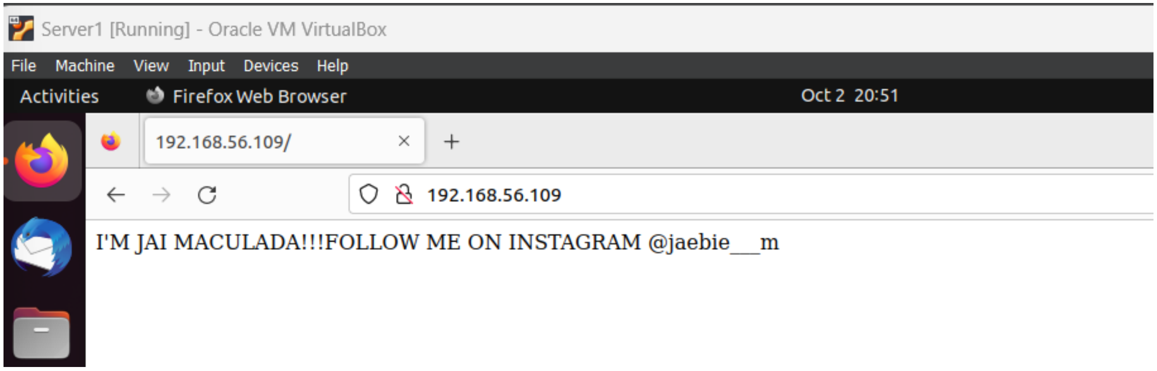
```
TASK [copy default html file for site] ****************************************
*
changed: [192.168.56.102]
changed: [192.168.56.105]
```

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. **It outputs the content of default_site.html once we put the ip address of Server1 and CentOS in the browser.**



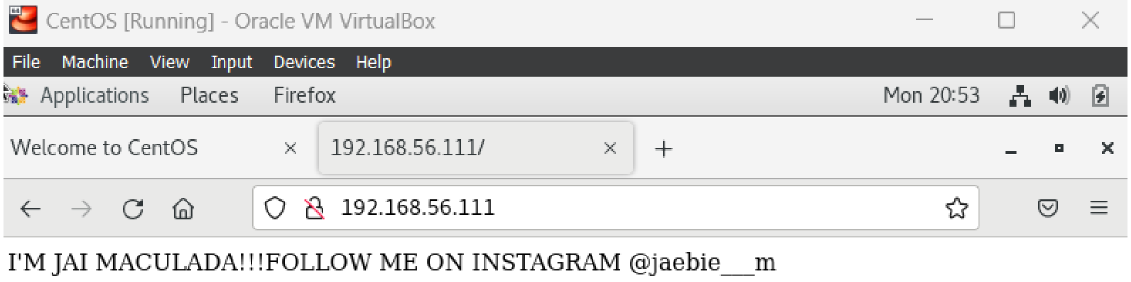
```
jai@workstation:~/CPE232_HOA7.1/files$ cat default_site.html
I'M JAI MACULADA!!!FOLLOW ME ON INSTAGRAM @jaebie___m
```

**SERVER1:**


Server1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Firefox Web Browser
Oct 2 20:51
192.168.56.109/
192.168.56.109

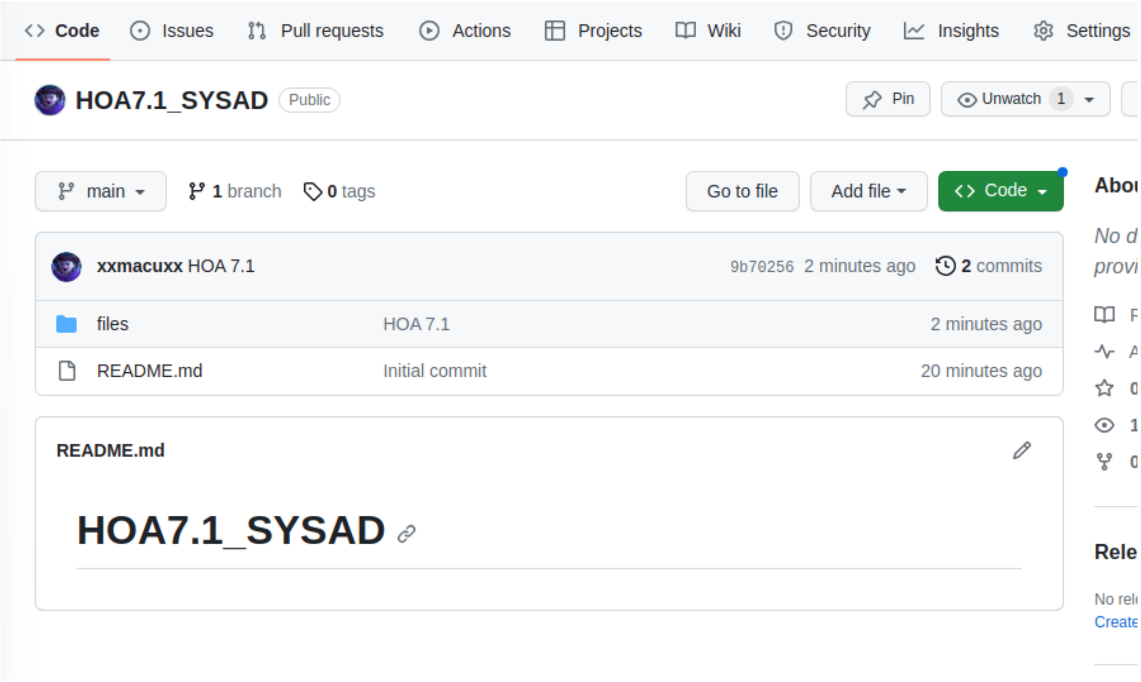
I'M JAI MACULADA!!!FOLLOW ME ON INSTAGRAM @jaebie__m

**CENTOS:**


CentOS [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places Firefox
Mon 20:53
Welcome to CentOS
192.168.56.111/
192.168.56.111

I'M JAI MACULADA!!!FOLLOW ME ON INSTAGRAM @jaebie__m

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



HOA7.1_SYSAD

xxmacuxx HOA 7.1 9b70256 2 minutes ago 2 commits

| | | |
|---|---|---|
| files | HOA 7.1 | 2 minutes ago |
| README.md | Initial commit | 20 minutes ago |

