Name: Kazuki A. Ogata	**Date Performed:** October 17, 2023
Course/Section: CPE 232 - CPE31S5	**Date Submitted:** October 18, 2023
Instructor: Engr. Roman Richard	**Semester and SY:** 1st semester S.Y 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.

```
kazuki@workstation:~/CPE232_HOA6.1$ mkdir files
kazuki@workstation:~/CPE232_HOA6.1$ ls
ansible.cfg  files  inventory  README.md  site.yml
```

Figure 3.1.1.1 - Creating a Directory

```
kazuki@workstation:~/CPE232_HOA6.1$ cd files
kazuki@workstation:~/CPE232_HOA6.1/files$ sudo nano default_site.html
[sudo] password for kazuki:
  GNU nano 6.2                                        default_site.html *

<!DOCTYPE html>
<html>

<head>
        <title>MOVIES</title>
</head>

<body>
        <h1>Top 10 Movies of all time (IMDb)</h1>
        <ol>
                <li>The Shawshank Redemption</li>
                <li>The Godfather</li>
                <li>The dark Knight</li>
                <li>The Godfather Part 2</li>
                <li>12 Angry Men</li>
                <li>Schindler's List</li>
                <li>The Lord of the Rings: The Return of the King</li>
                <li>Pulp Fiction</li>
                <li>The Lord of the Rings: The Fellowship of the Ring</li>
                <li>The Good, the Bad and the Ugly</li>
        </ol>
</body>
</html>
```

Figure 3.1.1.2 - Creating default_site.html File

```
kazuki@workstation:~/CPE232_HOA6.1/files$ cat default_site.html

<!DOCTYPE html>
<html>

<head>
        <title>MOVIES</title>
</head>

<body>
        <h1>Top 10 Movies of all time (IMDb)</h1>
        <ol>
                <li>The Shawshank Redemption</li>
                <li>The Godfather</li>
                <li>The dark Knight</li>
                <li>The Godfather Part 2</li>
                <li>12 Angry Men</li>
                <li>Schindler's List</li>
                <li>The Lord of the Rings: The Return of the King</li>
                <li>Pulp Fiction</li>
                <li>The Lord of the Rings: The Fellowship of the Ring</li>
                <li>The Good, the Bad and the Ugly</li>
        </ol>
</body>
</html>
```

Figure 3.1.1.3 - Checking if the "default_site.html" File is displaying the text.

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
```

Figure 3.1.2 - New Play in site.yml Playbook

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

```
TASK [copy default html file for site]
changed: [192.168.56.127]
changed: [192.168.56.126]
```

Figure 3.1.3.1 - Output of Modified site.yml File

It shows "changed" meaning it successfully copied the default html file to remote servers under the "web_server" play.

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.

```
kazuki@server1:~$ cat /var/www/html/index.html

<!DOCTYPE html>
<html>
<head>
        <title>MOVIES</title>
</head>

<body>
        <h1>Top 10 Movies of all time (IMDb)</h1>
        <p>
                1. The Shawshank Redemption
                2. The Godfather
                3. The dark Knight
                4. The Godfather Part 2
                5. 12 Angry Men
                6. Schindler's List
                7. The Lord of the Rings: The Return of the King
                8. Pulp Fiction
                9. The Lord of the Rings: The Fellowship of the Ring
                10. The Good, the Bad and the Ugly
        </p>
</body>
</html>
```

