Name: Jefferson Langbid	Date Performed: Oct 13, 2022
Course/Section: CPE232-CPE31S23	Date Submitted:
Instructor: Dr. Taylar	Semester and SY: 2022-2023

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.

```
jefferson@LocalMachine:~$ cd Langbid_PrelimExam
jefferson@LocalMachine:~/Langbid_PrelimExam$ mkdir files
jefferson@LocalMachine:~/Langbid_PrelimExam$ cd files
jefferson@LocalMachine:~/Langbid_PrelimExam/files$ sudo nano default_site.html
[sudo] password for jefferson:
```

```
jefferson@LocalMachine: ~/Langbid_PrelimExam/files
  GNU nano 6.2                    default_site.html *
<html>
<body>

<h1> Ansible </h1>

</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: 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.

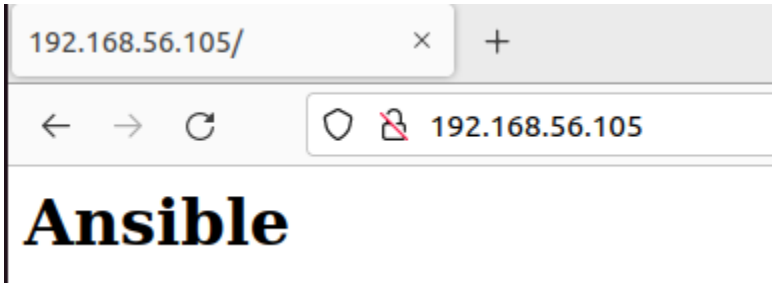
```
jefferson@LocalMachine: ~/Langbid_PrelimExam

TASK [copy default html file for site] ***************************************
changed: [192.168.56.105]
changed: [192.168.56.106]

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

TASK [Gathering Facts] *******************************************************
ok: [192.168.56.105]

TASK [install mariadb package (CentOS)] **************************************
skipping: [192.168.56.105]

TASK [Mariadb- Restarting/Enabling] ******************************************