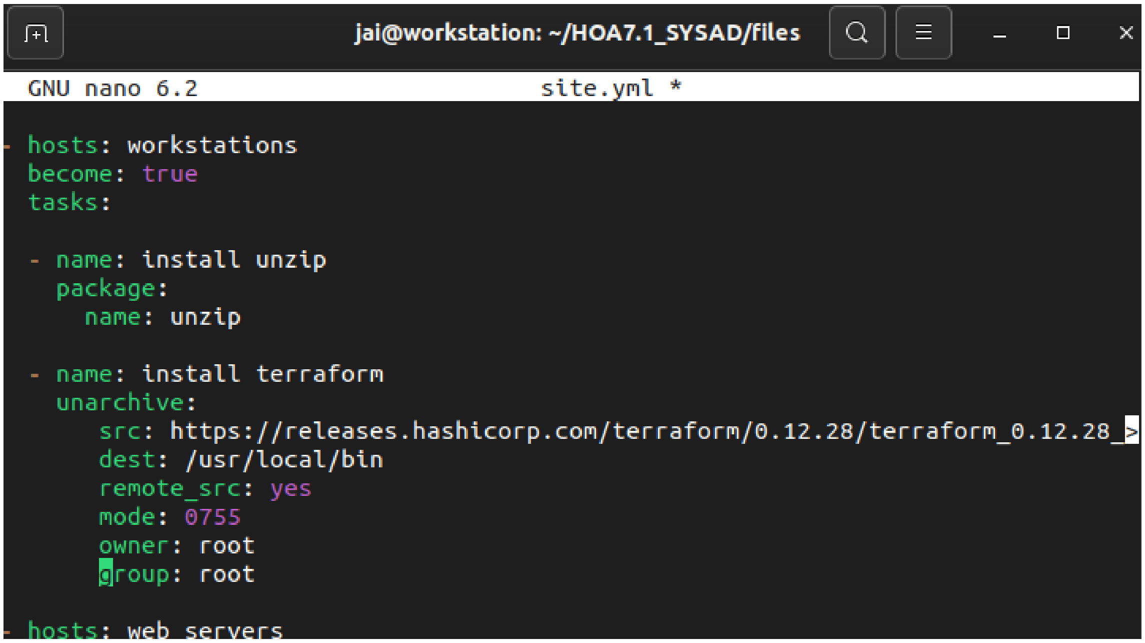
README.md

HOA7.1_SYSAD

**Task 2: Download a file and extract it to a remote server**

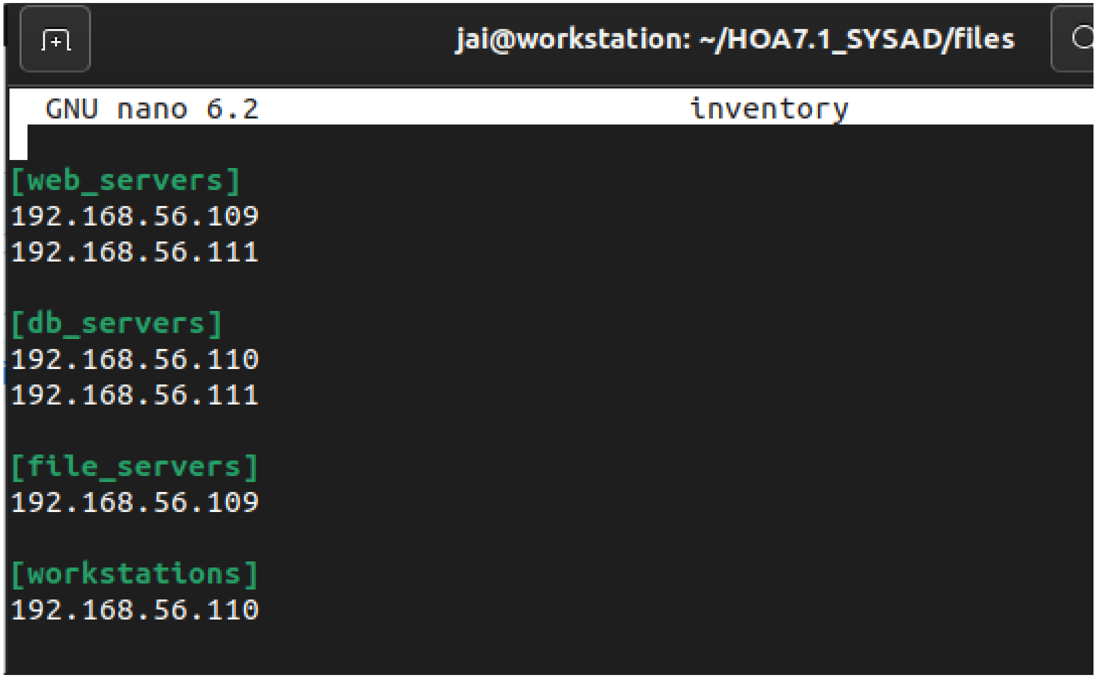
1. Edit the site.yml. Just before the web_servers play, create a new play:
   - hosts: workstations
     become: true

```
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
```



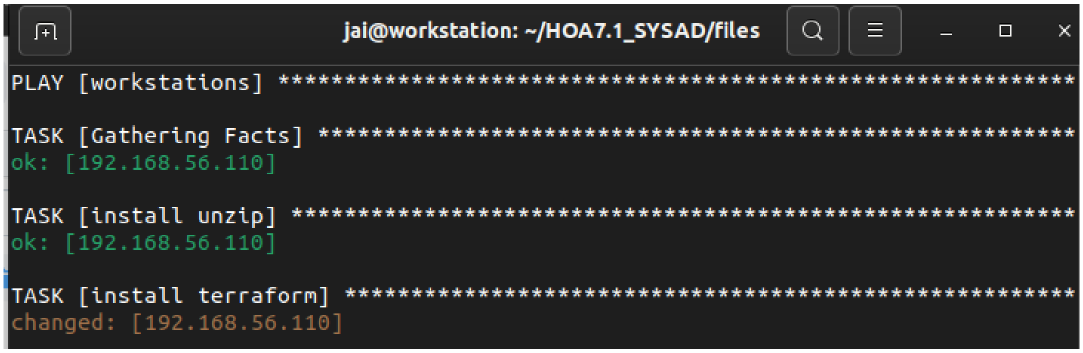
```
- hosts: workstations
  become: true
  tasks:

  - name: install unzip
    package:
      name: unzip

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

- hosts: web_servers
```

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

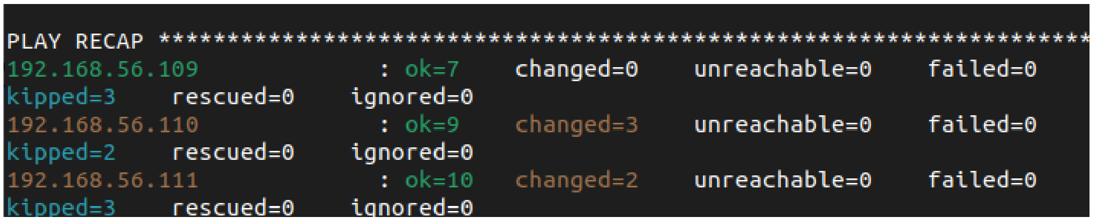


```
 GNU nano 6.2                              inventory

[web_servers]
192.168.56.109
192.168.56.111

[db_servers]
192.168.56.110
192.168.56.111

[file_servers]
192.168.56.109

[workstations]
192.168.56.110
```

