Name: Ian Carlo T. Bello	**Date Performed:** October 15, 2022
Course/Section: CPE31S2	**Date Submitted:** October 15, 2022
Instructor: Dr. Jonathan V. Taylar	**Semester and SY:** 1st sem – 3rd year

Activity 7: Managing Files and Creating Roles in Ansible

1. **Objectives:**

1.1 Manage files in remote servers

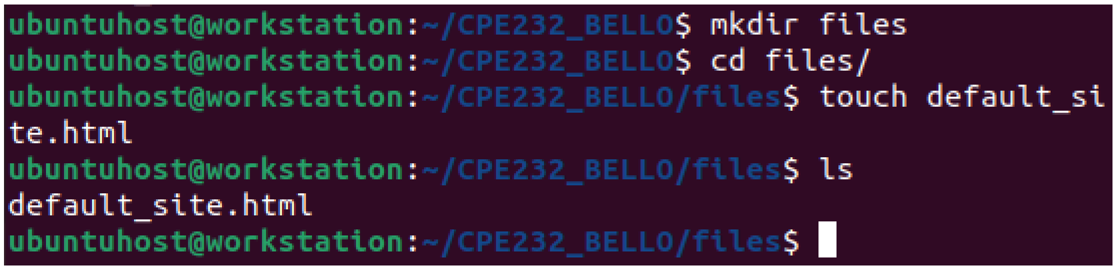
1.2 Implement roles in ansible

2. **Discussion**:

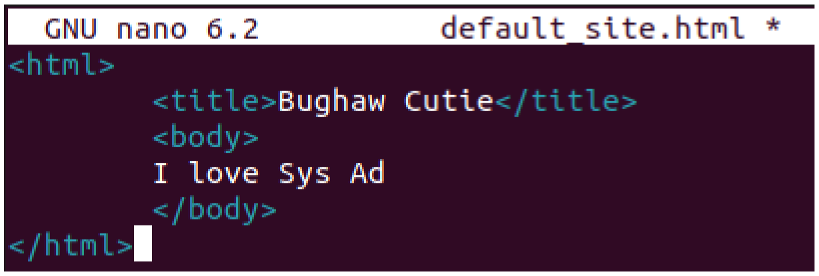
In this activity, we look at the concept of copying a file to a server. We are going to create a file into our git repository and use Ansible to grab that file and put it into a particular place so that we could do things like customize a default website, or maybe install a default configuration file.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



```
ubuntuhost@workstation:~/CPE232_BELLO$ mkdir files
ubuntuhost@workstation:~/CPE232_BELLO$ cd files/
ubuntuhost@workstation:~/CPE232_BELLO/files$ touch default_si
te.html
ubuntuhost@workstation:~/CPE232_BELLO/files$ ls
default_site.html
ubuntuhost@workstation:~/CPE232_BELLO/files$ 
```

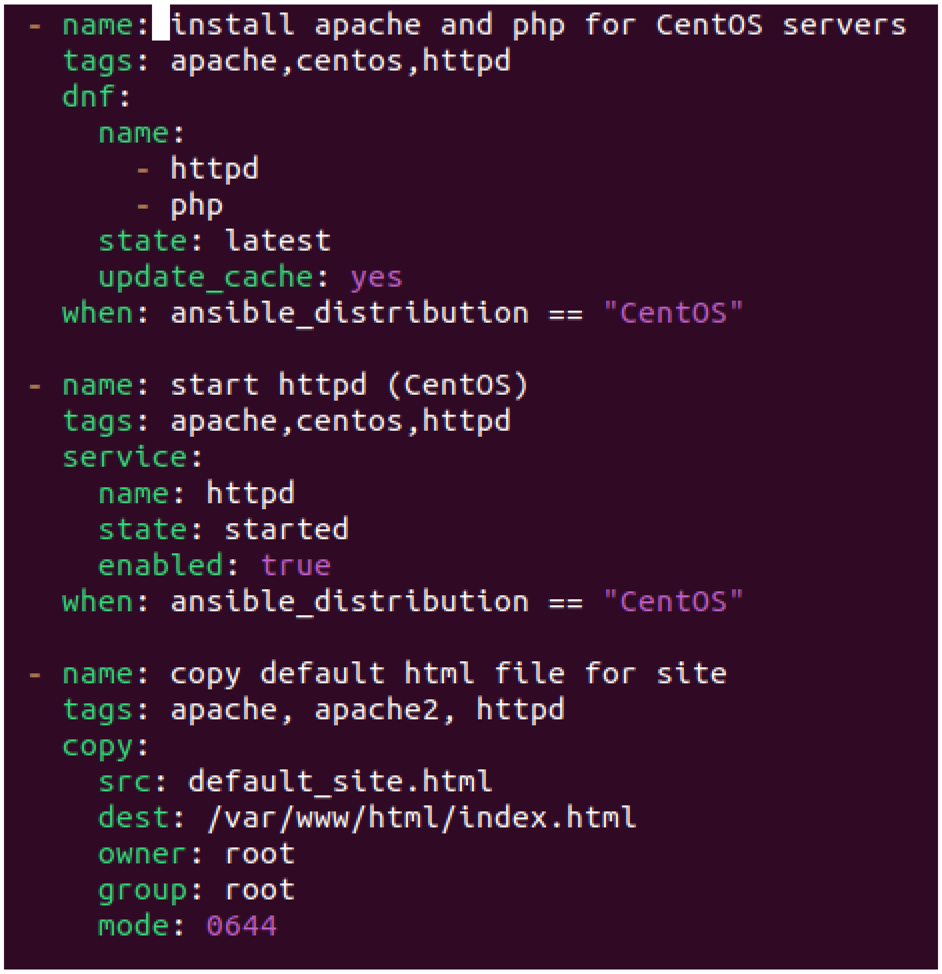


```
  GNU nano 6.2            default_site.html *
<html>
        <title>Bughaw Cutie</title>
        <body>
        I love Sys Ad
        </body>
</html>
```

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:

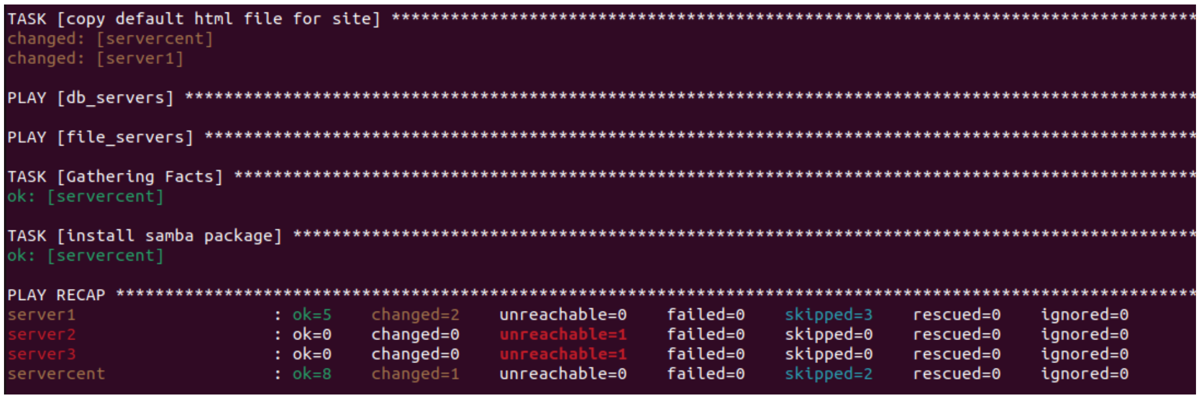
```
- name: copy default html file for site
  tags: apache, apache2, httpd
  copy:
    src: default_site.html
    dest: /var/www/html/index.html
    owner: root
```

group: root
mode: 0644



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

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

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

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

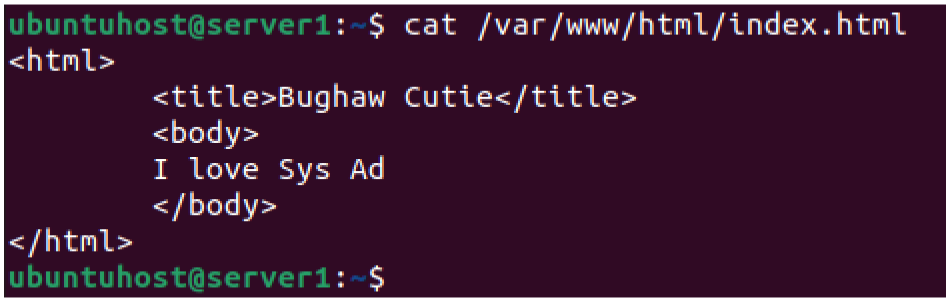


```
TASK [copy default html file for site] *****************************************************************************
changed: [servercent]
changed: [server1]

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

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

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

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

PLAY RECAP *********************************************************************************************************
server1                    : ok=5    changed=2    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0
server2                    : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0
server3                    : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0
servercent                 : ok=8    changed=1    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
```

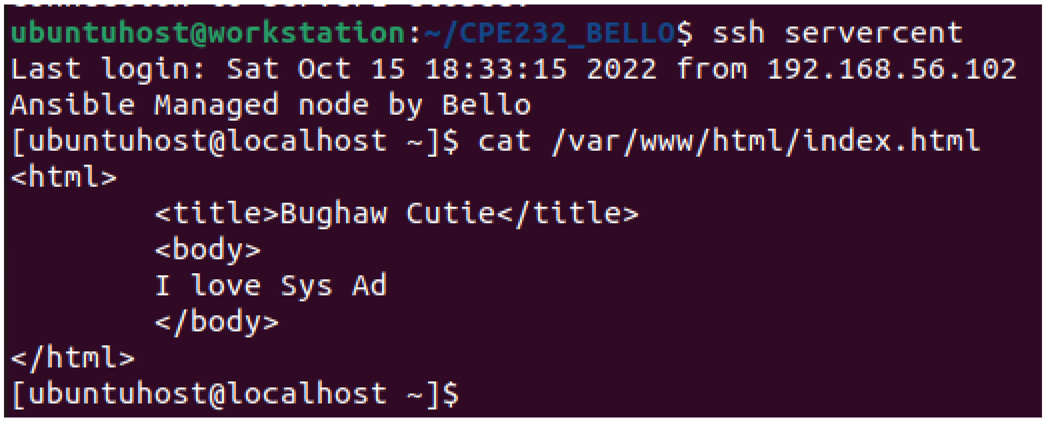
I haven't opened the server 3 to save ram capacity, but task was complete and changed the default html to what we created.

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

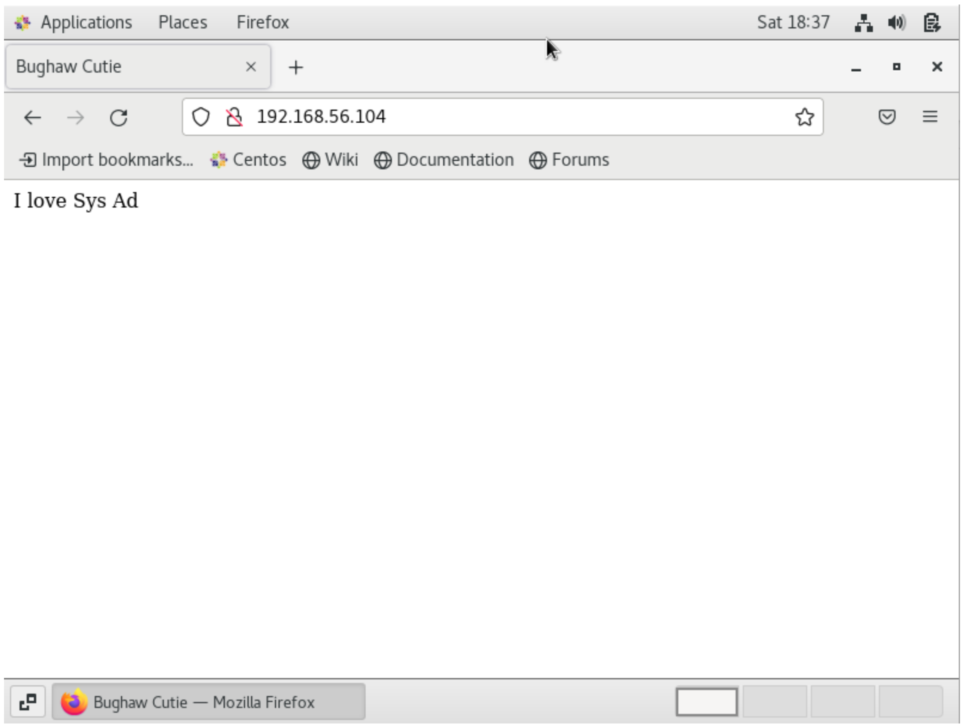
(*default_site.html*). Do both for Ubuntu and CentOS servers. On the CentOS server, go to the browser and type its IP address. Describe the output.



```
ubuntuhost@server1:~$ cat /var/www/html/index.html
<html>
        <title>Bughaw Cutie</title>
        <body>
        I love Sys Ad
        </body>
</html>
ubuntuhost@server1:~$
```