```
[kazuki@centos ~]$ cat /var/www/html/index.html

<!DOCTYPE html>
<html>

<head>
        <title>MOVIES</title>
</head>

<body>
        <h1>Top 10 Movies of all time (IMDb)</h1>
        <ol>
                <li>The Shawshank Redemption</li>
                <li>The Godfather</li>
                <li>The dark Knight</li>
                <li>The Godfather Part 2</li>
                <li>12 Angry Men</li>
                <li>Schindler's List</li>
                <li>The Lord of the Rings: The Return of the King</li>
                <li>Pulp Fiction</li>
                <li>The Lord of the Rings: The Fellowship of the Ring</li>
                <li>The Good, the Bad and the Ugly</li>
        </ol>
</body>
</html>
```

Figure 3.1.4.1-2 - Checking the index.html file for Ubuntu and CentOS

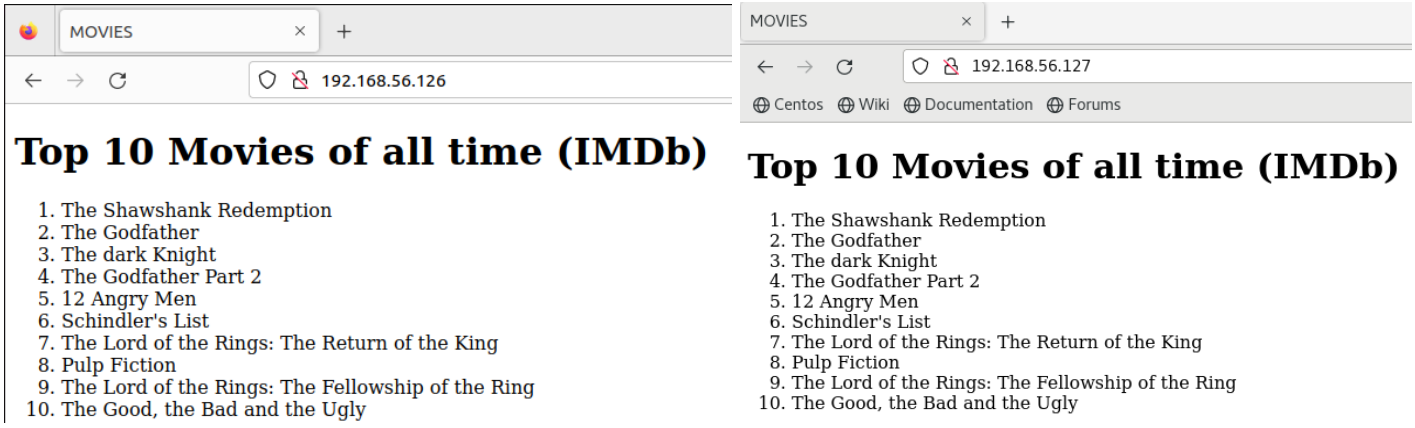
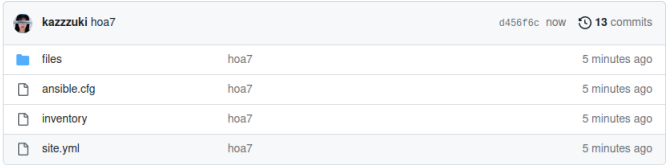


Figure 3.1.4.3-4 - Checking the html for Ubuntu and CentOS

It successfully displays the content of the default_site.html I made earlier in Ubuntu and CentOS web browsers. I put “head” in my html file, and it successfully changes the head/tab in the web browser into “MOVIES” and I also use the html syntax <lo> to list the top 10 movies of all time.

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

```
kazuki@workstation:~/CPE232_HOA6.1$ git commit -m "hoa7"
[main 69e169a] hoa7
 4 files changed, 41 insertions(+), 2 deletions(-)
 create mode 100644 ansible.cfg
 create mode 100644 files/default_site.html
kazuki@workstation:~/CPE232_HOA6.1$ git push origin main
Enumerating objects: 16, done.
Counting objects: 100% (16/16), done.
Delta compression using up to 2 threads
Compressing objects: 100% (11/11), done.
Writing objects: 100% (13/13), 1.83 KiB | 375.00 KiB/s, done.
Total 13 (delta 5), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (5/5), completed with 2 local objects.
To github.com:kazzzuki/CPE232_HOA6.1.git
   a650977..69e169a  main -> main
```



kazzzuki hoa7

Code | Blame | 25 lines (21 loc) · 500 Bytes | Code 55% faster with GitHub Copilot

```
1
2   <!DOCTYPE html>
3   <html>
4
5   <head>
6           <title>MOVIES</title>
7   </head>
8
9   <body>
10          <h1>Top 10 Movies of all time (IMDb)</h1>
11          <ol>
12                  <li>The Shawshank Redemption</li>
13                  <li>The Godfather</li>
14                  <li>The dark Knight</li>
15                  <li>The Godfather Part 2</li>
16                  <li>12 Angry Men</li>
17                  <li>Schindler's List</li>
18                  <li>The Lord of the Rings: The Return of the King</li>
19                  <li>Pulp Fiction</li>
20                  <li>The Lord of the Rings: The Fellowship of the Ring</li>
21                  <li>The Good, the Bad and the Ugly</li>
22          </ol>
23  </body>
24  </html>
```

Figure 3.1.5.1-3 - Syncing Local Repository with GitHub

I used the “git commit -m” and “git push” commands to sync my local repository in my GitHub. In checking if it successfully syncs, I went to the web browser and got to GitHub and it successfully shows that all files are updated.

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
```

Figure 3.2.1.1 - Modified site.yml File

```
PLAY [workstations] ********
skipping: no hosts matched
```

Figure 3.2.1.2 - Output of the Modified site.yml File

It skips the new play since there is no “workstation” group in the inventory file.