changed: [192.168.56.105]

TASK [Mariadb- Restarting/Enabling] ******************************************
changed: [192.168.56.105]

TASK [install mariadb package (Ubuntu)] **************************************
ok: [192.168.56.105]

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

TASK [Gathering Facts] *******************************************************
ok: [192.168.56.106]

TASK [install samba package] *************************************************
ok: [192.168.56.106]

PLAY RECAP *******************************************************************
192.168.56.105             : ok=9    changed=4    unreachable=0    failed=0    skipped=4    rescued=0
    ignored=0
192.168.56.106             : ok=8    changed=2    unreachable=0    failed=0    skipped=2    rescued=0
    ignored=0
```

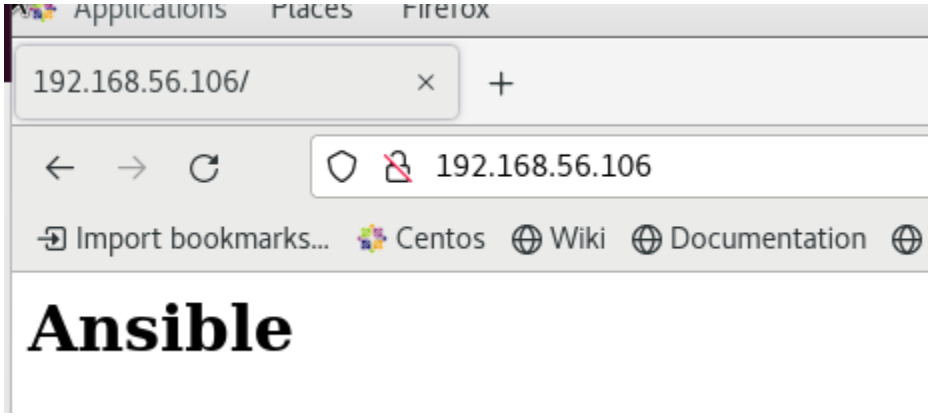
```
TASK [copy default html file for site] ***************************************
changed: [192.168.56.105]
changed: [192.168.56.106]
```

It run successfully and it changed the html file for the site in the 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.

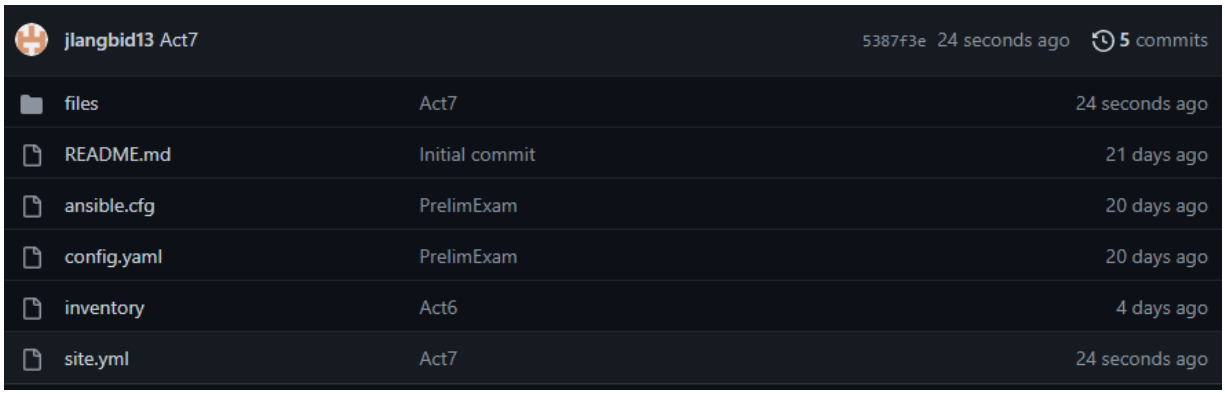
```
jefferson@LocalMachine:~/Langbid_PrelimExam/files$ cat default_site.html
<html>
<body>

