Name: Christian dave L. Berja	**Date Performed: 18/10/2023**
Course/Section: CPE3S5	**Date Submitted: 18/10/2023**
Instructor: Engr. Roman Richard	**Semester and SY: 1st yr 2023**

Activity 7: Managing Files and Creating Roles in Ansible

1. Objectives:

1.1 Manage files in remote servers

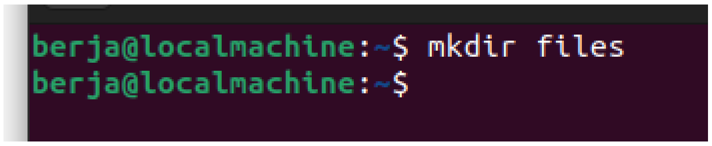
1.2 Implement roles in ansible

2. Discussion:

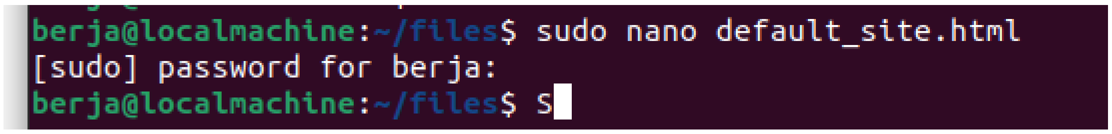
In this activity, we look at the concept of copying a file to a server. We are going to create a file into our git repository and use Ansible to grab that file and put it into a particular place so that we could do things like customize a default website, or maybe install a default configuration file.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

Task 1: Create a file and copy it to remote servers

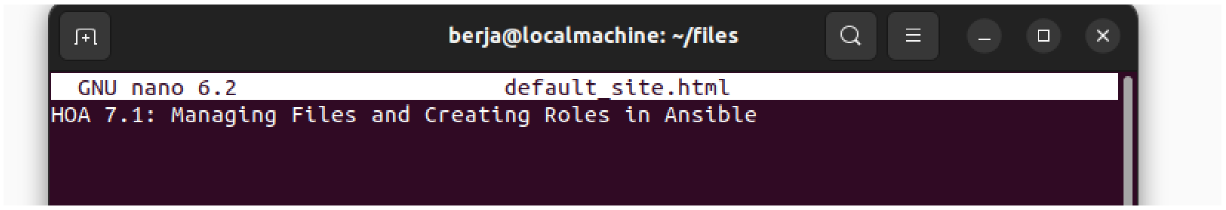
1. Using the previous directory we created, create a directory, and named it "*files*." Create a file inside that directory and name it "*default_site.html*." Edit the file and put basic HTML syntax. Any content will do, as long as it will display text later. Save the file and exit.



```
berja@localmachine:~$ mkdir files
berja@localmachine:~$
```



```
berja@localmachine:~/files$ sudo nano default_site.html
[sudo] password for berja:
berja@localmachine:~/files$ S
```



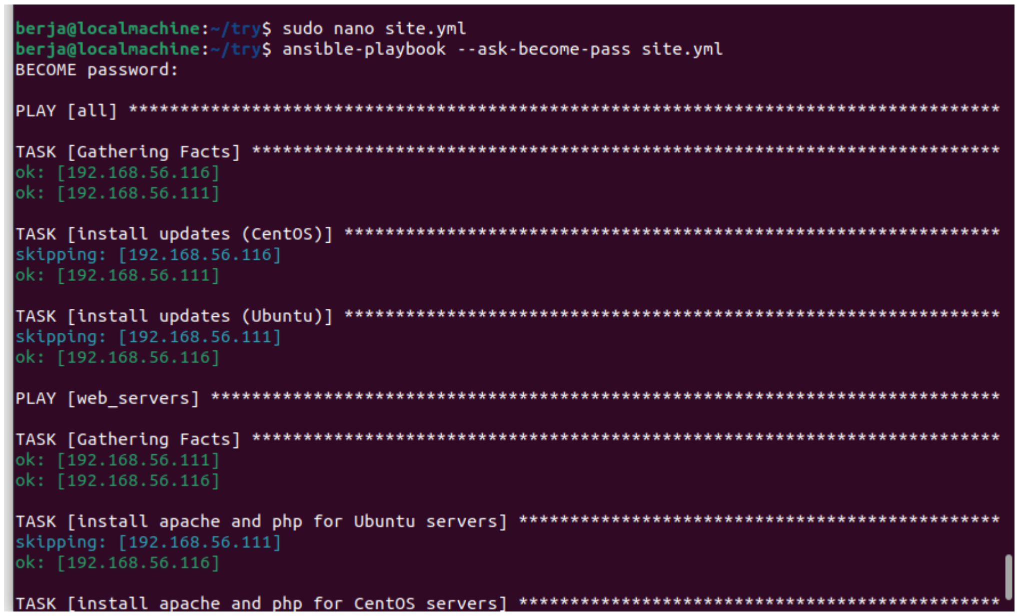
```
berja@localmachine: ~/files
  GNU nano 6.2                    default_site.html
HOA 7.1: Managing Files and Creating Roles in Ansible
```

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:

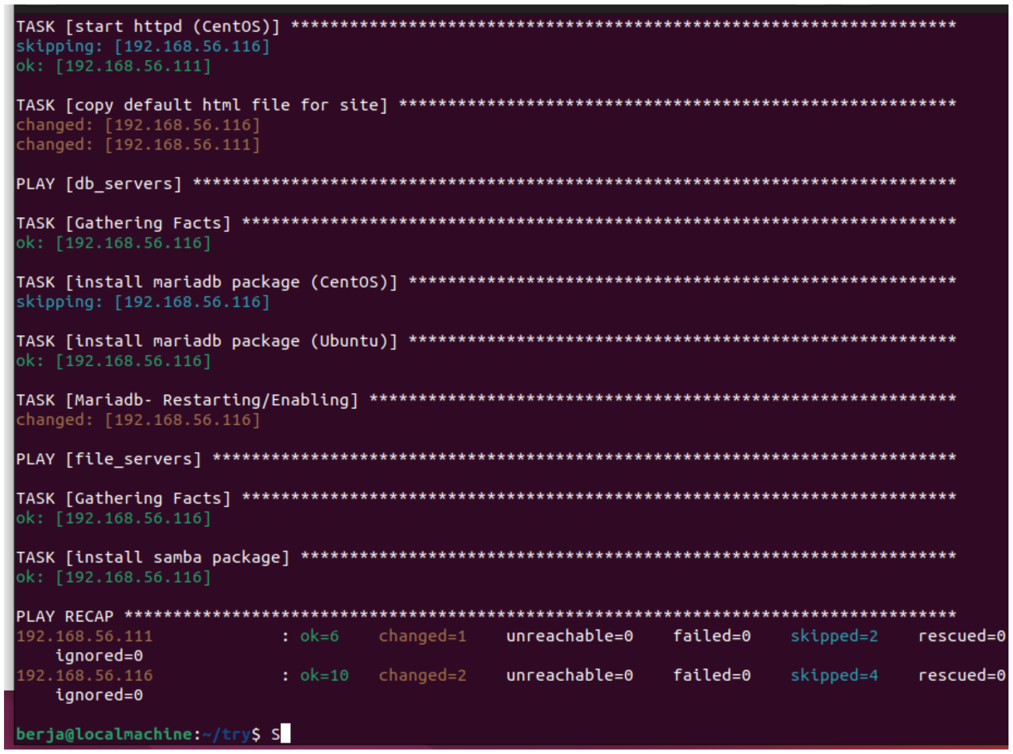
```
- name: copy default html file for site

  tags: apache, apache2, httpd
  copy:
    src: default_site.html
    dest: /var/www/html/index.html
```