3. Run the playbook. Describe the output. **It ran successfully.**

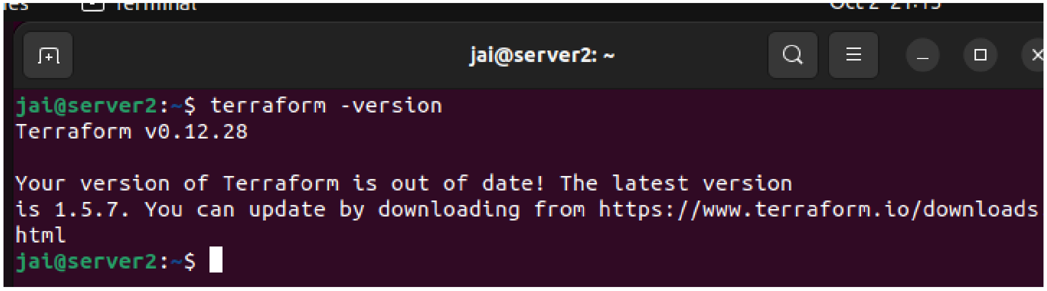


```
PLAY [workstations] *************************************************************

TASK [Gathering Facts] **********************************************************
ok: [192.168.56.110]

TASK [install unzip] ************************************************************
ok: [192.168.56.110]

TASK [install terraform] ********************************************************
changed: [192.168.56.110]
```



```
PLAY RECAP ***********************************************************************
192.168.56.109             : ok=7    changed=0    unreachable=0    failed=0
kipped=3    rescued=0    ignored=0
192.168.56.110             : ok=9    changed=3    unreachable=0    failed=0
kipped=2    rescued=0    ignored=0
192.168.56.111             : ok=10   changed=2    unreachable=0    failed=0
kipped=3    rescued=0    ignored=0
```

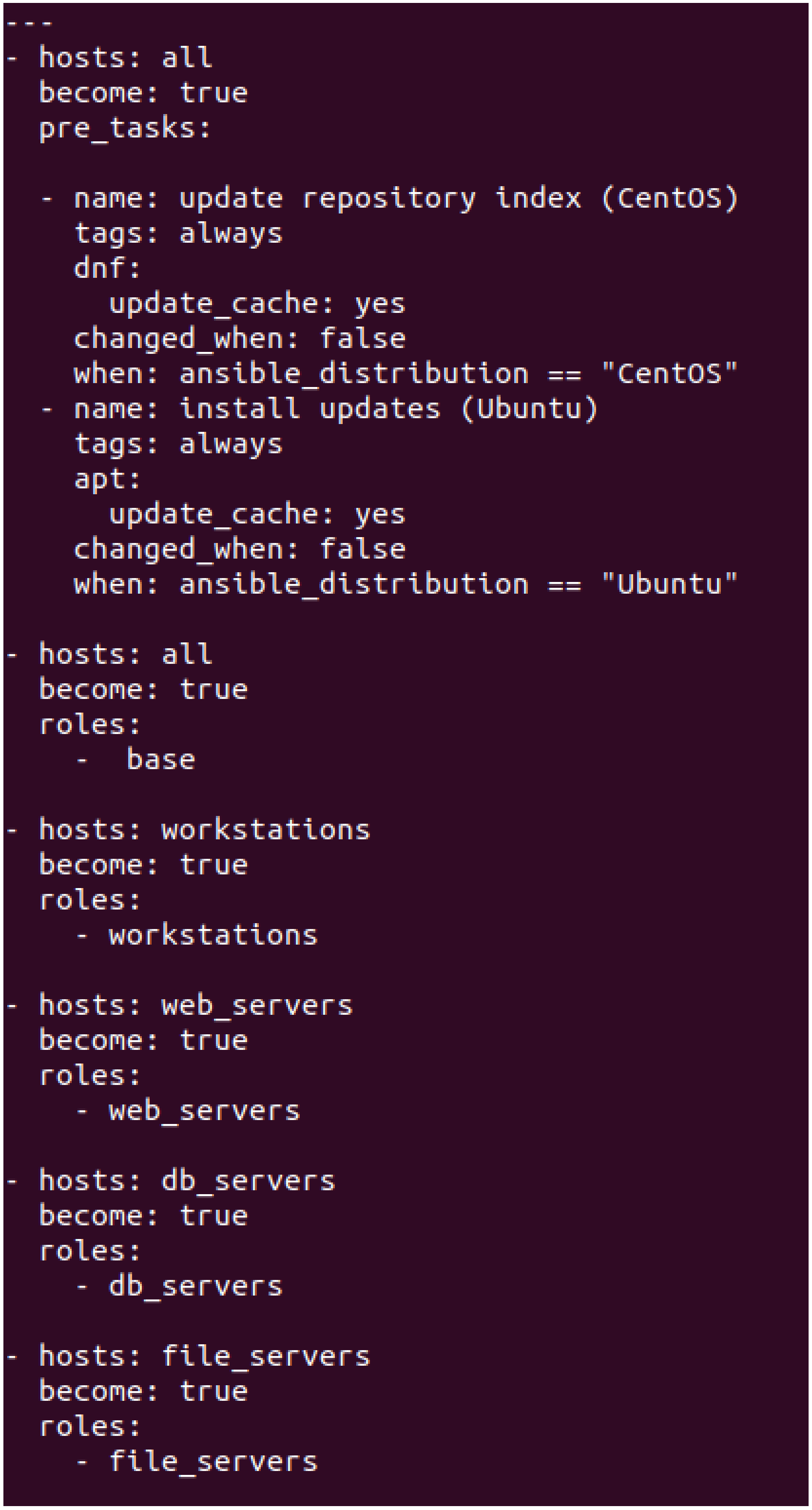
4. On the Ubuntu remote workstation, type terraform to verify installation of terraform. Describe the output. **The terraform that was installed is version 0.12.28.**



```
jai@server2:~$ terraform -version
Terraform v0.12.28

Your version of Terraform is out of date! The latest version
is 1.5.7. You can update by downloading from https://www.terraform.io/downloads
html
jai@server2:~$ 
```

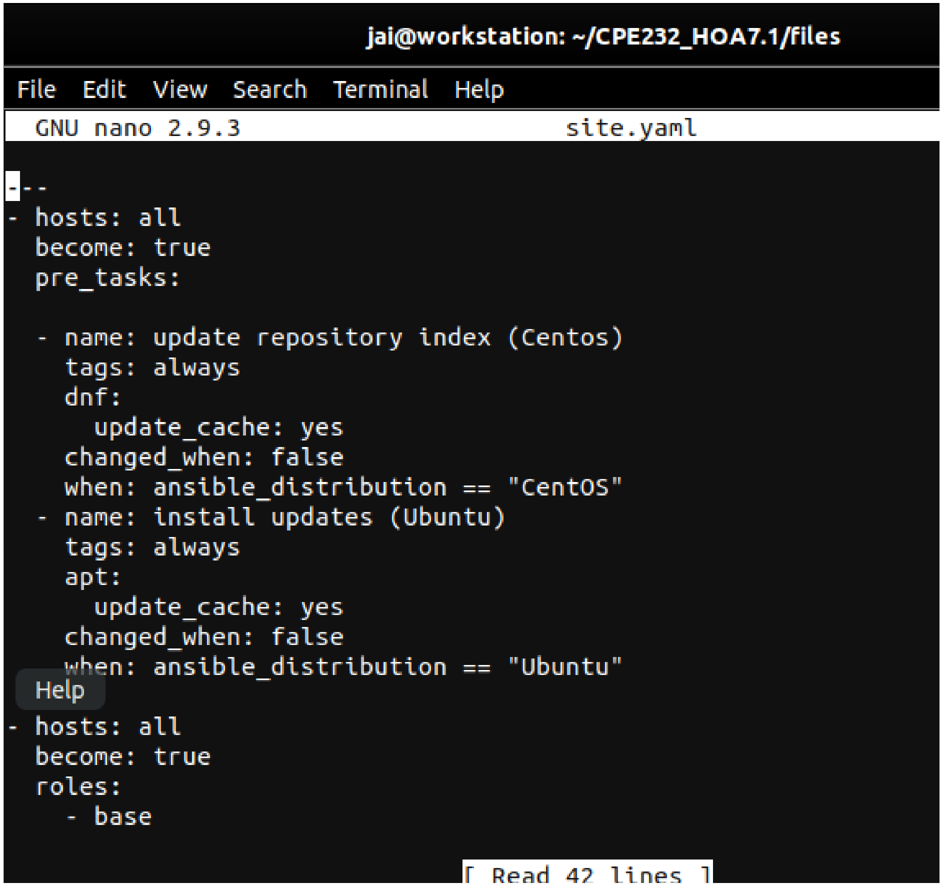
### Task 3: Create roles

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)