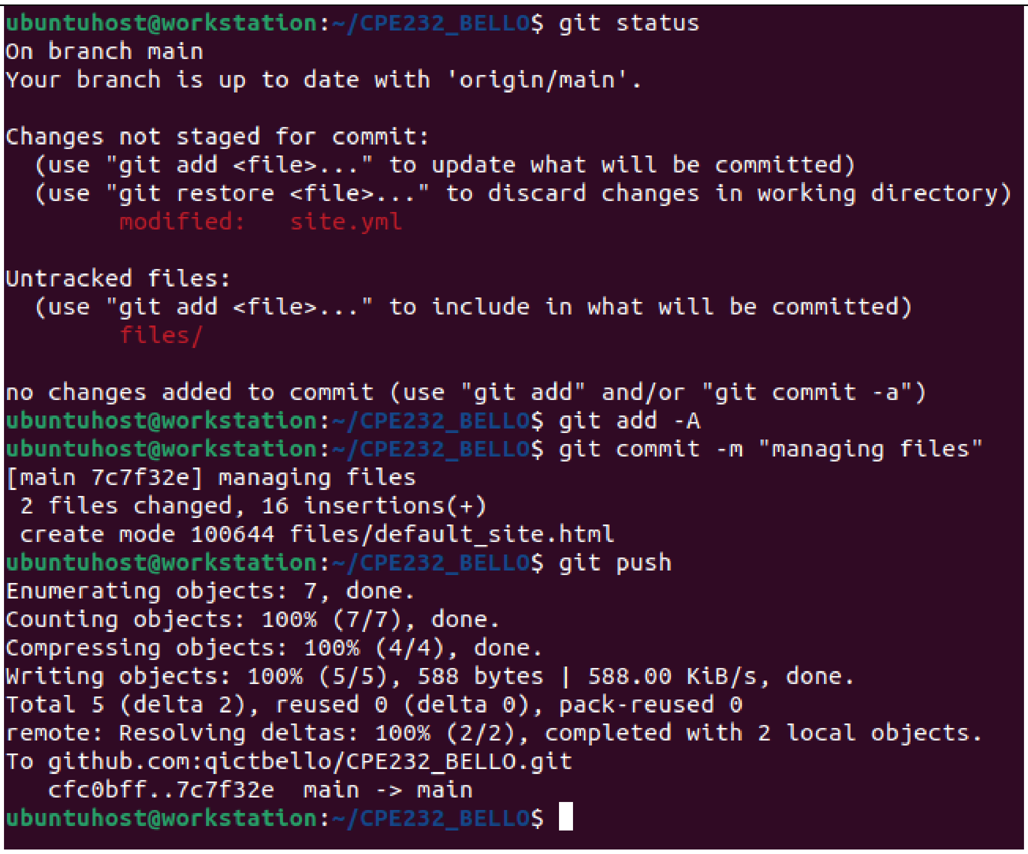
```
ubuntuhost@workstation:~/CPE232_BELLO$ ssh servercent
Last login: Sat Oct 15 18:33:15 2022 from 192.168.56.102
Ansible Managed node by Bello
[ubuntuhost@localhost ~]$ cat /var/www/html/index.html
<html>
        <title>Bughaw Cutie</title>
        <body>
        I love Sys Ad
        </body>
</html>
[ubuntuhost@localhost ~]$
```



I love Sys Ad

We changed the default html in the apache installed and opened it which is the file we suppose to create.

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



```
ubuntuhost@workstation:~/CPE232_BELLO$ git status
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:   site.yml

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

no changes added to commit (use "git add" and/or "git commit -a")
ubuntuhost@workstation:~/CPE232_BELLO$ git add -A
ubuntuhost@workstation:~/CPE232_BELLO$ git commit -m "managing files"
[main 7c7f32e] managing files
 2 files changed, 16 insertions(+)
 create mode 100644 files/default_site.html
ubuntuhost@workstation:~/CPE232_BELLO$ git push
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 588 bytes | 588.00 KiB/s, done.
Total 5 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To github.com:qictbello/CPE232_BELLO.git
   cfc0bff..7c7f32e  main -> main
ubuntuhost@workstation:~/CPE232_BELLO$
```

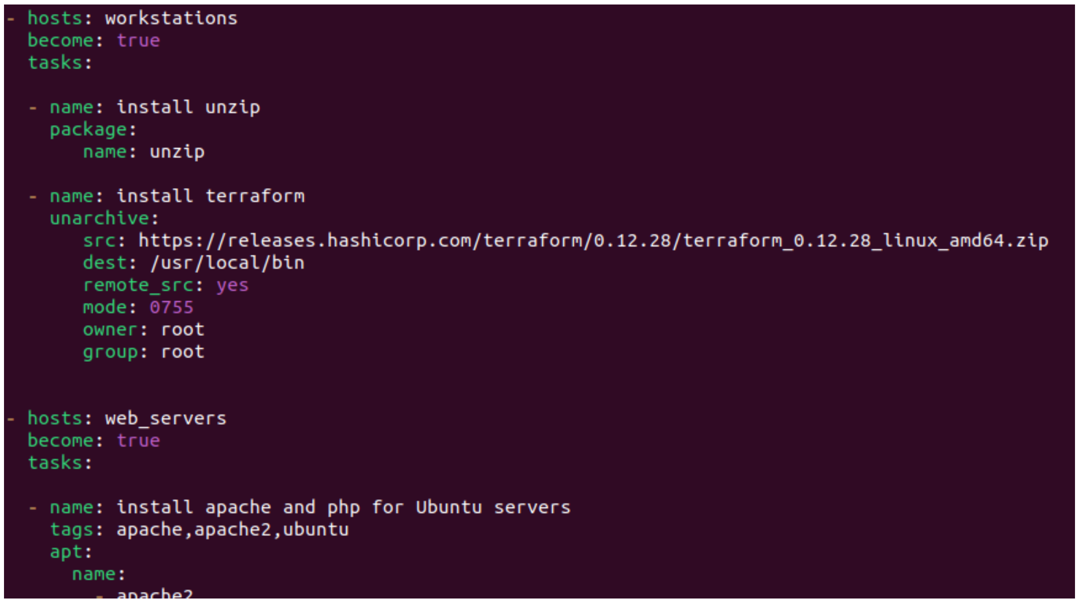
We add and commit the changes into our repository. We pushed it finally to GitHub.

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:

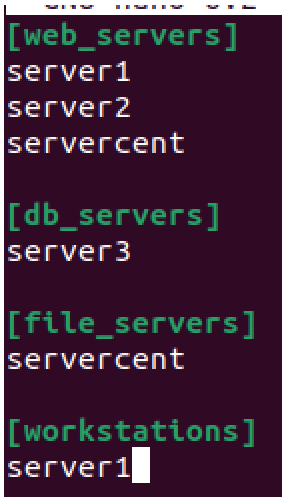
```
- hosts: workstations
  become: true
  tasks:

  - 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



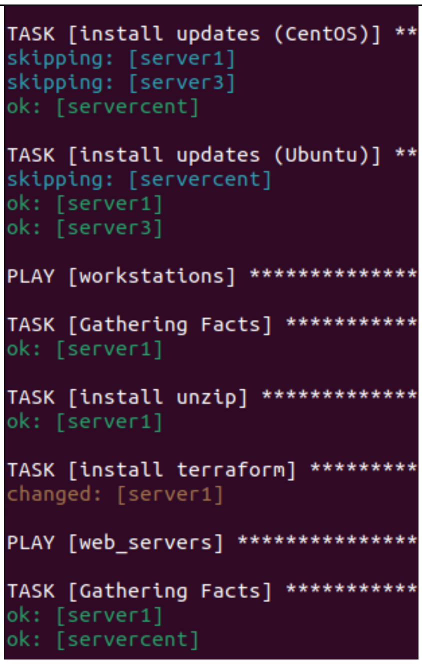
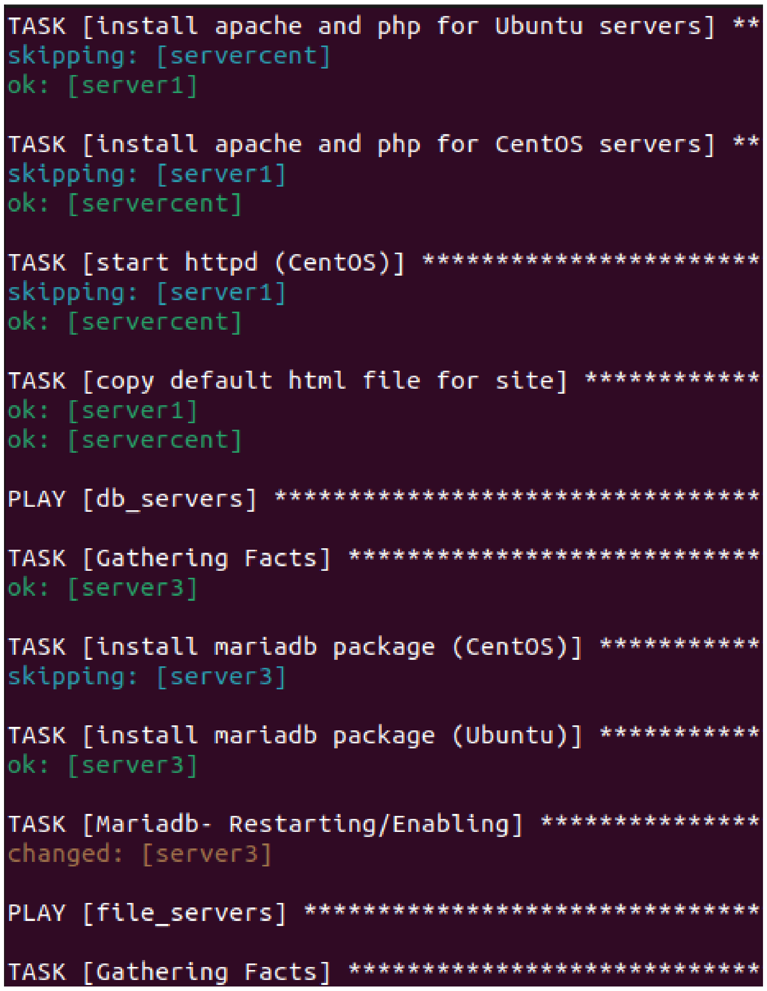
```
- hosts: workstations
  become: true
  tasks:

  - name: install 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: web_servers
  become: true
  tasks:

  - name: install apache and php for Ubuntu servers
    tags: apache,apache2,ubuntu
    apt:
      name:
        - apache2
```

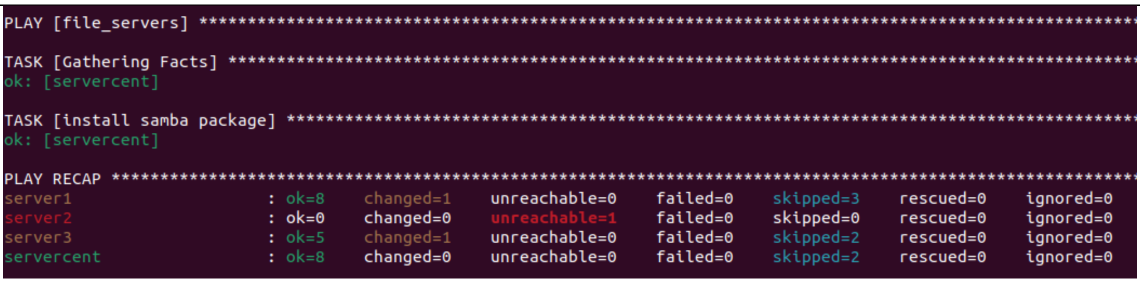
2. Edit the inventory file and add workstations group. Add any Ubuntu remote server. Make sure to remember the IP address.



```
[web_servers]
server1
server2
servercent

[db_servers]
server3

[file_servers]
servercent

[workstations]
server1
```

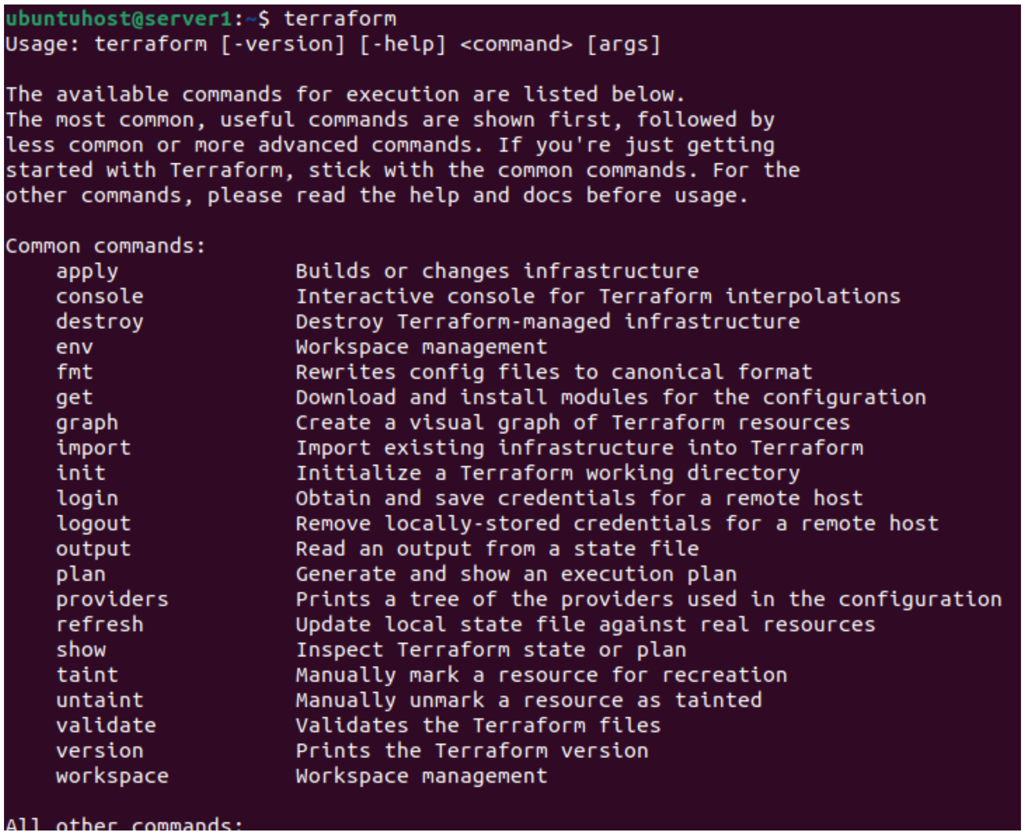
3. Run the playbook. Describe the output.