owner: root
group: root
mode: 0644

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



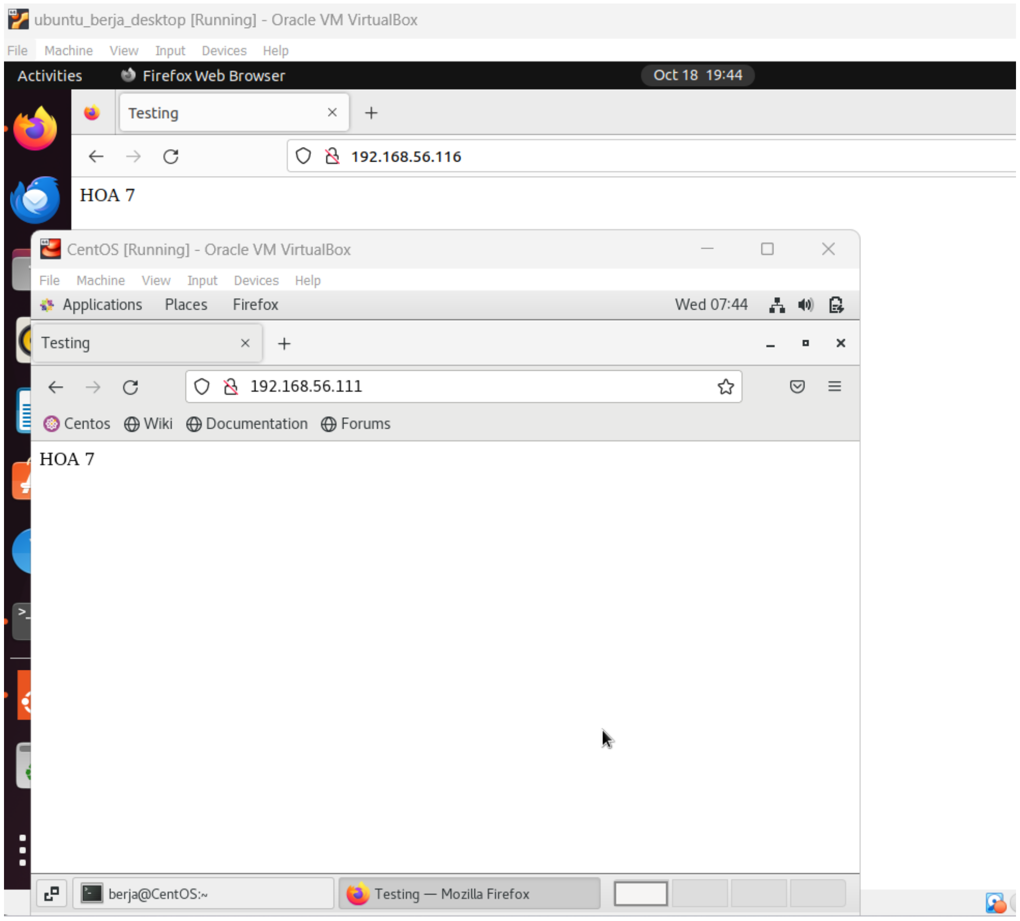
```
berja@localmachine:~/try$ sudo nano site.yml
berja@localmachine:~/try$ ansible-playbook --ask-become-pass site.yml
BECOME password:

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

TASK [Gathering Facts] ******************************************************************
ok: [192.168.56.116]
ok: [192.168.56.111]

TASK [install updates (CentOS)] *********************************************************
skipping: [192.168.56.116]
ok: [192.168.56.111]

TASK [install updates (Ubuntu)] *********************************************************
skipping: [192.168.56.111]
ok: [192.168.56.116]

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

TASK [Gathering Facts] ******************************************************************
ok: [192.168.56.111]
ok: [192.168.56.116]

TASK [install apache and php for Ubuntu servers] ****************************************
skipping: [192.168.56.111]
ok: [192.168.56.116]

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



```
TASK [start httpd (CentOS)] *************************************************************
skipping: [192.168.56.116]
ok: [192.168.56.111]

TASK [copy default html file for site] **************************************************
changed: [192.168.56.116]
changed: [192.168.56.111]

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

TASK [Gathering Facts] ******************************************************************
ok: [192.168.56.116]

TASK [install mariadb package (CentOS)] *************************************************
skipping: [192.168.56.116]

TASK [install mariadb package (Ubuntu)] *************************************************
ok: [192.168.56.116]

TASK [Mariadb- Restarting/Enabling] *****************************************************
changed: [192.168.56.116]

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

TASK [Gathering Facts] ******************************************************************
ok: [192.168.56.116]

TASK [install samba package] ************************************************************
ok: [192.168.56.116]

PLAY RECAP ******************************************************************************
192.168.56.111             : ok=6    changed=1    unreachable=0    failed=0    skipped=2    rescued=0
    ignored=0
192.168.56.116             : ok=10   changed=2    unreachable=0    failed=0    skipped=4    rescued=0
    ignored=0

berja@localmachine:~/try$ S
```

- **After inputting the code that is given inside the web_servers, it can be seen that for both CentOS and Ubuntu they have been changed and added to ansible.**

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.

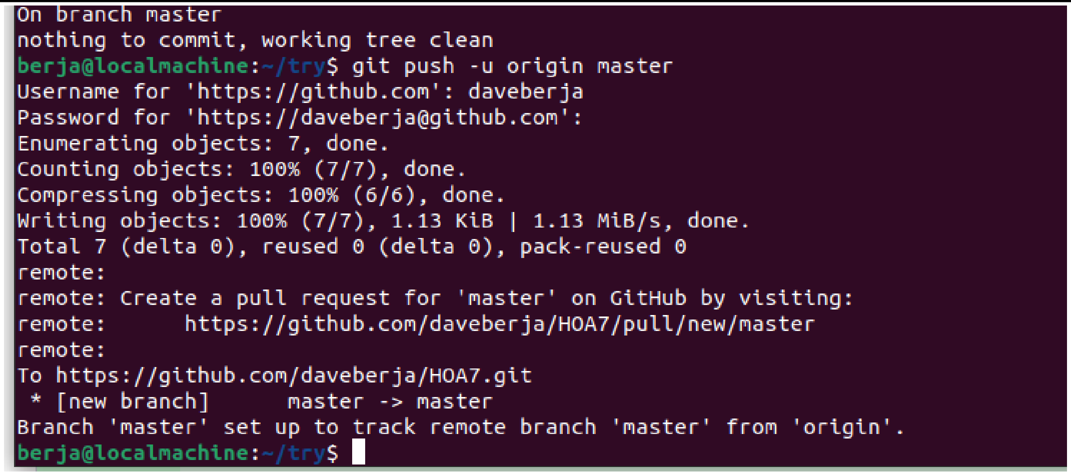

ubuntu_berja_desktop [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Firefox Web Browser
Oct 18 19:44
Testing
192.168.56.116
CentOS [Running] - Oracle VM VirtualBox
Applications Places Firefox
Wed 07:44
192.168.56.111
Centos Wiki Documentation Forums
berja@CentOS:~
Testing — Mozilla Firefox

HOA 7

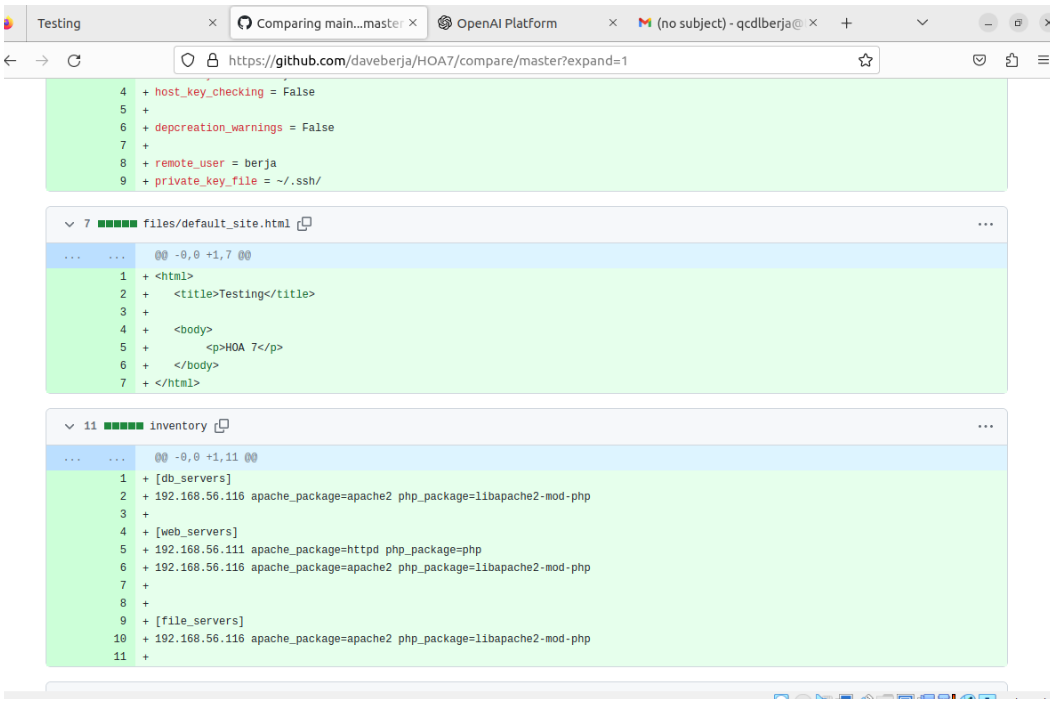
HOA 7

- **After adding the code it can be seen when inputting the IP from the original servers inside the website the code that i inputted showed when i run the ip**

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