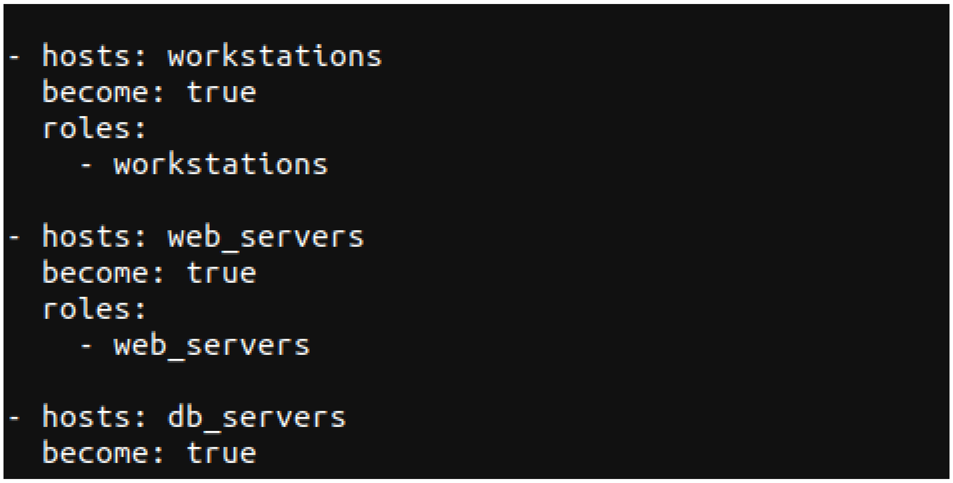


```
---
- hosts: all
  become: true
  pre_tasks:

  - 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.


jai@workstation: ~/CPE232_HOA7.1/files
File Edit View Search Terminal Help
Help

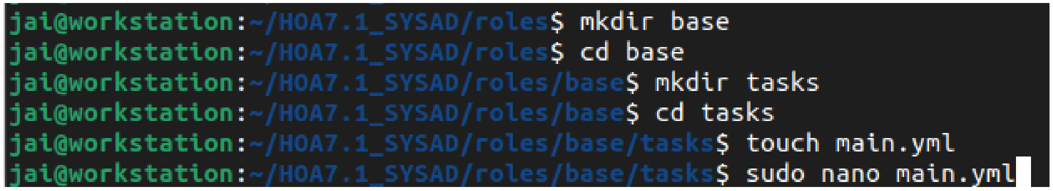
```
 GNU nano 2.9.3                      site.yaml

---
- 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

                               [ Read 42 lines ]
```



```
- hosts: workstations
  become: true
  roles:
    - workstations

- hosts: web_servers
  become: true
  roles:
    - web_servers

- hosts: db_servers
  become: true
```

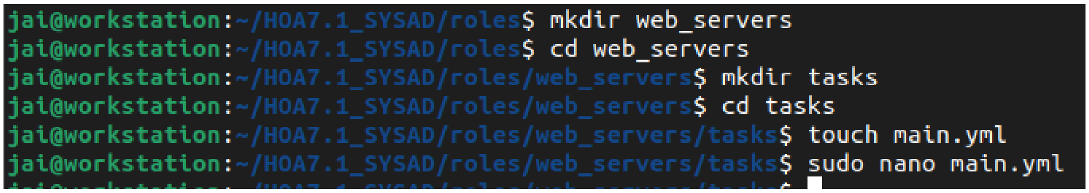
2. Under the same directory, create a new directory and name it roles. Enter the roles directory and create new directories: base, web_servers, file_servers, db_servers and workstations. For each directory, create a directory and name it tasks.

**BASE**



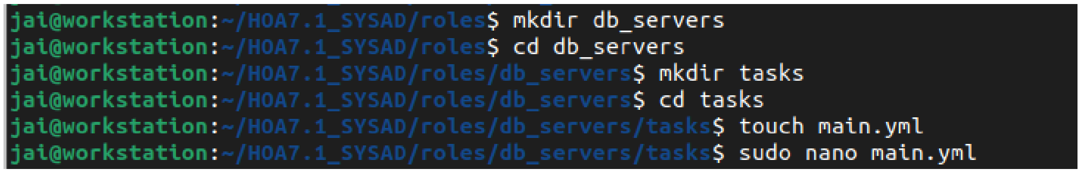
```
jai@workstation:~/HOA7.1_SYSAD/roles$ mkdir base
jai@workstation:~/HOA7.1_SYSAD/roles$ cd base
jai@workstation:~/HOA7.1_SYSAD/roles/base$ mkdir tasks
jai@workstation:~/HOA7.1_SYSAD/roles/base$ cd tasks
jai@workstation:~/HOA7.1_SYSAD/roles/base/tasks$ touch main.yml
jai@workstation:~/HOA7.1_SYSAD/roles/base/tasks$ sudo nano main.yml
```

## WEB_SERVERS



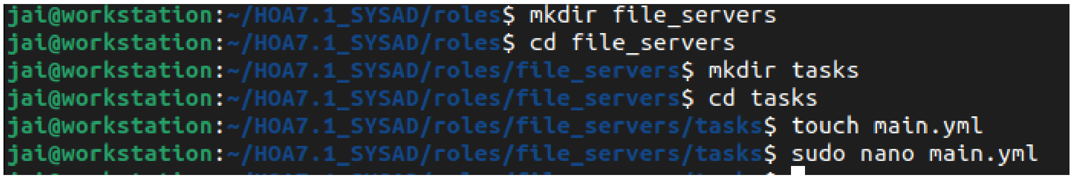
```
jai@workstation:~/HOA7.1_SYSAD/roles$ mkdir web_servers
jai@workstation:~/HOA7.1_SYSAD/roles$ cd web_servers
jai@workstation:~/HOA7.1_SYSAD/roles/web_servers$ mkdir tasks
jai@workstation:~/HOA7.1_SYSAD/roles/web_servers$ cd tasks
jai@workstation:~/HOA7.1_SYSAD/roles/web_servers/tasks$ touch main.yml
jai@workstation:~/HOA7.1_SYSAD/roles/web_servers/tasks$ sudo nano main.yml
```

## DB_SERVERS



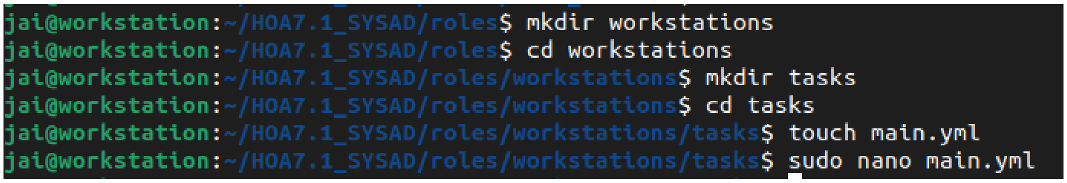
```
jai@workstation:~/HOA7.1_SYSAD/roles$ mkdir db_servers
jai@workstation:~/HOA7.1_SYSAD/roles$ cd db_servers
jai@workstation:~/HOA7.1_SYSAD/roles/db_servers$ mkdir tasks
jai@workstation:~/HOA7.1_SYSAD/roles/db_servers$ cd tasks
jai@workstation:~/HOA7.1_SYSAD/roles/db_servers/tasks$ touch main.yml
jai@workstation:~/HOA7.1_SYSAD/roles/db_servers/tasks$ sudo nano main.yml
```

## FILE_SERVERS