```
TASK [install updates (CentOS)] **
skipping: [server1]
skipping: [server3]
ok: [servercent]

TASK [install updates (Ubuntu)] **
skipping: [servercent]
ok: [server1]
ok: [server3]

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

TASK [Gathering Facts] ***********
ok: [server1]

TASK [install unzip] *************
ok: [server1]

TASK [install terraform] *********
changed: [server1]

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

TASK [Gathering Facts] ***********
ok: [server1]
ok: [servercent]
```

```
TASK [install apache and php for Ubuntu servers] **
skipping: [servercent]
ok: [server1]

TASK [install apache and php for CentOS servers] **
skipping: [server1]
ok: [servercent]

TASK [start httpd (CentOS)] ************************
skipping: [server1]
ok: [servercent]

TASK [copy default html file for site] ************
ok: [server1]
ok: [servercent]

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

TASK [Gathering Facts] ****************************
ok: [server3]

TASK [install mariadb package (CentOS)] ***********
skipping: [server3]

TASK [install mariadb package (Ubuntu)] ***********
ok: [server3]

TASK [Mariadb- Restarting/Enabling] ***************
changed: [server3]

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

TASK [Gathering Facts] ****************************
```



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

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

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

PLAY RECAP *************************************************************************************
server1                    : ok=8    changed=1    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0
server2                    : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0
server3                    : ok=5    changed=1    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
servercent                 : ok=8    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
```

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

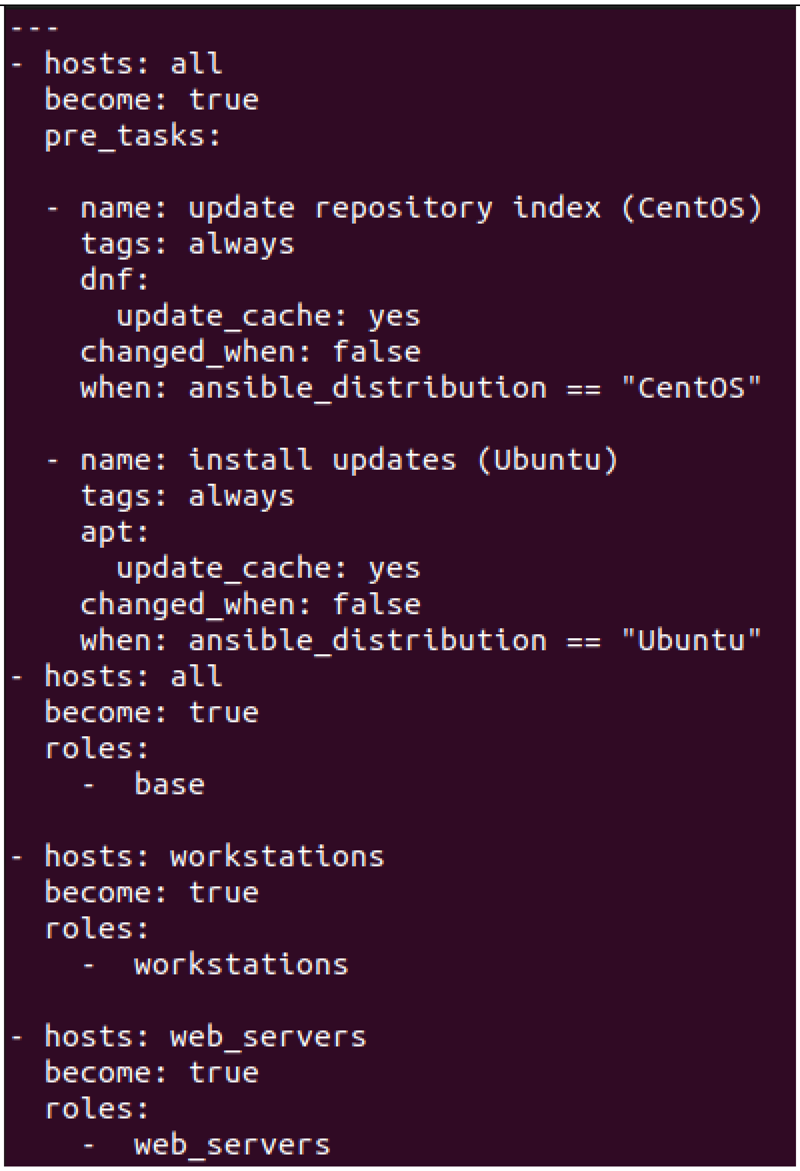


```
ubuntuhost@server1:~$ terraform
Usage: terraform [-version] [-help] <command> [args]

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:
```

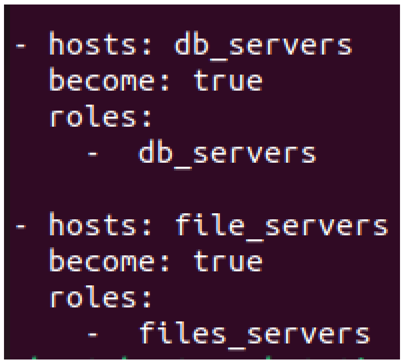
We successfully downloaded Terraform in the workstations which is server 1.

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
```

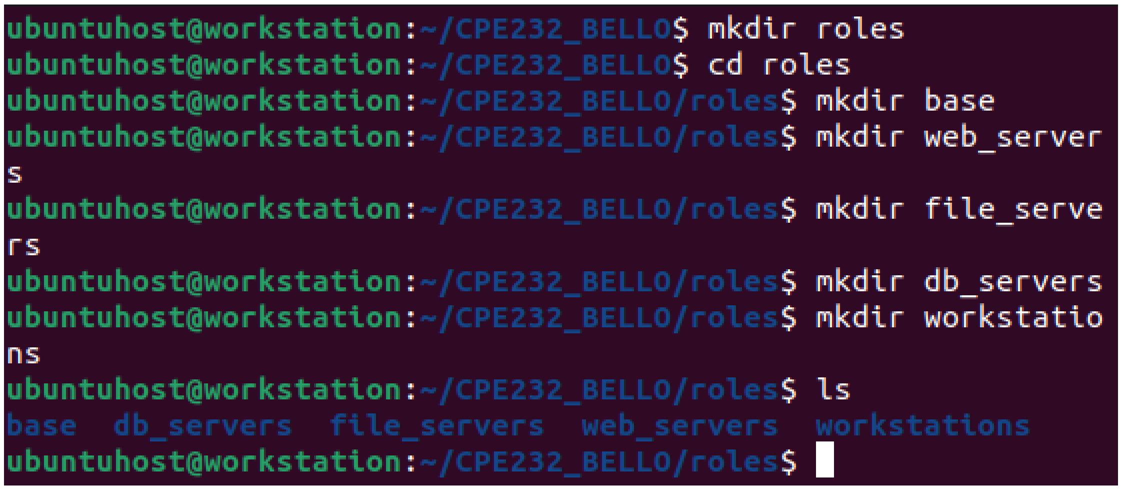