```
On branch master
nothing to commit, working tree clean
berja@localmachine:~/try$ git push -u origin master
Username for 'https://github.com': daveberja
Password for 'https://daveberja@github.com':
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Compressing objects: 100% (6/6), done.
Writing objects: 100% (7/7), 1.13 KiB | 1.13 MiB/s, done.
Total 7 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:      https://github.com/daveberja/HOA7/pull/new/master
remote:
To https://github.com/daveberja/HOA7.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
berja@localmachine:~/try$
```



Testing
Comparing main...master
OpenAI Platform
(no subject) - qcdlberja@
https://github.com/daveberja/HOA7/compare/master?expand=1

```
+ host_key_checking = False
+
+ depcreation_warnings = False
+
+ remote_user = berja
+ private_key_file = ~/.ssh/
```

7 files/default_site.html

```
@@ -0,0 +1,7 @@
+ <html>
+    <title>Testing</title>
+
+    <body>
+          <p>HOA 7</p>
+    </body>
+ </html>
```

11 inventory

```
@@ -0,0 +1,11 @@
+ [db_servers]
+ 192.168.56.116 apache_package=apache2 php_package=libapache2-mod-php
+
+ [web_servers]
+ 192.168.56.111 apache_package=httpd php_package=php
+ 192.168.56.116 apache_package=apache2 php_package=libapache2-mod-php
+
+
+ [file_servers]
+ 192.168.56.116 apache_package=apache2 php_package=libapache2-mod-php
+
```

- **After committing my directories and ansible file was inputted inside the GitHub**

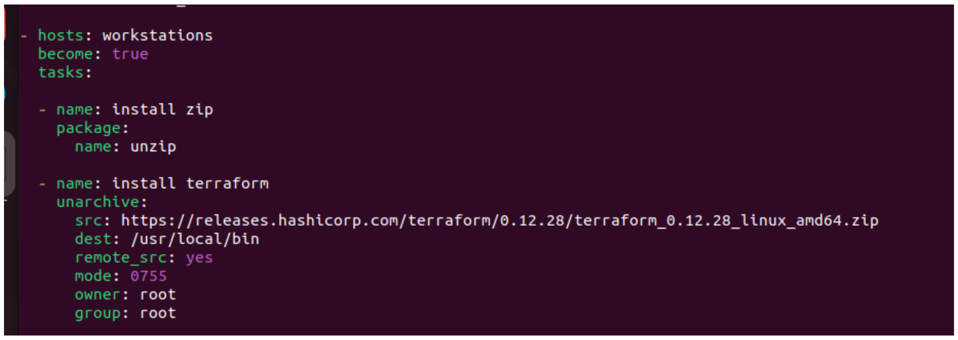
Task 2: Download a file and extract it to a remote server

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

     - name: install unzip

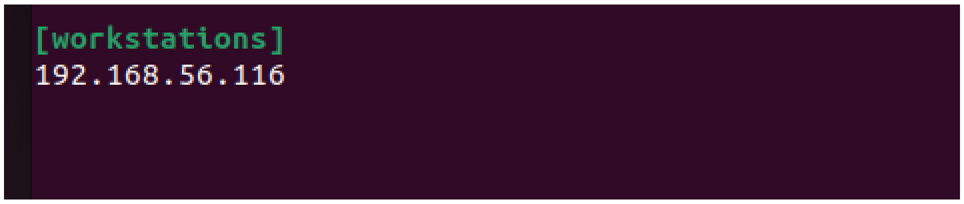
package:
    name: unzip

- 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



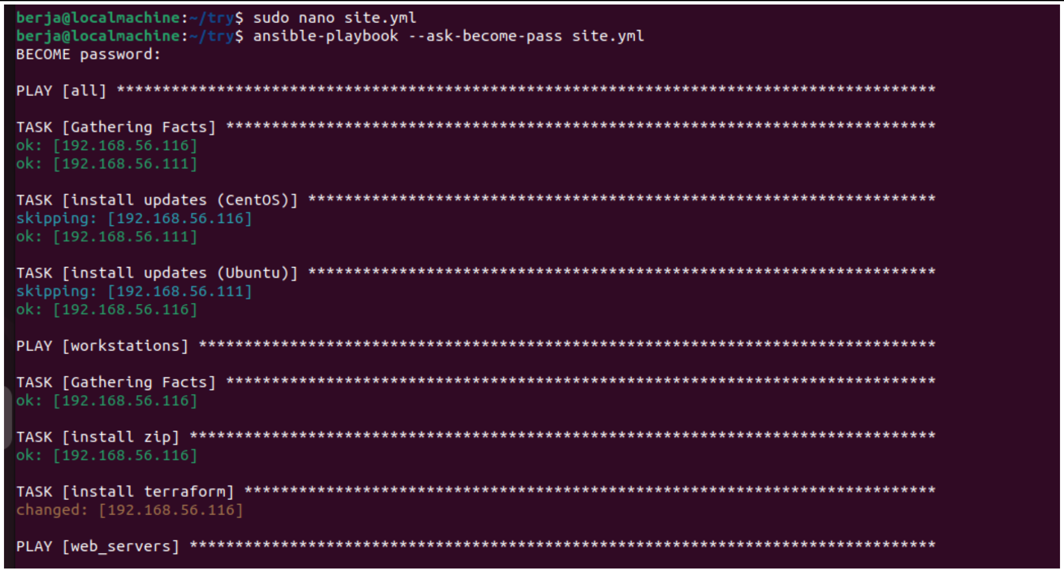
```
- hosts: workstations
  become: true
  tasks:

  - name: install zip
    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.