```
jai@workstation:~/HOA7.1_SYSAD/roles$ mkdir file_servers
jai@workstation:~/HOA7.1_SYSAD/roles$ cd file_servers
jai@workstation:~/HOA7.1_SYSAD/roles/file_servers$ mkdir tasks
jai@workstation:~/HOA7.1_SYSAD/roles/file_servers$ cd tasks
jai@workstation:~/HOA7.1_SYSAD/roles/file_servers/tasks$ touch main.yml
jai@workstation:~/HOA7.1_SYSAD/roles/file_servers/tasks$ sudo nano main.yml
```

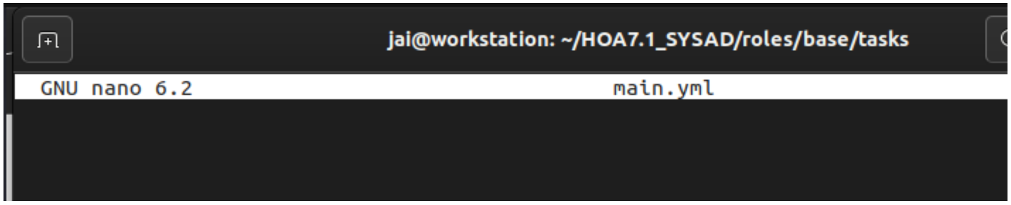
## WORKSTATIONS



```
jai@workstation:~/HOA7.1_SYSAD/roles$ mkdir workstations
jai@workstation:~/HOA7.1_SYSAD/roles$ cd workstations
jai@workstation:~/HOA7.1_SYSAD/roles/workstations$ mkdir tasks
jai@workstation:~/HOA7.1_SYSAD/roles/workstations$ cd tasks
jai@workstation:~/HOA7.1_SYSAD/roles/workstations/tasks$ touch main.yml
jai@workstation:~/HOA7.1_SYSAD/roles/workstations/tasks$ sudo nano main.yml
```

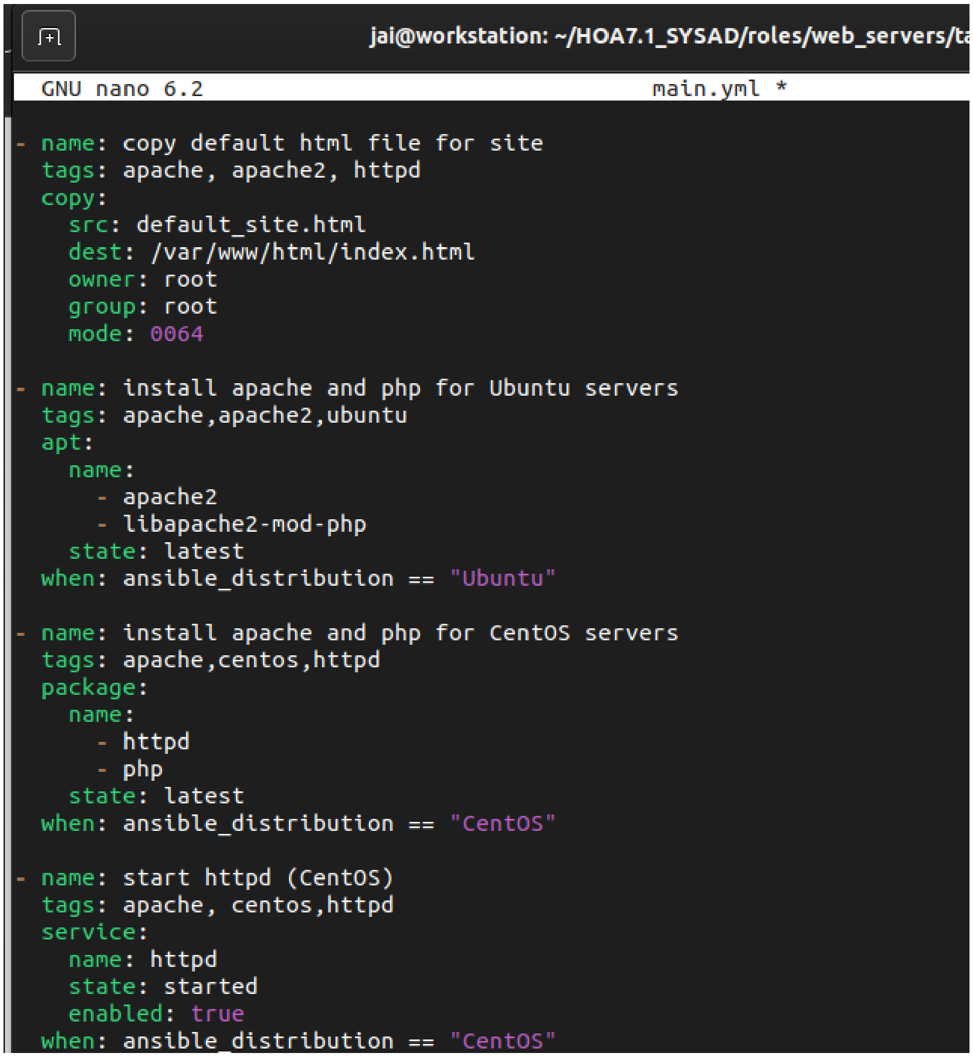
3. Go to tasks for all directory and create a file. Name it main.yml. In each of the tasks for all directories, copy and paste the code from the old site.yml file. Show all contents of main.yml files for all tasks.

## BASE



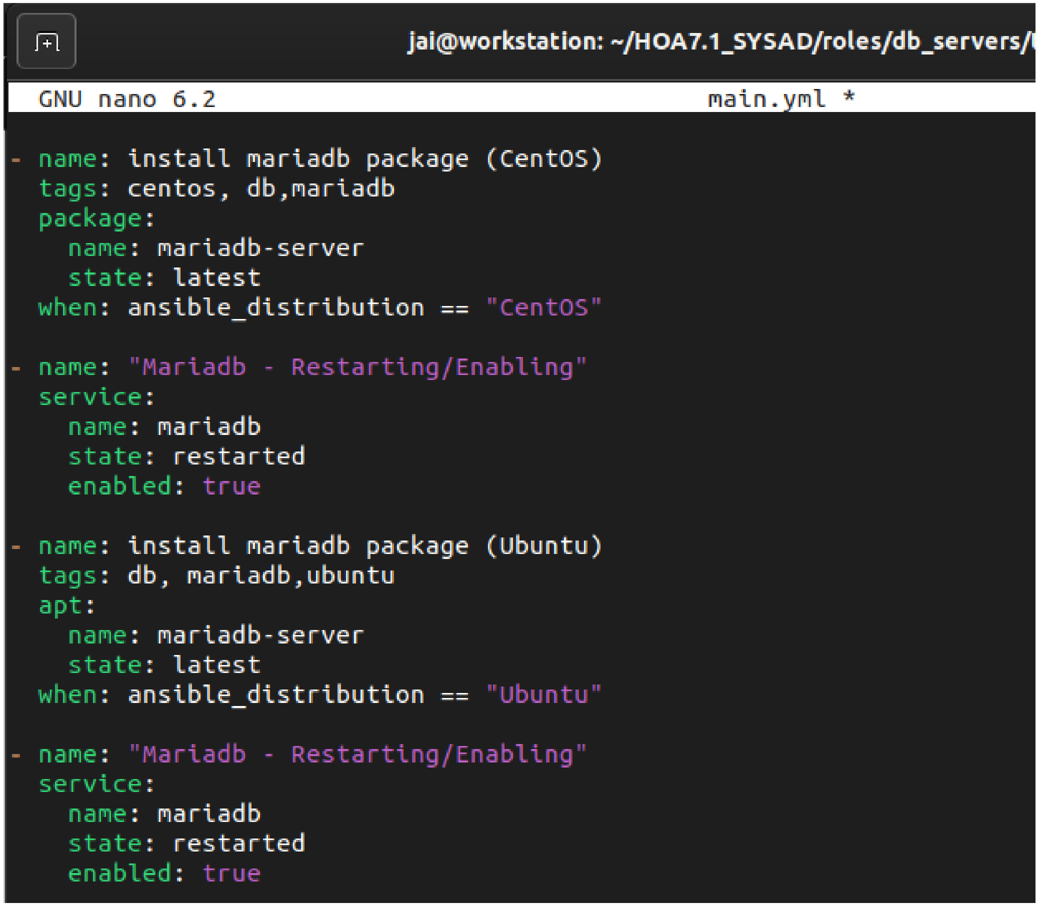
```
jai@workstation: ~/HOA7.1_SYSAD/roles/base/tasks
GNU nano 6.2                                        main.yml
```

## WEB_SERVERS



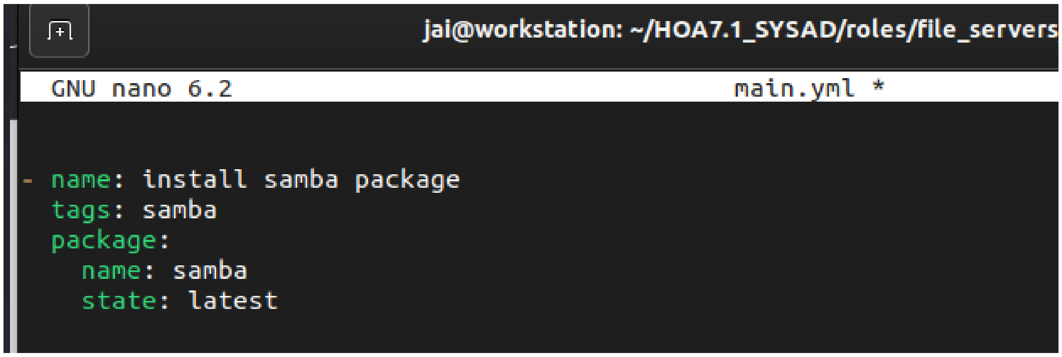
```
jai@workstation: ~/HOA7.1_SYSAD/roles/web_servers/ta
GNU nano 6.2                                        main.yml *