```
- hosts: db_servers
  become: true
  roles:
    -  db_servers

- hosts: file_servers
  become: true
  roles:
    -  files_servers
```

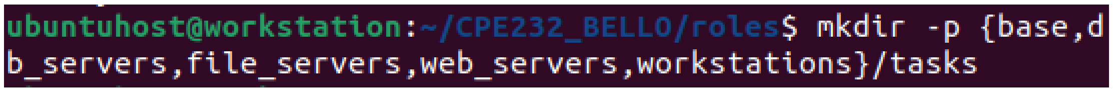
Save the file and exit.

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.

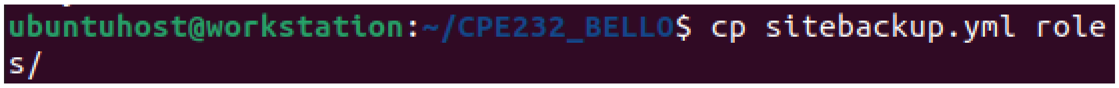


```
ubuntuhost@workstation:~/CPE232_BELLO$ mkdir roles
ubuntuhost@workstation:~/CPE232_BELLO$ cd roles
ubuntuhost@workstation:~/CPE232_BELLO/roles$ mkdir base
ubuntuhost@workstation:~/CPE232_BELLO/roles$ mkdir web_server
s
ubuntuhost@workstation:~/CPE232_BELLO/roles$ mkdir file_serve
rs
ubuntuhost@workstation:~/CPE232_BELLO/roles$ mkdir db_servers
ubuntuhost@workstation:~/CPE232_BELLO/roles$ mkdir workstatio
ns
ubuntuhost@workstation:~/CPE232_BELLO/roles$ ls
base  db_servers  file_servers  web_servers  workstations
ubuntuhost@workstation:~/CPE232_BELLO/roles$ 
```

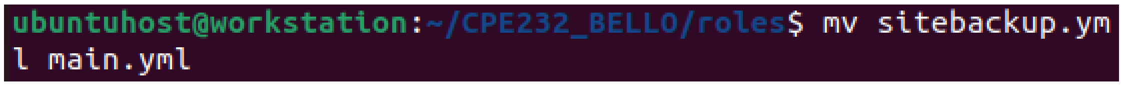


```
ubuntuhost@workstation:~/CPE232_BELLO/roles$ mkdir -p {base,d
b_servers,file_servers,web_servers,workstations}/tasks
```

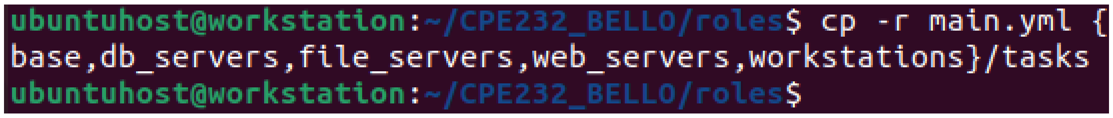
3. Go to tasks for all directory and create a file. Name it main.yml. In each of the tasks for all directories, copy and paste the code from the old site.yml file. Show all contents of main.yml files for all tasks.



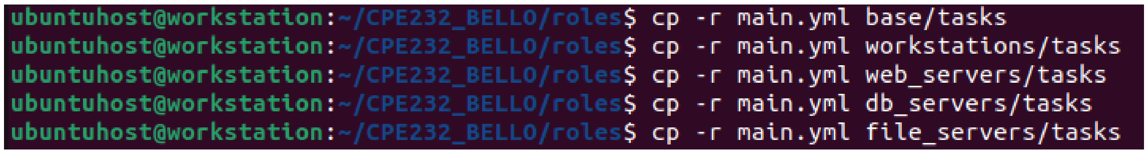
```
ubuntuhost@workstation:~/CPE232_BELLO$ cp sitebackup.yml role
s/
```



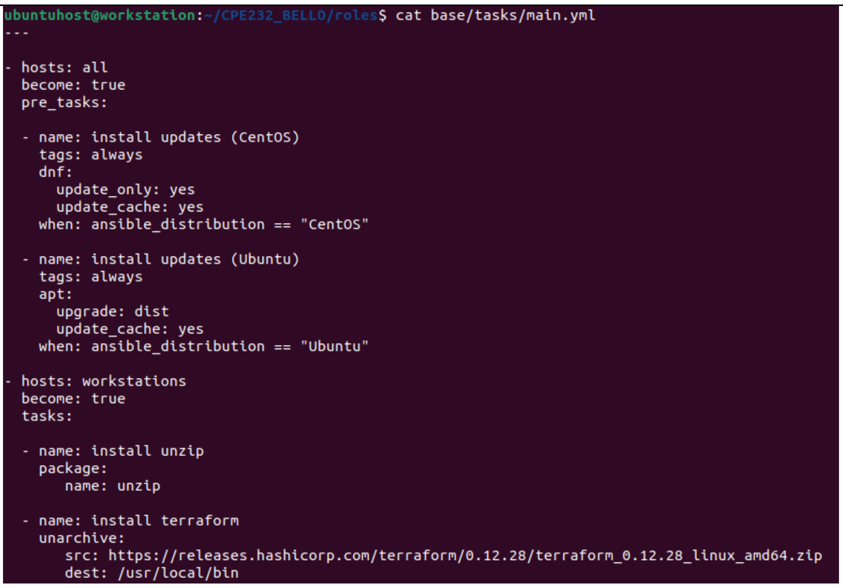
```
ubuntuhost@workstation:~/CPE232_BELLO/roles$ mv sitebackup.ym
l main.yml
```



```
ubuntuhost@workstation:~/CPE232_BELLO/roles$ cp -r main.yml {
base,db_servers,file_servers,web_servers,workstations}/tasks
ubuntuhost@workstation:~/CPE232_BELLO/roles$
```

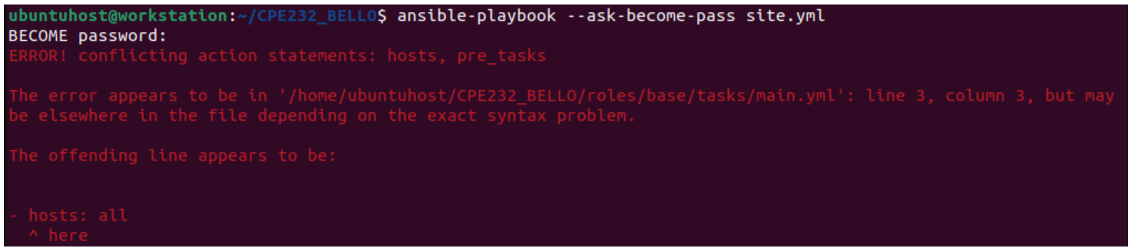