<h1> Ansible </h1>

</body>
</html>
```





It displayed the html output that is inputted in the ansible file directory.

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



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
```

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
[web_servers]
192.168.56.105
192.168.56.106

[db_servers]
192.168.56.105

[file_servers]
192.168.56.106

[workstations]
192.168.56.105
```

3. Run the playbook. Describe the output.

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

TASK [Gathering Facts] *********************************************************************************
ok: [192.168.56.105]

TASK [install unzip] ***********************************************************************************
ok: [192.168.56.105]

TASK [install terraform] *******************************************************************************
changed: [192.168.56.105]

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

TASK [Gathering Facts] *********************************************************************************
ok: [192.168.56.106]
ok: [192.168.56.105]
```

The playbook ran successfully and it installed terraform in the specific ip address that is imputed.

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

```
jefferson@Server1:~$ terraform --version
Terraform v0.12.28

Your version of Terraform is out of date! The latest version
is 1.3.2. You can update by downloading from https://www.terraform.io/downloads
.html
```

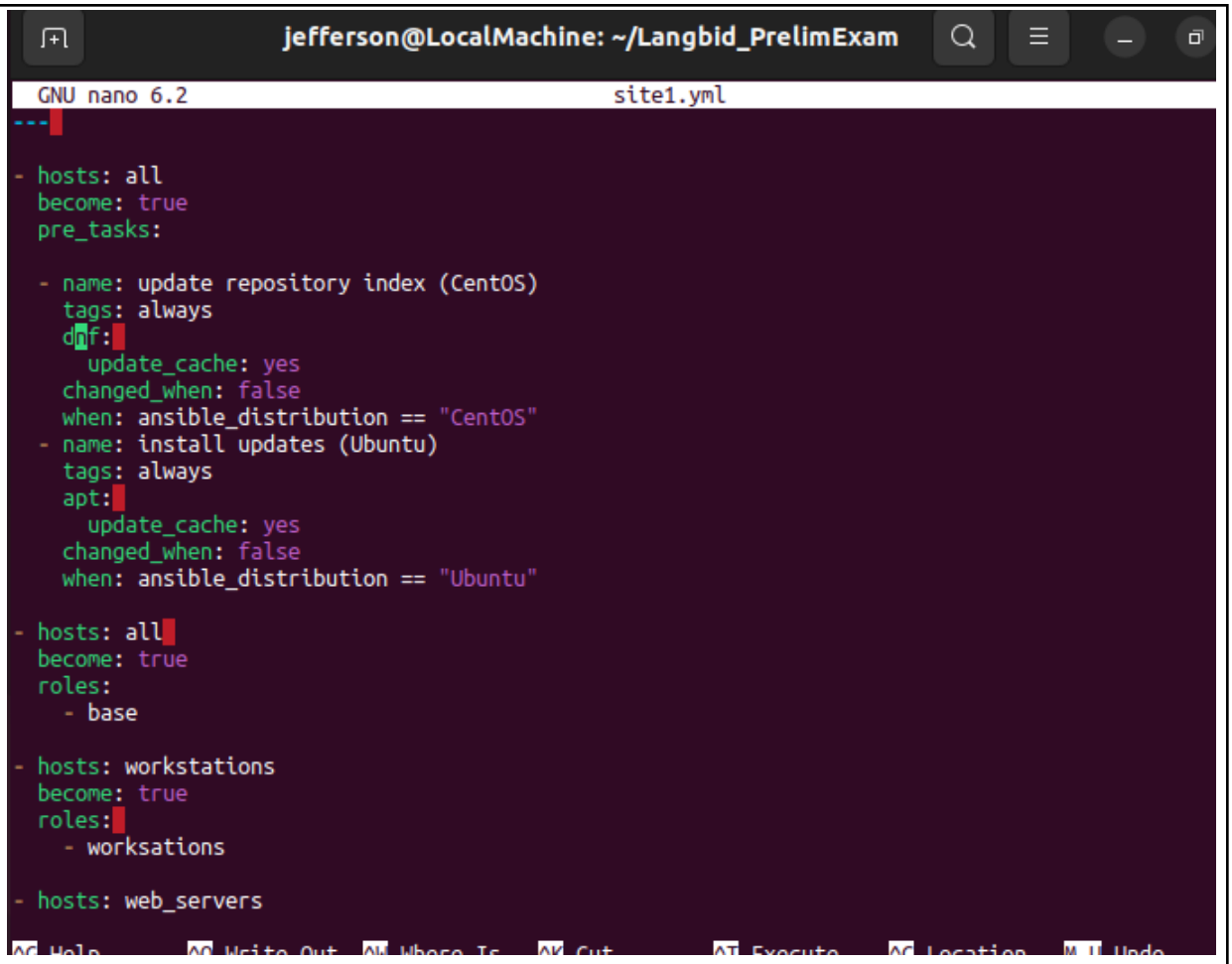
it is now installed to the specific ubuntu workstation ip address.

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.



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.

```
jefferson@LocalMachine-VirtualBox:~/Langbid_PrelimExam/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
```

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.

```
jefferson@LocalMachine:~/Langbid_PrelimExam/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
jefferson@LocalMachine:~/Langbid_PrelimExam/roles$
```

Base:

jefferson@LocalMachine: ~/Langbid_PrelimExam/roles/bas...

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

 - 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
```

jefferson@LocalMachine: ~/Langbid_PrelimExam/roles/bas...

GNU nano 6.2 main.yml

```
     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


 - 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
```

db_servers:

```
  GNU nano 6.2                                        main.yml
---

  - 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,mariadbm, ubuntu
    apt:
      name: mariadb-server
      state: latest
    when: ansible_distribution == "Ubuntu"

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

web_servers:

```
jefferson@LocalMachine: ~/Langbid_PrelimExam/roles/web...
 GNU nano 6.2                                        main.yml
---

- 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
  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
```

file_servers:

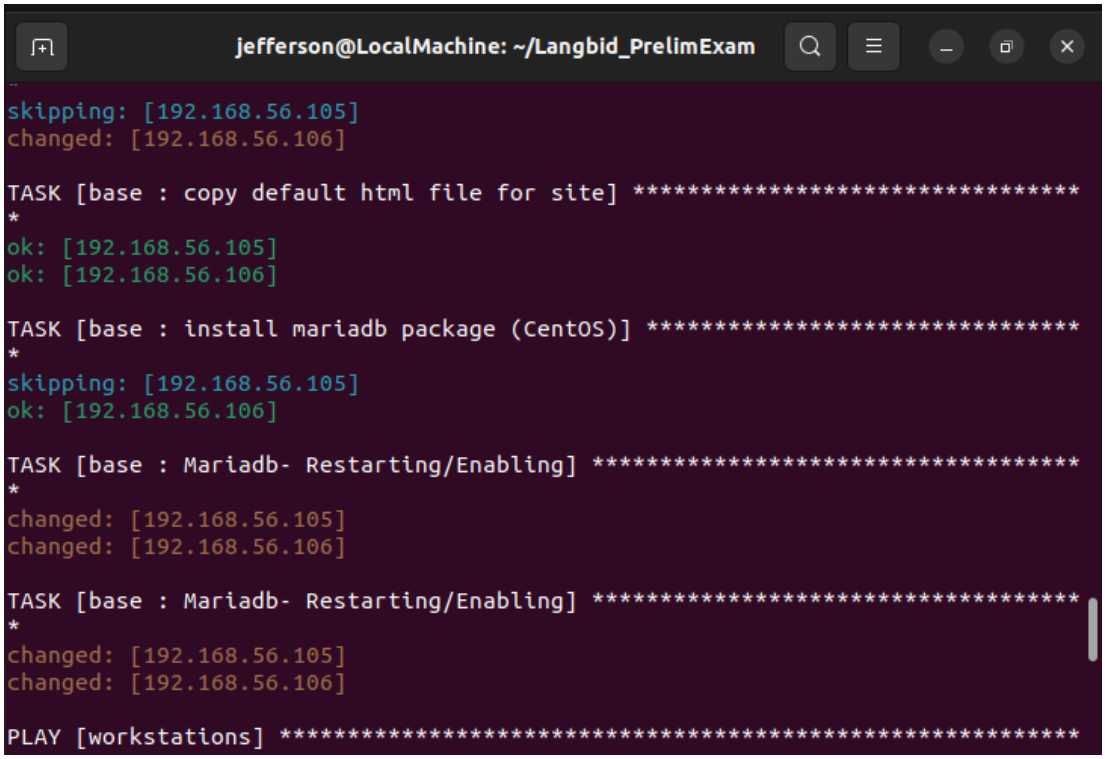
```
  GNU nano 6.2                                          main.yml
---

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

workstations:

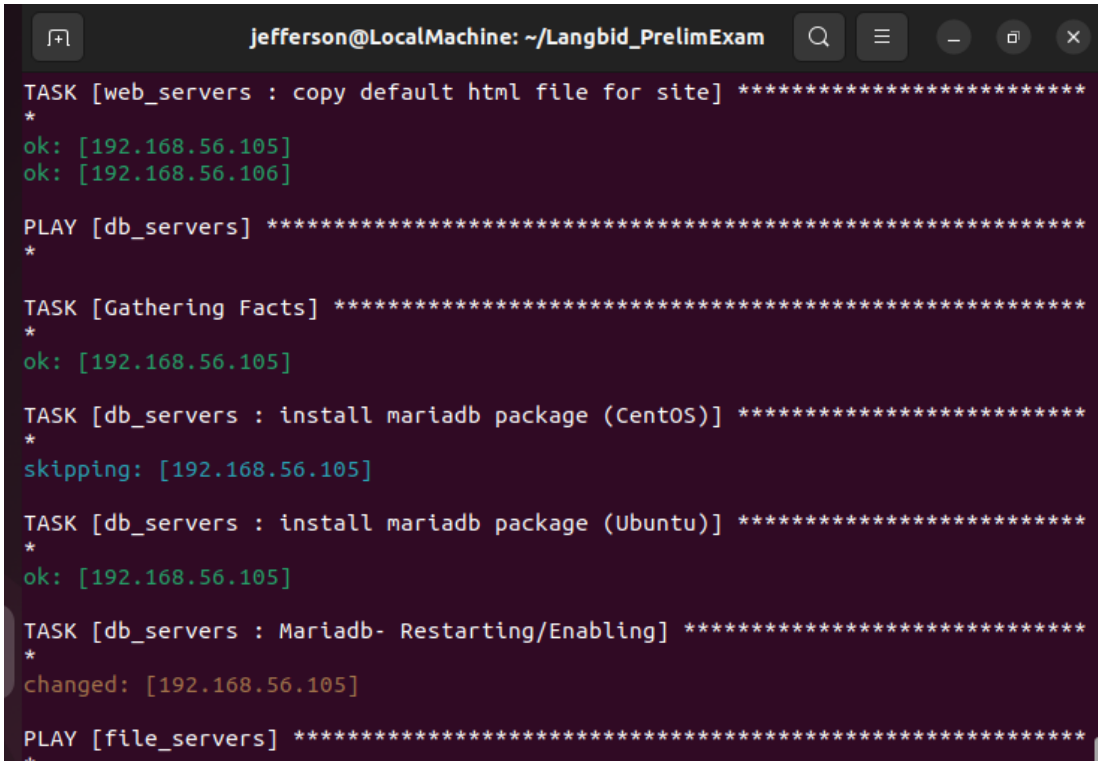
```
jefferson@LocalMachine: ~/Langbid_PrelimExam/roles/wor...
  GNU nano 6.2                                  main.yml
---