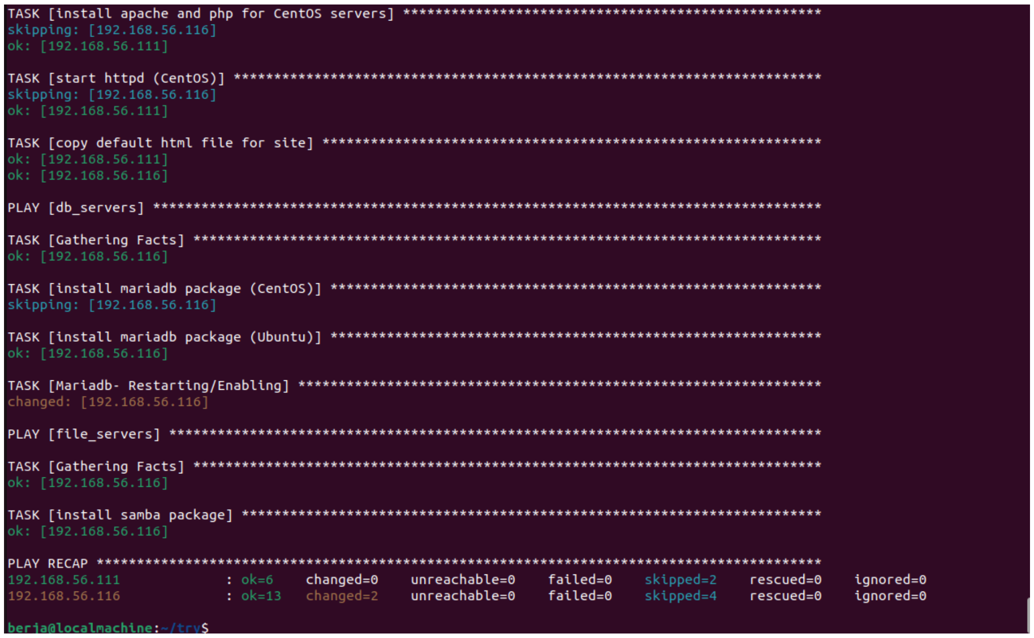


```
[workstations]
192.168.56.116
```

3. Run the playbook. Describe the output.



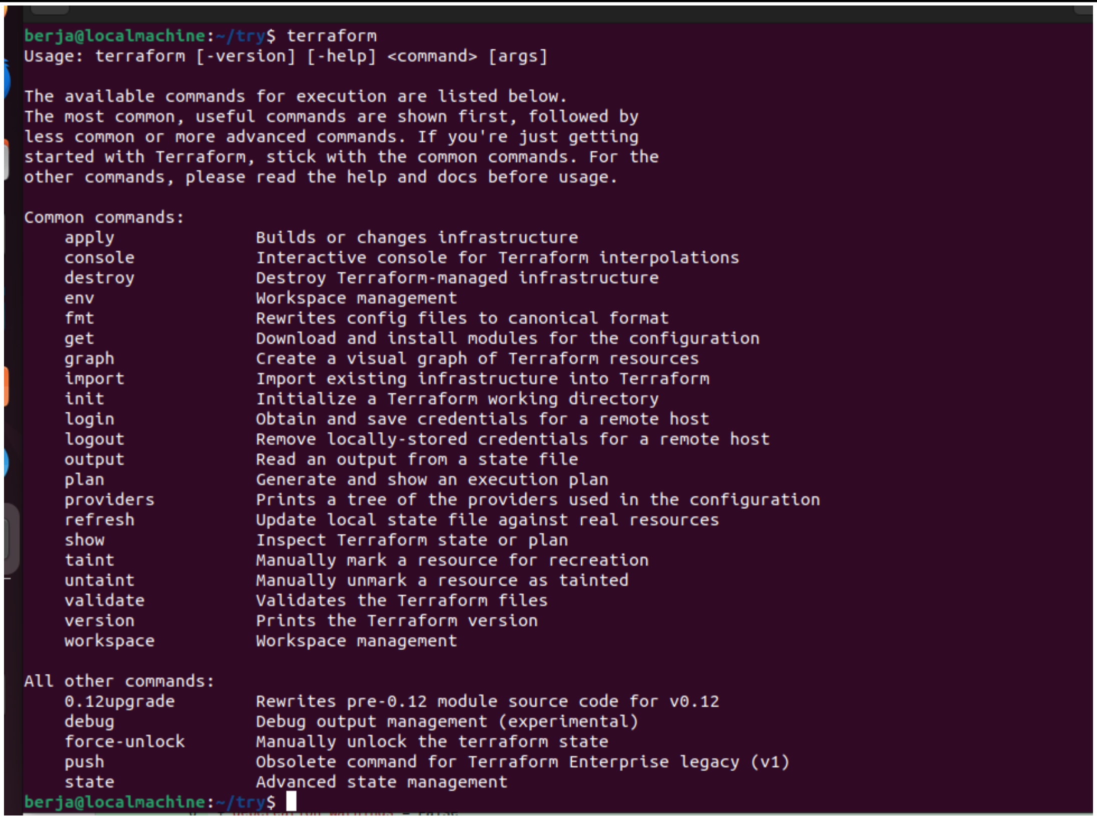
```
berja@localmachine:~/try$ sudo nano site.yml
berja@localmachine:~/try$ ansible-playbook --ask-become-pass site.yml
BECOME password:

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

TASK [Gathering Facts] *************************************************************************
ok: [192.168.56.116]
ok: [192.168.56.111]

TASK [install updates (CentOS)] ****************************************************************
skipping: [192.168.56.116]
ok: [192.168.56.111]

TASK [install updates (Ubuntu)] ****************************************************************
skipping: [192.168.56.111]
ok: [192.168.56.116]

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

TASK [Gathering Facts] *************************************************************************
ok: [192.168.56.116]

TASK [install zip] *****************************************************************************
ok: [192.168.56.116]

TASK [install terraform] ***********************************************************************
changed: [192.168.56.116]

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



```
TASK [install apache and php for CentOS servers] ***************************************************
skipping: [192.168.56.116]
ok: [192.168.56.111]

TASK [start httpd (CentOS)] ************************************************************************
skipping: [192.168.56.116]
ok: [192.168.56.111]

TASK [copy default html file for site] *************************************************************
ok: [192.168.56.111]
ok: [192.168.56.116]

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

TASK [Gathering Facts] *****************************************************************************
ok: [192.168.56.116]

TASK [install mariadb package (CentOS)] ************************************************************
skipping: [192.168.56.116]

TASK [install mariadb package (Ubuntu)] ************************************************************
ok: [192.168.56.116]

TASK [Mariadb- Restarting/Enabling] ****************************************************************
changed: [192.168.56.116]

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

TASK [Gathering Facts] *****************************************************************************
ok: [192.168.56.116]

TASK [install samba package] ***********************************************************************
ok: [192.168.56.116]