```
ubuntuhost@workstation:~/CPE232_BELLO/roles$ cp -r main.yml base/tasks
ubuntuhost@workstation:~/CPE232_BELLO/roles$ cp -r main.yml workstations/tasks
ubuntuhost@workstation:~/CPE232_BELLO/roles$ cp -r main.yml web_servers/tasks
ubuntuhost@workstation:~/CPE232_BELLO/roles$ cp -r main.yml db_servers/tasks
ubuntuhost@workstation:~/CPE232_BELLO/roles$ cp -r main.yml file_servers/tasks
```



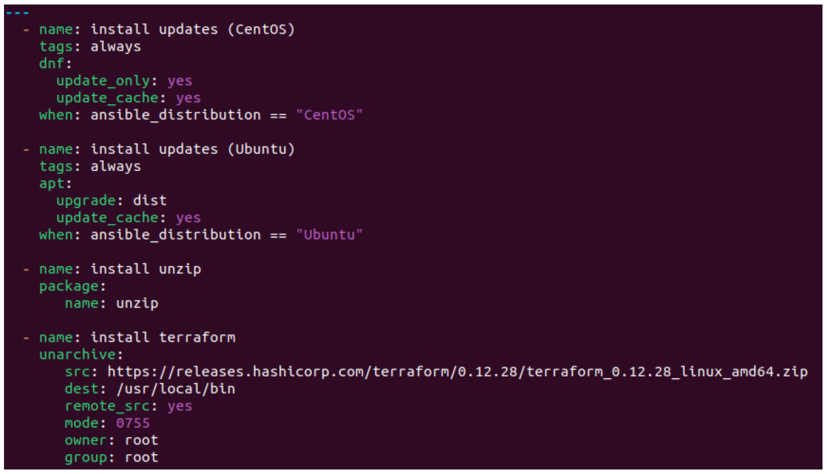
```
ubuntuhost@workstation:~/CPE232_BELLO/roles$ cat base/tasks/main.yml
---

- hosts: all
  become: true
  pre_tasks:

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

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

- hosts: workstations
  become: true
  tasks:

  - name: install unzip
    package:
       name: unzip

  - name: install terraform
    unarchive:
       src: https://releases.hashicorp.com/terraform/0.12.28/terraform_0.12.28_linux_amd64.zip
       dest: /usr/local/bin
```

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

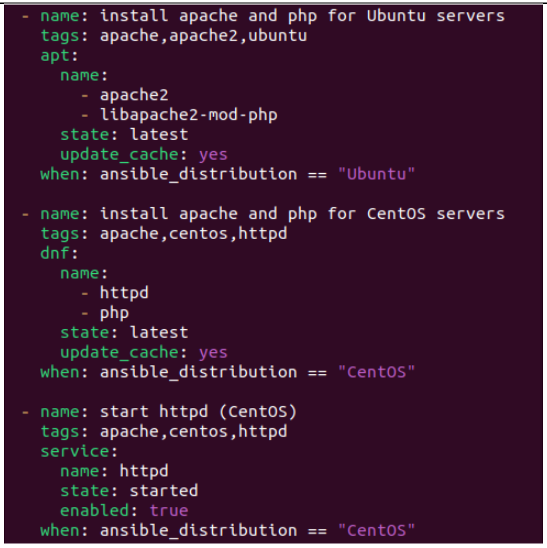


```
ubuntuhost@workstation:~/CPE232_BELLO$ ansible-playbook --ask-become-pass site.yml
BECOME password:
ERROR! conflicting action statements: hosts, pre_tasks

The error appears to be in '/home/ubuntuhost/CPE232_BELLO/roles/base/tasks/main.yml': line 3, column 3, but may
be elsewhere in the file depending on the exact syntax problem.

The offending line appears to be:


- hosts: all
  ^ here
```

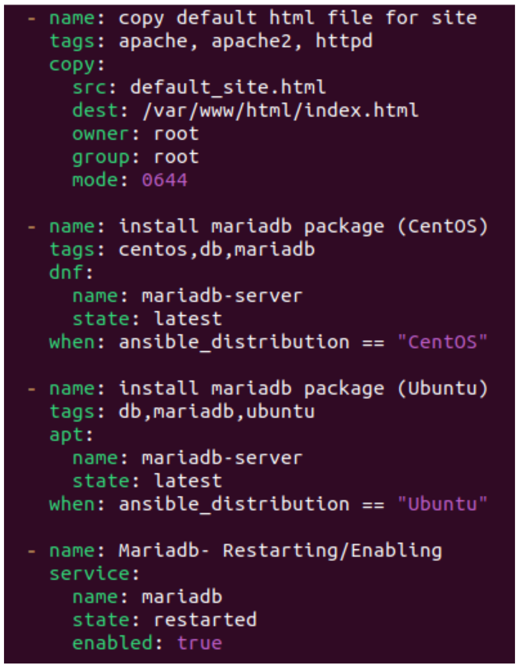
After running the site.yml we get this error. To fix this we need to remove – hosts become and tasks per command/task in each yml.



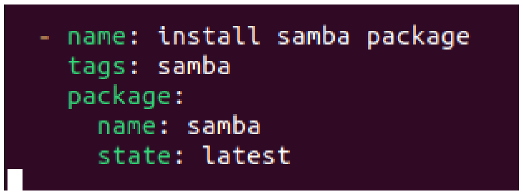
```
---
  - name: install updates (CentOS)
    tags: always
    dnf:
      update_only: yes
      update_cache: yes
    when: ansible_distribution == "CentOS"