- 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: 0064

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

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

## DB_SERVERS



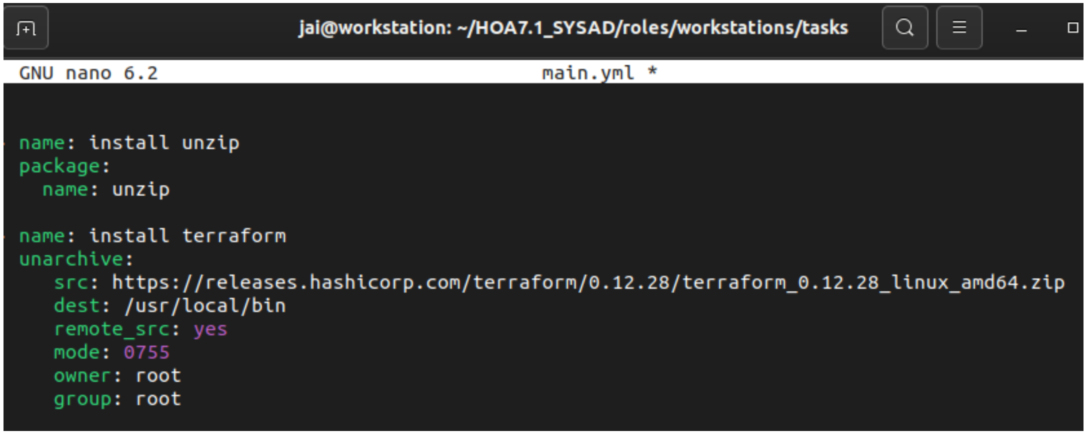
```
GNU nano 6.2                                        main.yml *

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

- name: "Mariadb - Restarting/Enabling"
  service:
    name: mariadb
    state: restarted
    enabled: true
```