PLAY RECAP *****************************************************************************************
192.168.56.111             : ok=6    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.56.116             : ok=13   changed=2    unreachable=0    failed=0    skipped=4    rescued=0    ignored=0

berja@localmachine:~/try$
```

- **The task was successfully installed zip and changed the install terraform which installed it also**

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

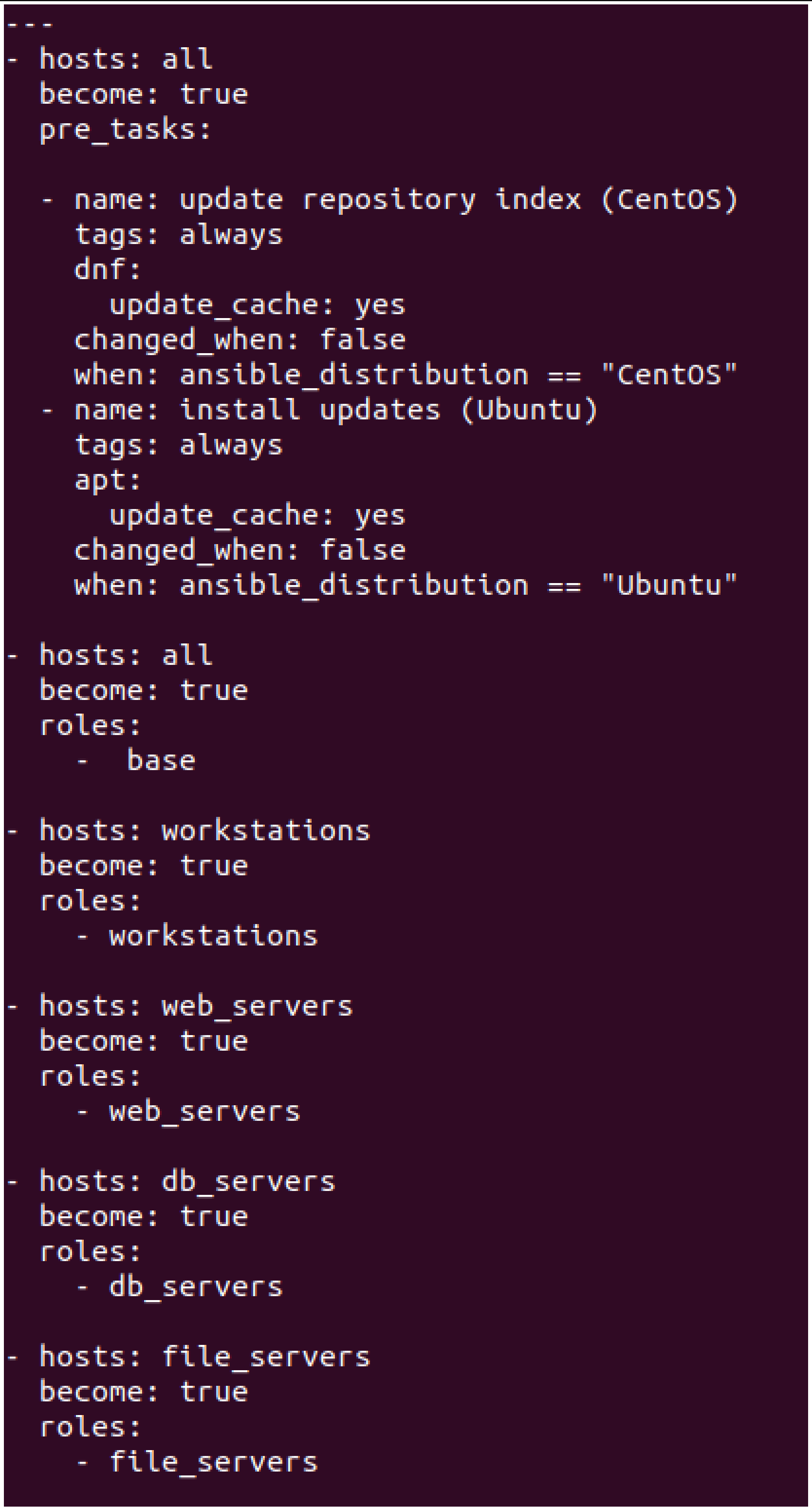


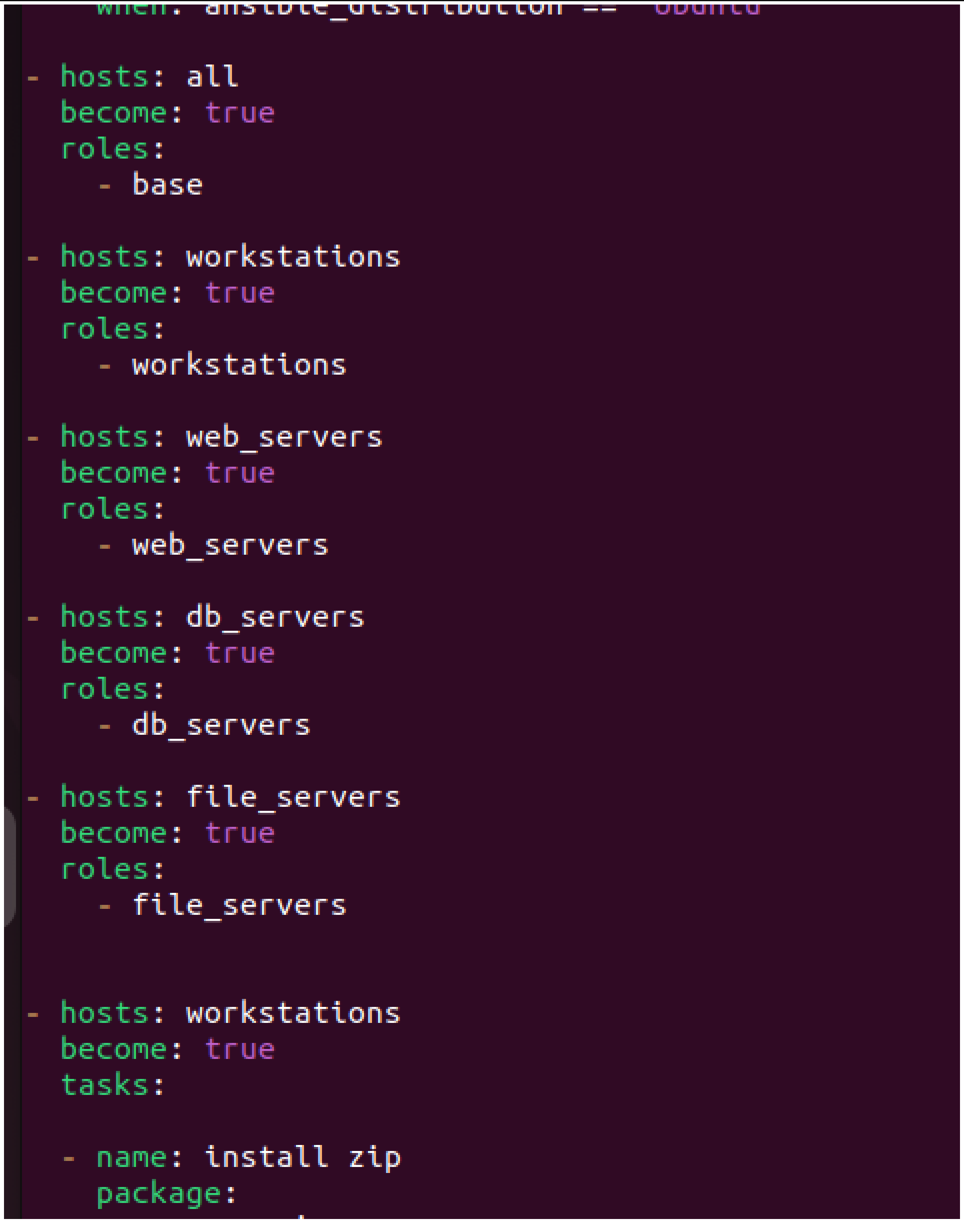
```
berja@localmachine:~/try$ terraform
Usage: terraform [-version] [-help] <command> [args]