  - name: install unzip
    package:
      name: unzip

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

      mode: 0755
      owner: root
      group: root
```

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

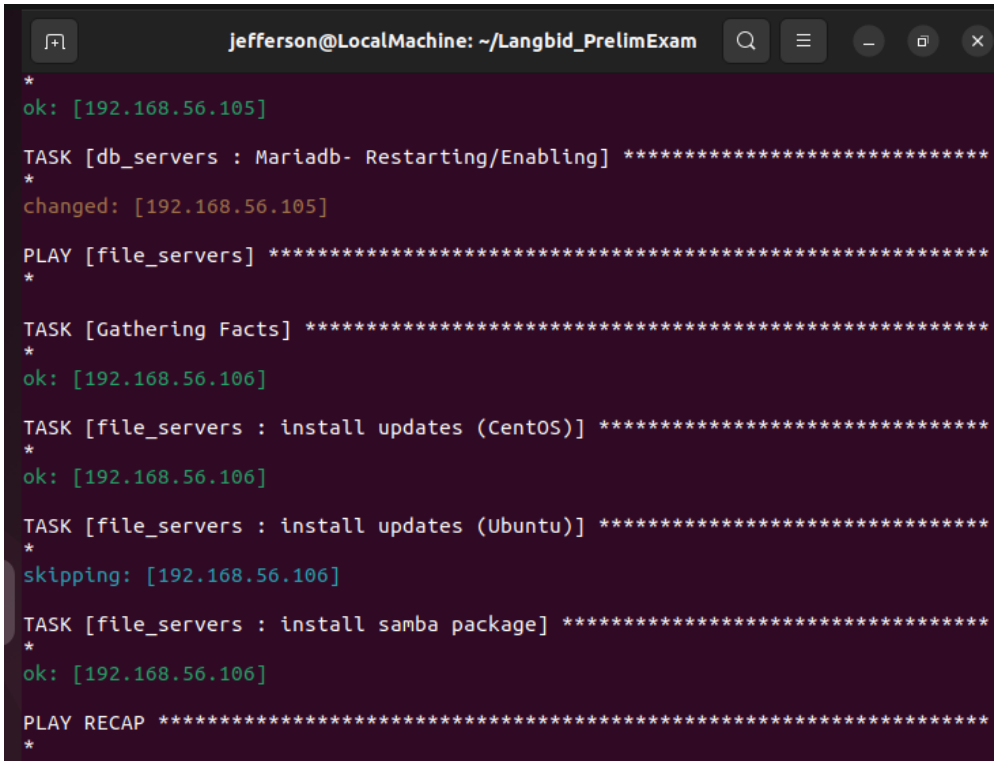
Base:



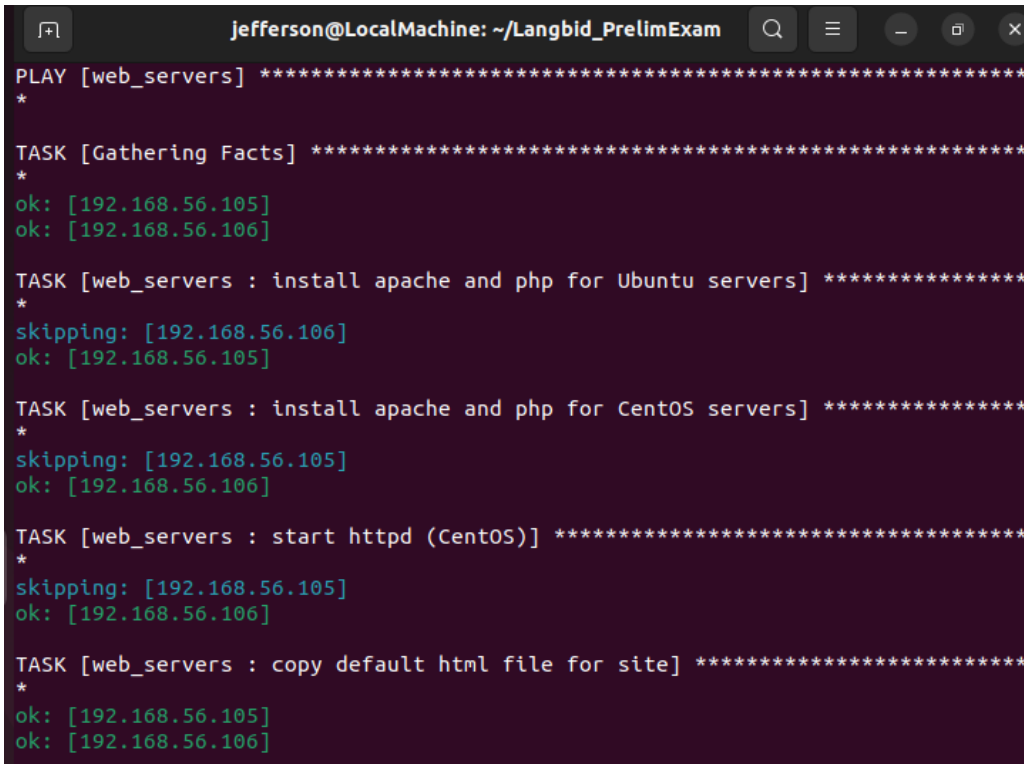
db_servers:



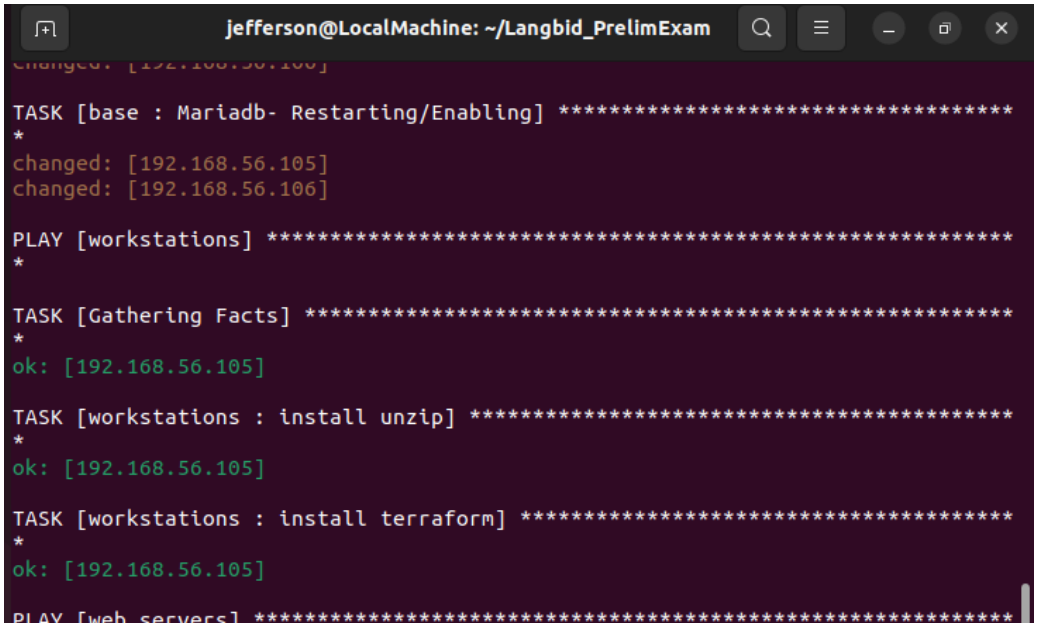
File_servers:

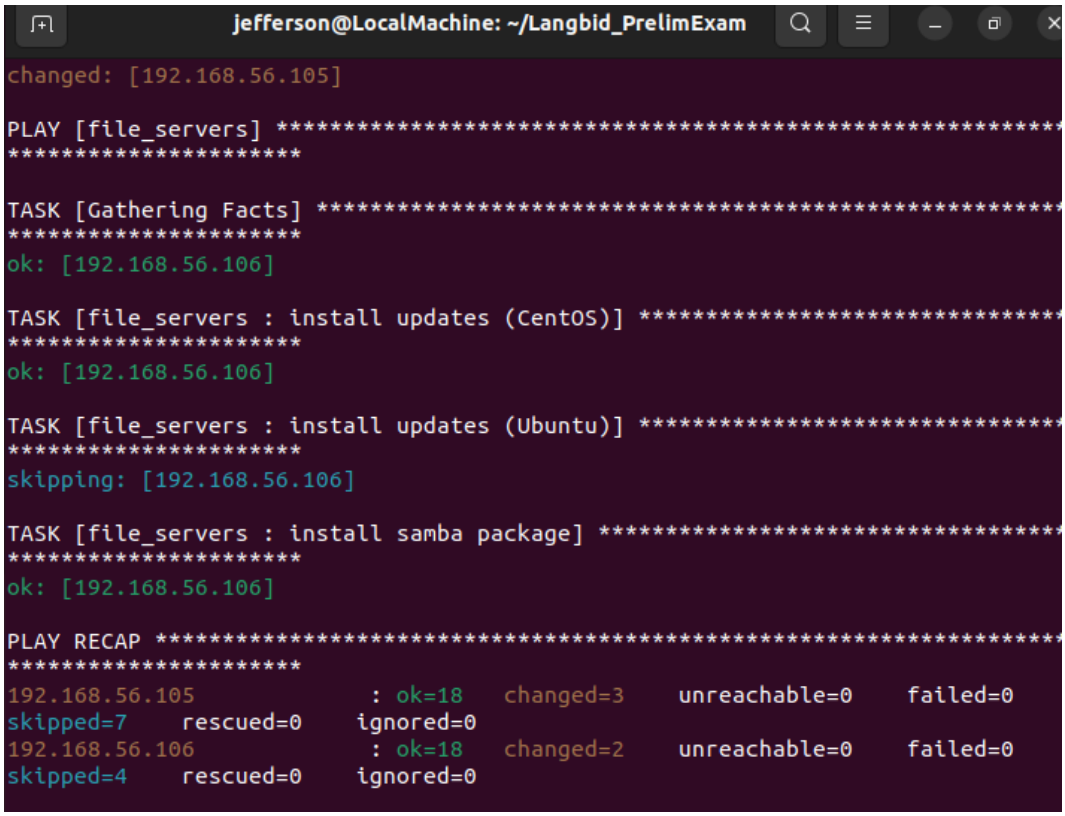


Web_servers:



Workstations:





The code ran successfully and it installed the following files or data in each specific IP address by the specific roles to Ubuntu and CentOS. The output is 3 changes that means it is installed in Ubuntu with 7 skipped because the

command is distributed to CentOS, on the other hand CentOS have 2 changed and 4 skipped because of distribution on Ubuntu.

Reflections:

Answer the following:

1. What is the importance of creating roles?
   Importance of creating a role in Ansible is to make it easier to manage and organize so it can be coded accordingly and help distribute specific installations in the playbook. Also, it helps when you are using github repository because it will ensure that everyone that can access your repository will update or know what you are doing and edited files.
2. What is the importance of managing files?

   Managing files is important because it will make the coder or programmer organize and can find the files you may need. Having you organizing or managing files may also help to free some space and delete unnecessary files in the drive.

“I affirm that I will not give or receive any unauthorized help on this activity, and that all work will be my own.”