### FILE_SERVERS

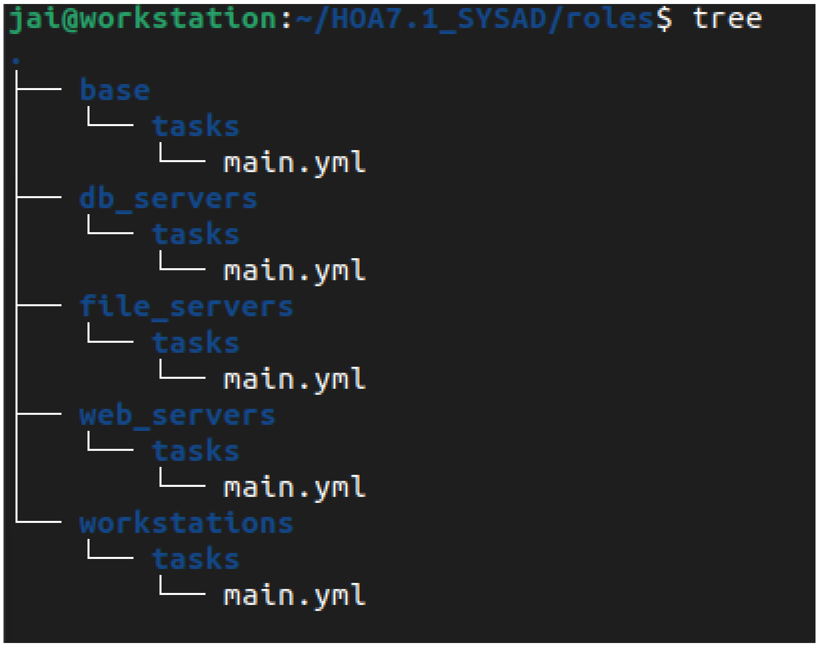


```
GNU nano 6.2                                        main.yml *


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

### WORKSTATIONS

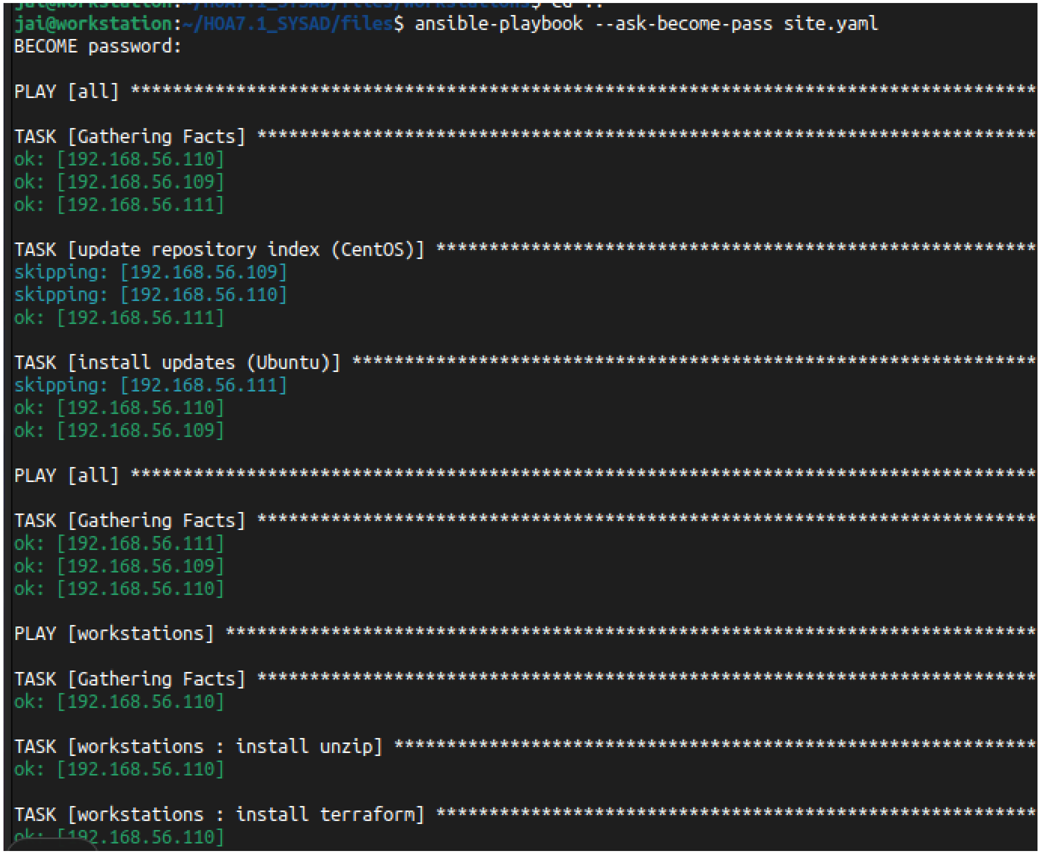


```
GNU nano 6.2                                    main.yml *


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
```

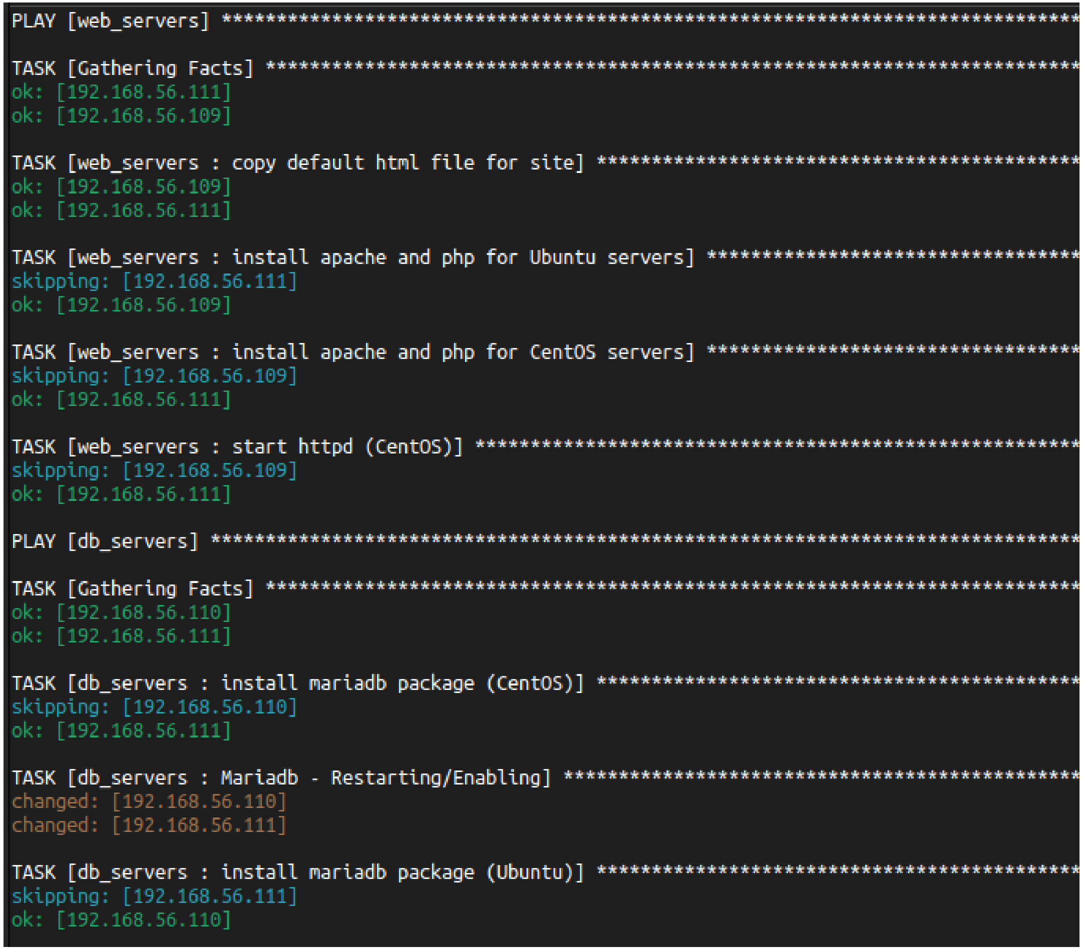