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
    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
berja@localmachine:~/try$
```

- **After installing the terraform, by inputting the terraform code this showed all the commands inside the terraform itself.**

Task 3: Create roles

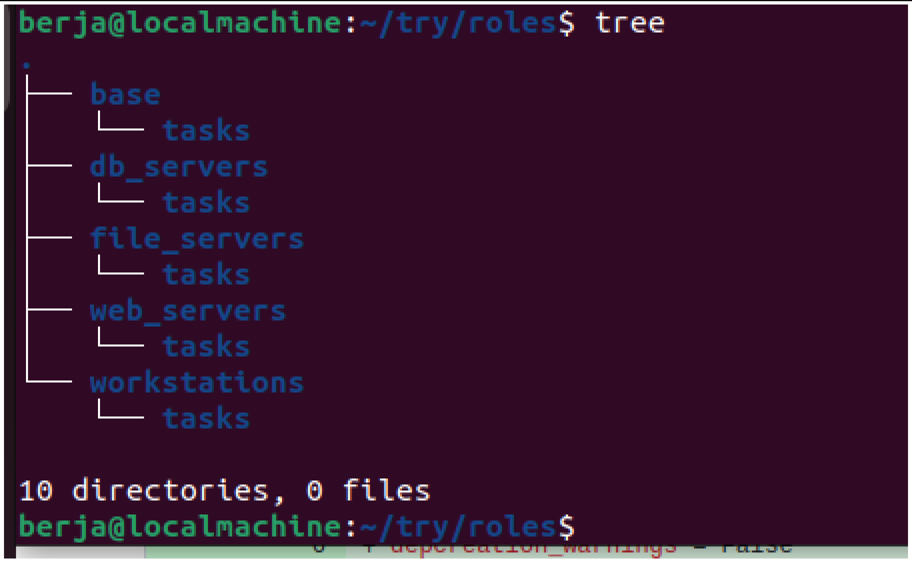
1. Edit the site.yml. Configure roles as follows: (make sure to create a copy of the old site.yml file because you will be copying the specific plays for all groups)


```
---
- 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.



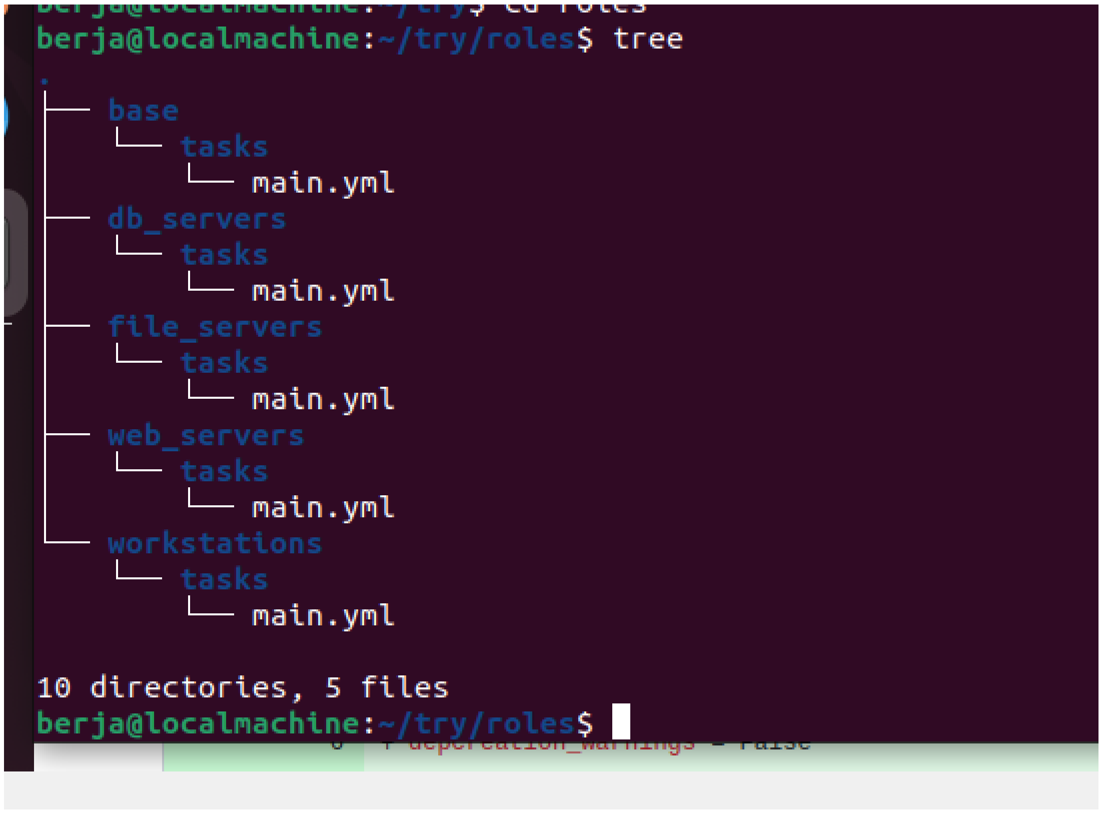
```
    when: ansible_distribution == "Ubuntu"

- 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


- hosts: workstations
  become: true
  tasks:

  - name: install zip
    package:
```

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.



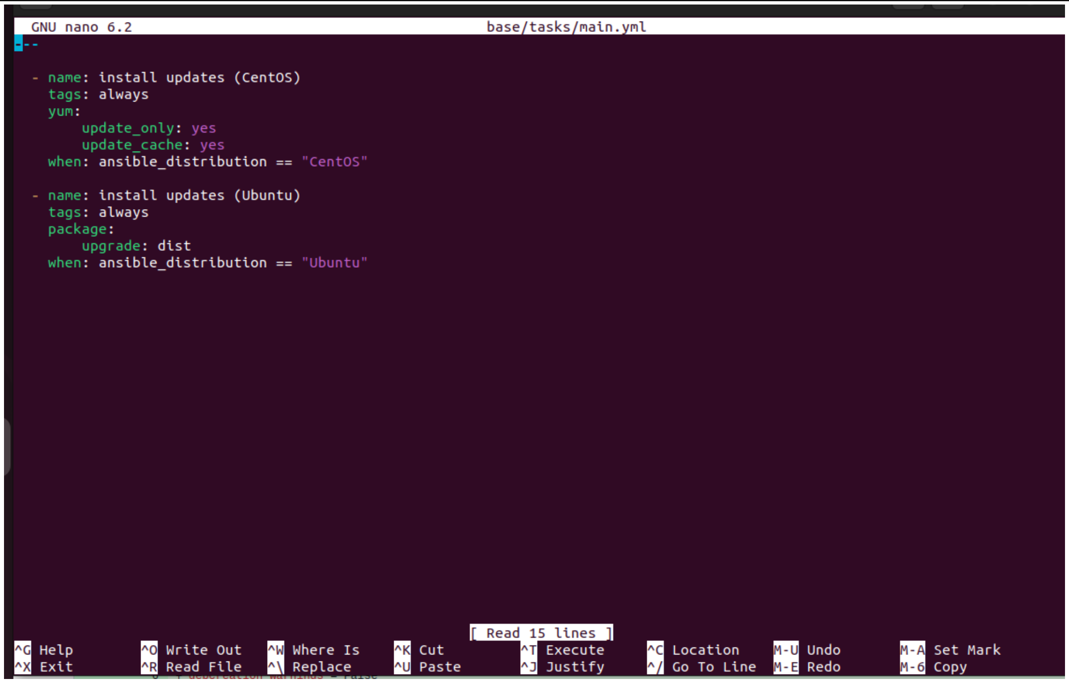
```
berja@localmachine:~/try/roles$ tree
.
├── base
│   └── tasks
├── db_servers
│   └── tasks
├── file_servers
│   └── tasks
├── web_servers
│   └── tasks
└── workstations
    └── tasks

10 directories, 0 files
berja@localmachine:~/try/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.



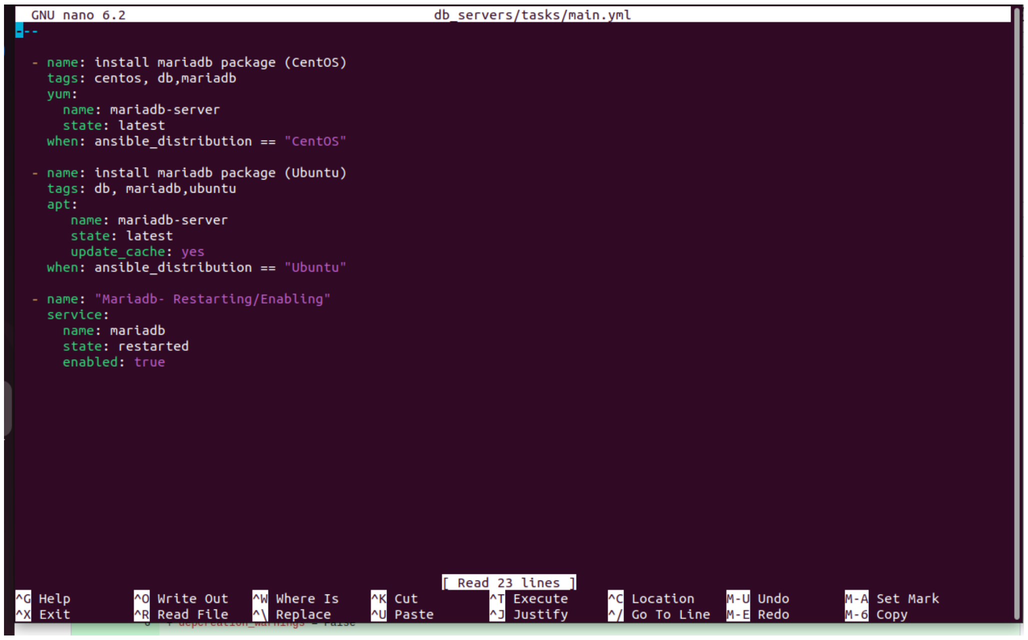
```
berja@localmachine:~/try/roles$ tree
.
├── base
│   └── tasks
│       └── main.yml
├── db_servers
│   └── tasks
│       └── main.yml
├── file_servers
│   └── tasks
│       └── main.yml
├── web_servers
│   └── tasks
│       └── main.yml
└── workstations
    └── tasks
        └── main.yml