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

  - 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
```



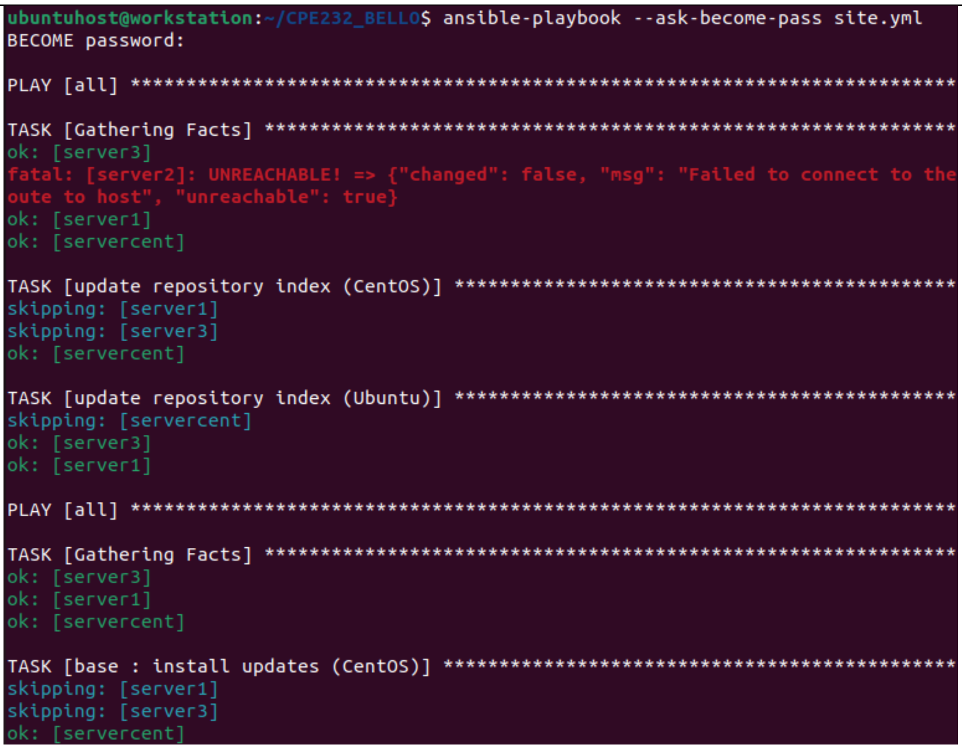
```yaml
- name: install apache and php for Ubuntu servers
  tags: apache,apache2,ubuntu
  apt:
    name:
      - apache2
      - libapache2-mod-php
    state: latest
    update_cache: yes
  when: ansible_distribution == "Ubuntu"

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

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

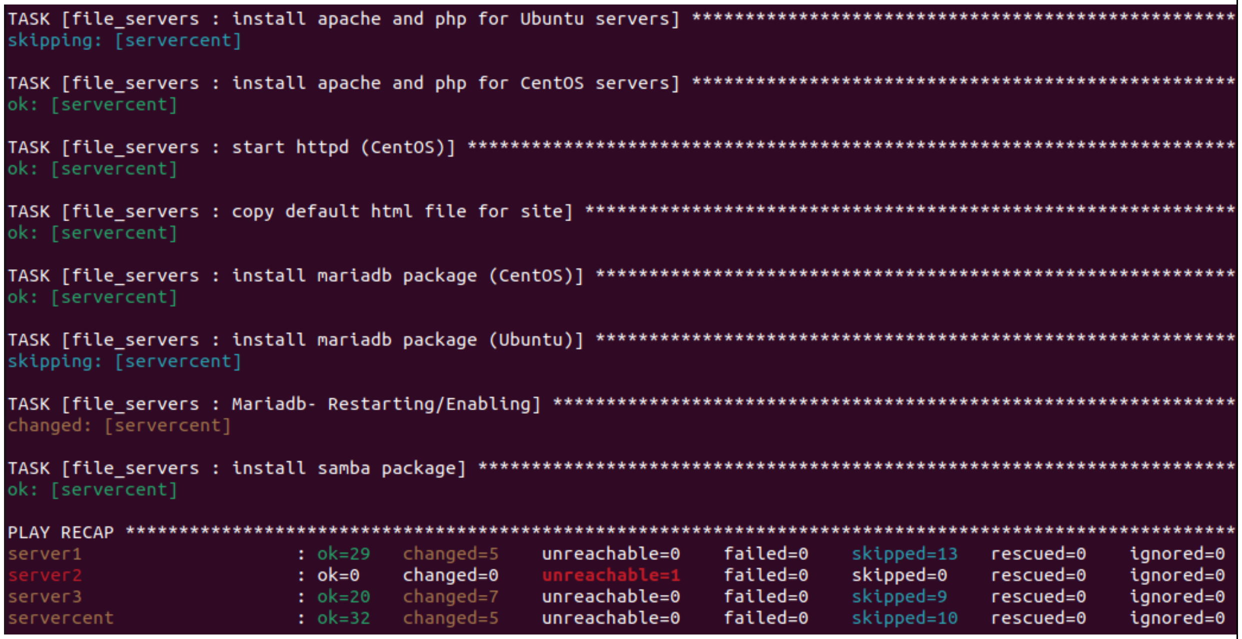


```yaml
- name: copy default html file for site
  tags: apache, apache2, httpd
  copy:
    src: default_site.html
    dest: /var/www/html/index.html
    owner: root
    group: root
    mode: 0644

- name: install mariadb package (CentOS)
  tags: centos,db,mariadb
  dnf:
    name: mariadb-server
    state: latest
  when: ansible_distribution == "CentOS"

- 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
```



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


```
ubuntuhost@workstation:~/CPE232_BELLO$ ansible-playbook --ask-become-pass site.yml
BECOME password:

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

TASK [Gathering Facts] *************************************************************
ok: [server3]
fatal: [server2]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the
oute to host", "unreachable": true}
ok: [server1]
ok: [servercent]

TASK [update repository index (CentOS)] ********************************************
skipping: [server1]
skipping: [server3]
ok: [servercent]

TASK [update repository index (Ubuntu)] ********************************************
skipping: [servercent]
ok: [server3]
ok: [server1]

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

TASK [Gathering Facts] *************************************************************
ok: [server3]
ok: [server1]
ok: [servercent]

TASK [base : install updates (CentOS)] *********************************************
skipping: [server1]
skipping: [server3]
ok: [servercent]
```


```
TASK [file_servers : install apache and php for Ubuntu servers] ***********************************************************
skipping: [servercent]

TASK [file_servers : install apache and php for CentOS servers] ***********************************************************
ok: [servercent]

TASK [file_servers : start httpd (CentOS)] ********************************************************************************
ok: [servercent]

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

TASK [file_servers : install mariadb package (CentOS)] ********************************************************************
ok: [servercent]

TASK [file_servers : install mariadb package (Ubuntu)] ********************************************************************
skipping: [servercent]

TASK [file_servers : Mariadb- Restarting/Enabling] ************************************************************************
changed: [servercent]

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

PLAY RECAP ****************************************************************************************************************
server1                    : ok=29   changed=5    unreachable=0    failed=0    skipped=13   rescued=0    ignored=0
server2                    : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0
server3                    : ok=20   changed=7    unreachable=0    failed=0    skipped=9    rescued=0    ignored=0
servercent                 : ok=32   changed=5    unreachable=0    failed=0    skipped=10   rescued=0    ignored=0
```

We removed the – hosts become and tasks in each line. This will run every command on all roles. We can also specify which commands only runs for which hosts by only placing specified command in each task main.yml. After we copied the old site.yml in each roles task the whole command will run in each host. We need to splice each command for each host.

Reflections:

Answer the following:

1. What is the importance of creating roles?

It is important to create isolation between different roles so there will be no confusion in which go where and how commands run. Isolation will help each host debugging of codes that will run into them. Creating a role is easier since set of commands are separate from the main file that we need to run.

2. What is the importance of managing files?

We always do manipulate file in such huge organization. Datacenters and servers do share file and we do need to copy faster in each server with only one command that's why we run ansible. Control node do have the authority to update, revise and fix files and sometimes replace and delete them.