```
jai@workstation:~/HOA7.1_SYSAD/roles$ tree
.
├── base
│   └── tasks
│       └── main.yml
├── db_servers
│   └── tasks
│       └── main.yml
├── file_servers
│   └── tasks
│       └── main.yml
├── web_servers
│   └── tasks
│       └── main.yml
└── workstations
    └── tasks
        └── main.yml
```

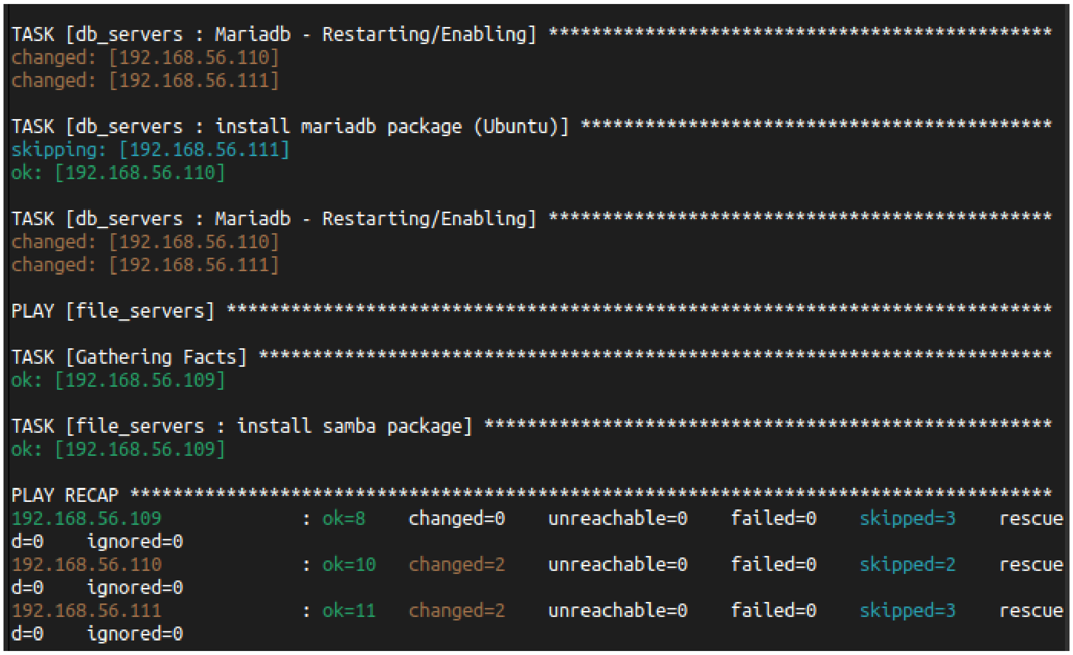
4. Run the site.yml playbook and describe the output. **It has the same output to the old site.yml, the only difference is that the roles were assigned.**



```
jai@workstation:~/HOA7.1_SYSAD/files/workstations$ cd ..
jai@workstation:~/HOA7.1_SYSAD/files$ ansible-playbook --ask-become-pass site.yaml
BECOME password:

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

TASK [Gathering Facts] **************************************************************************
ok: [192.168.56.110]
ok: [192.168.56.109]
ok: [192.168.56.111]

TASK [update repository index (CentOS)] *********************************************************
skipping: [192.168.56.109]
skipping: [192.168.56.110]
ok: [192.168.56.111]

TASK [install updates (Ubuntu)] *****************************************************************
skipping: [192.168.56.111]
ok: [192.168.56.110]
ok: [192.168.56.109]

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

TASK [Gathering Facts] **************************************************************************
ok: [192.168.56.111]
ok: [192.168.56.109]
ok: [192.168.56.110]

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

TASK [Gathering Facts] **************************************************************************
ok: [192.168.56.110]

TASK [workstations : install unzip] *************************************************************
ok: [192.168.56.110]

TASK [workstations : install terraform] *********************************************************
ok: [192.168.56.110]
```



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

TASK [Gathering Facts] ***************************************************************************
ok: [192.168.56.111]
ok: [192.168.56.109]

TASK [web_servers : copy default html file for site] *********************************************
ok: [192.168.56.109]
ok: [192.168.56.111]

TASK [web_servers : install apache and php for Ubuntu servers] ***********************************
skipping: [192.168.56.111]
ok: [192.168.56.109]

TASK [web_servers : install apache and php for CentOS servers] ***********************************
skipping: [192.168.56.109]
ok: [192.168.56.111]

TASK [web_servers : start httpd (CentOS)] ********************************************************
skipping: [192.168.56.109]
ok: [192.168.56.111]

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

TASK [Gathering Facts] ***************************************************************************
ok: [192.168.56.110]
ok: [192.168.56.111]

TASK [db_servers : install mariadb package (CentOS)] *********************************************
skipping: [192.168.56.110]
ok: [192.168.56.111]

TASK [db_servers : Mariadb - Restarting/Enabling] ************************************************
changed: [192.168.56.110]
changed: [192.168.56.111]

TASK [db_servers : install mariadb package (Ubuntu)] *********************************************
skipping: [192.168.56.111]
ok: [192.168.56.110]
```



```
TASK [db_servers : Mariadb - Restarting/Enabling] ***********************************************
changed: [192.168.56.110]
changed: [192.168.56.111]

TASK [db_servers : install mariadb package (Ubuntu)] ********************************************
skipping: [192.168.56.111]
ok: [192.168.56.110]

TASK [db_servers : Mariadb - Restarting/Enabling] ***********************************************
changed: [192.168.56.110]
changed: [192.168.56.111]

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

TASK [Gathering Facts] **************************************************************************
ok: [192.168.56.109]

TASK [file_servers : install samba package] *****************************************************
ok: [192.168.56.109]

PLAY RECAP **************************************************************************************
192.168.56.109             : ok=8    changed=0    unreachable=0    failed=0    skipped=3    rescue
d=0    ignored=0
192.168.56.110             : ok=10   changed=2    unreachable=0    failed=0    skipped=2    rescue
d=0    ignored=0
192.168.56.111             : ok=11   changed=2    unreachable=0    failed=0    skipped=3    rescue
d=0    ignored=0
```

**GITHUB LINK:** https://github.com/xxmacuxx/HOA7.1_SYSAD.git

**Reflections:**

Answer the following:

1. What is the importance of creating roles?

   In order to automate effectively in an Ubuntu playbook, roles must be created. Roles facilitate task organization and make playbooks modular, improving code reuse. This streamlines teamwork and maintenance while assuring dependable and consistent server configuration, thereby saving time and minimizing errors.

2. What is the importance of managing files?

   In Ubuntu, managing files is crucial for data organization, guaranteeing system reliability, and maximizing storage. It makes it easier to find, use, and organize documents, programs, and configurations. The proper administration of files also improves security by limiting access and permissions. Overall, it maintains the efficiency, security, and organization of your Ubuntu system.

***CONCLUSION:***

After performing this activity, I was able to encounter roles. While testing the playbook, I realizd that tasks requiring complicated automation are made simpler by implementing roles in an Ubuntu playbook. To encourage consistency and lessen code duplication, it enables me to bundle related plays and jobs into reusable modules. This makes playbook upkeep simpler and maks sure that the configuration management process is more streamlined and effective. In essence, playbook creation is streamlined by responsibilities, whch makes it easier to control and more effective. Overall, I had fun performing this activity since this will be very useful in the future HOA's that we'll make.