10 directories, 5 files
berja@localmachine:~/try/roles$
```

BASE:



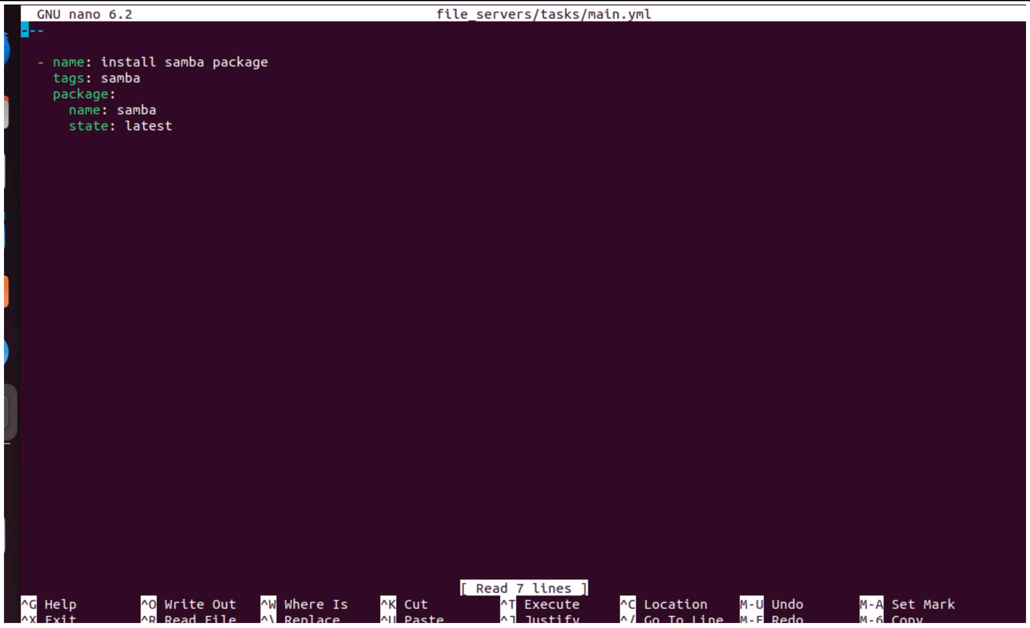
```
GNU nano 6.2                                  base/tasks/main.yml
---

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

- name: install updates (Ubuntu)
  tags: always
  package:
      upgrade: dist
  when: ansible_distribution == "Ubuntu"

                                  [ Read 15 lines ]
^G Help      ^O Write Out  ^W Where Is  ^K Cut    ^T Execute  ^C Location   M-U Undo  M-A Set Mark
^X Exit      ^R Read File  ^\ Replace   ^U Paste  ^J Justify  ^/ Go To Line M-E Redo  M-6 Copy
```

DB_SERVERS:



```
GNU nano 6.2                                  db_servers/tasks/main.yml
---

- name: install mariadb package (CentOS)
  tags: centos, db,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
     update_cache: yes
  when: ansible_distribution == "Ubuntu"

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

                                  [ Read 23 lines ]
^G Help      ^O Write Out  ^W Where Is  ^K Cut    ^T Execute  ^C Location   M-U Undo  M-A Set Mark
^X Exit      ^R Read File  ^\ Replace   ^U Paste  ^J Justify  ^/ Go To Line M-E Redo  M-6 Copy
```

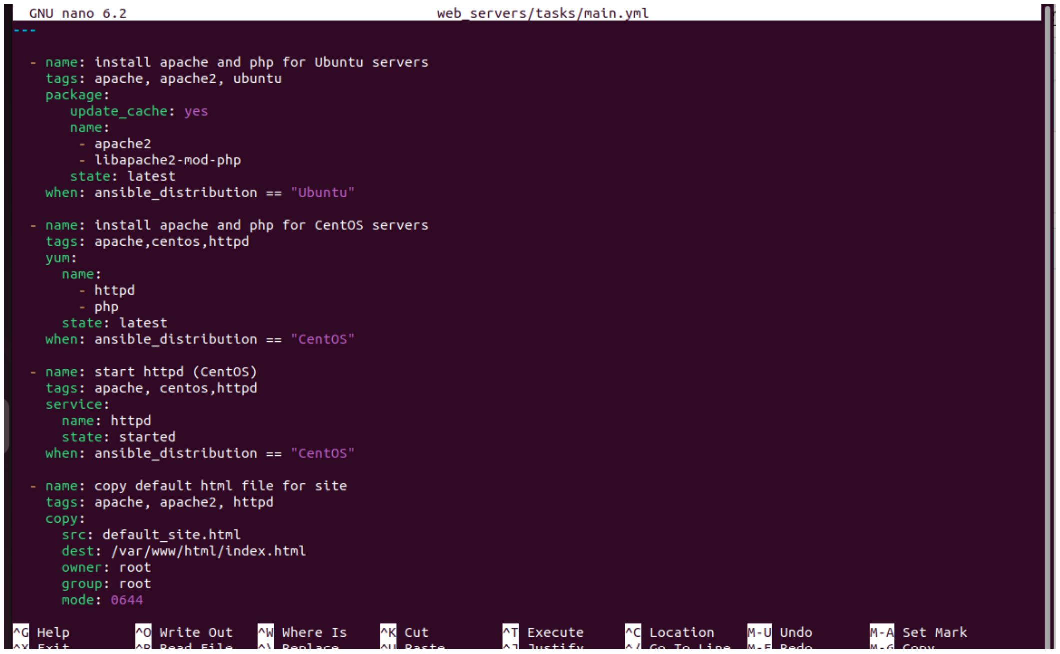
FILE_SERVERS:



```
  GNU nano 6.2                                    file_servers/tasks/main.yml
---

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

```
                                             [ Read 7 lines ]
^G Help        ^O Write Out   ^W Where Is    ^K Cut         ^T Execute     ^C Location    M-U Undo       M-A Set Mark
```

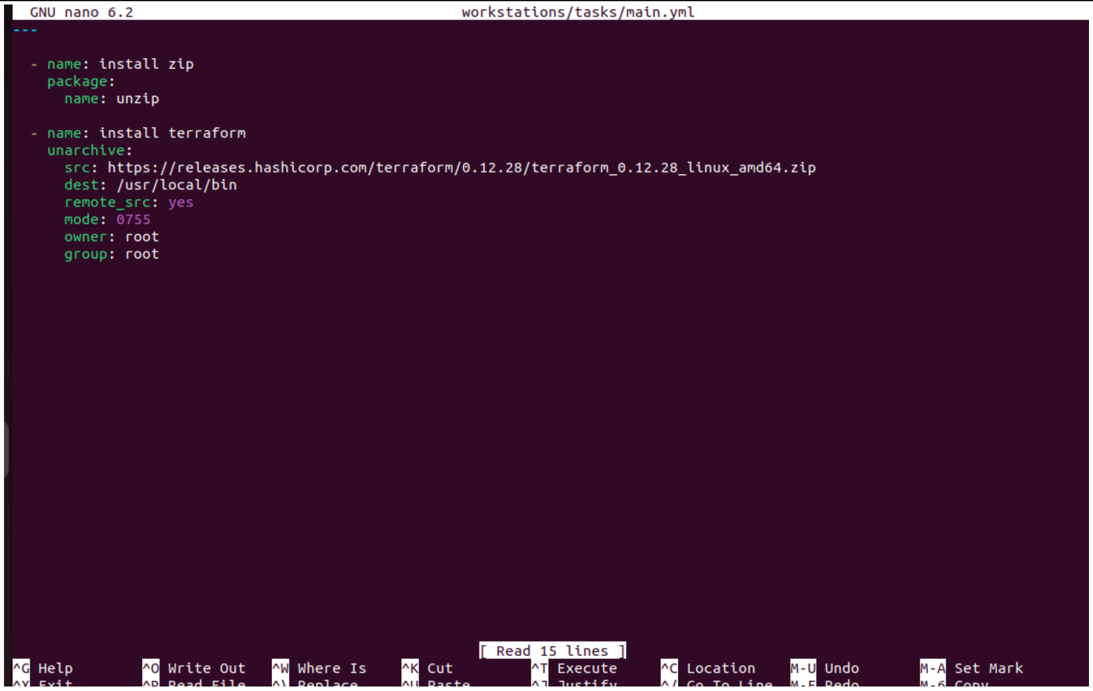
WEB_SERVERS:



```
  GNU nano 6.2                                    web_servers/tasks/main.yml
---

  - name: install apache and php for Ubuntu servers
    tags: apache, apache2, ubuntu
    package:
       update_cache: yes
       name:
         - apache2
         - libapache2-mod-php
       state: latest
    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"

  - name: start httpd (CentOS)
    tags: apache, centos,httpd
    service:
      name: httpd
      state: started
    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
```

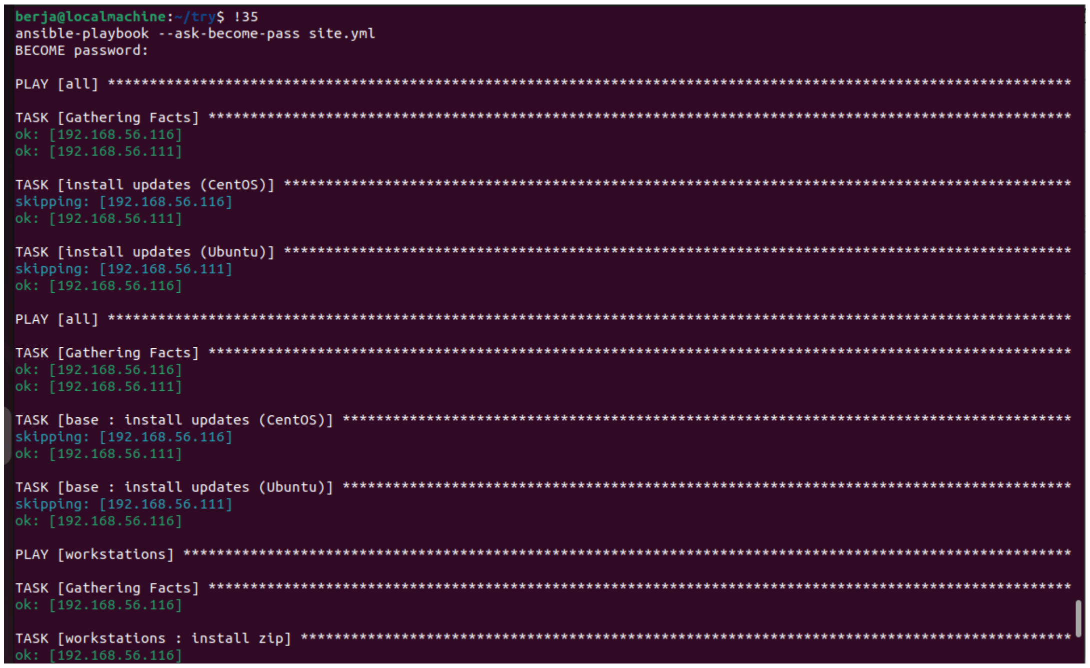
```
^G Help        ^O Write Out   ^W Where Is    ^K Cut         ^T Execute     ^C Location    M-U Undo       M-A Set Mark
```

WORKSTATIONS:



```
  GNU nano 6.2                                  workstations/tasks/main.yml
---

  - name: install zip
    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

                                                       [ Read 15 lines ]
^G Help          ^O Write Out    ^W Where Is     ^K Cut          ^T Execute      ^C Location     M-U Undo        M-A Set Mark
```

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



```
berja@localmachine:~/try$ !35
ansible-playbook --ask-become-pass site.yml
BECOME password:

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

TASK [Gathering Facts] ***************************************************************************************
ok: [192.168.56.116]
ok: [192.168.56.111]

TASK [install updates (CentOS)] ******************************************************************************
skipping: [192.168.56.116]
ok: [192.168.56.111]

TASK [install updates (Ubuntu)] ******************************************************************************
skipping: [192.168.56.111]
ok: [192.168.56.116]

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

TASK [Gathering Facts] ***************************************************************************************
ok: [192.168.56.116]
ok: [192.168.56.111]

TASK [base : install updates (CentOS)] ***********************************************************************
skipping: [192.168.56.116]
ok: [192.168.56.111]

TASK [base : install updates (Ubuntu)] ***********************************************************************
skipping: [192.168.56.111]
ok: [192.168.56.116]

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

TASK [Gathering Facts] ***************************************************************************************
ok: [192.168.56.116]

TASK [workstations : install zip] ****************************************************************************
ok: [192.168.56.116]
```

- **After putting all the code in the yml for each specific directories the task was all the same as before when installing some skipped, ok, and changed inside the ansible this help by showing no error inside it.**

Reflections:

Answer the following:

1. What is the importance of creating roles?

   Creating tasks for specific servers without the use of roles may be feasible, but it's not the most efficient approach for resource management. This is mainly because not all scripts are neatly organized within a single playbook; rather, they are spread across different directories associated with different roles. This underscores the importance of roles in the context of server management. Moreover, an essential aspect of system administration is the fundamental management of corporate servers, which plays a vital role in ensuring a secure and orderly computing environment.

2. What is the importance of managing files?

   Managing files across different servers is crucial for the overall health, security, and efficiency of enterprise server management. It enhances data security, operational efficiency, data integrity, and the capacity to effectively address a variety of operational and security challenges. Effective file management, when consistently practiced, forms a solid foundation for administering an enterprise environment.