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

```
[workstations]
192.168.25.125
```

Figure 3.2.2 - Modified inventory File

I added 1 Ubuntu remote server (server 3) in my inventory file under the “workstations” group.

3. Run the playbook. Describe the output.

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

TASK [Gathering Facts] ****
ok: [192.168.56.125]

TASK [install unzip] ******
ok: [192.168.56.125]

TASK [install terraform] **
changed: [192.168.56.125]
```

Figure 3.2.3 - Output of the Modified stie.yml File

The “Gathering Facts” shows “ok” output meaning the IP Address in my workstations group has connections. The tasks “install unzip” and “install terraform” were successfully executed.

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

```
kazuki@server3:~$ terraform --version
Terraform v0.12.28

Your version of Terraform is out of date! The latest version
is 1.6.1. You can update by downloading from https://www.terraform.io/downloads.html
```

Figure 3.2.4 - Verifying Installation of Terraform

The output shows that the new play in my playbook was successfully executed, it installed the terraform based on the link in “src” which is version 0.12.28. The output of “terraform –version” also shows that the version of terraform is out of date and it adds the link of the newest/latest version of terraform.

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.

```
kazuki@workstation:~/CPE232_HOA6.1$ mkdir roles
kazuki@workstation:~/CPE232_HOA6.1$ cd roles
kazuki@workstation:~/CPE232_HOA6.1/roles$ mkdir base web_servers file_servers db_servers workstations
kazuki@workstation:~/CPE232_HOA6.1/roles$ ls
base  db_servers  file_servers  web_servers  workstations
kazuki@workstation:~/CPE232_HOA6.1/roles$ cd base
kazuki@workstation:~/CPE232_HOA6.1/roles/base$ mkdir tasks
kazuki@workstation:~/CPE232_HOA6.1/roles/base$ ls
tasks
kazuki@workstation:~/CPE232_HOA6.1/roles/base$ cd ../web_servers
kazuki@workstation:~/CPE232_HOA6.1/roles/web_servers$ mkdir tasks
kazuki@workstation:~/CPE232_HOA6.1/roles/web_servers$ ls
tasks
kazuki@workstation:~/CPE232_HOA6.1/roles/web_servers$ cd ../file_servers
kazuki@workstation:~/CPE232_HOA6.1/roles/file_servers$ mkdir tasks
kazuki@workstation:~/CPE232_HOA6.1/roles/file_servers$ ls
tasks
kazuki@workstation:~/CPE232_HOA6.1/roles/file_servers$ cd ../db_servers
kazuki@workstation:~/CPE232_HOA6.1/roles/db_servers$ mkdir tasks
kazuki@workstation:~/CPE232_HOA6.1/roles/db_servers$ ls
tasks
kazuki@workstation:~/CPE232_HOA6.1/roles/db_servers$ cd ../workstations
kazuki@workstation:~/CPE232_HOA6.1/roles/workstations$ mkdir tasks
kazuki@workstation:~/CPE232_HOA6.1/roles/workstations$ ls
tasks
```

Figure 3.3.2.1 - Creating Directories

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.

```
kazuki@workstation:~/CPE232_HOA6.1$ sudo nano /home/kazuki/CPE232_HOA6.1/roles/base/tasks/main.yml
kazuki@workstation:~/CPE232_HOA6.1$ cat /home/kazuki/CPE232_HOA6.1/roles/base/tasks/main.yml
  - name: install updates (CentOS)
    tags: always
    dnf:
      update_only: yes
      update_cache: yes
      use_backend: dnf
    when: ansible_distribution == "CentOS"
    async: 3600
    poll: 0

  - name: install updates (Ubuntu)
    tags: always
    apt:
      upgrade: dist
      update_cache: yes
    when: ansible_distribution == "Ubuntu"
    async: 3600
    poll: 0
```

Figure 3.3.3.1 - Contents of main.yml in /roles/base/tasks

```
kazuki@workstation:~/CPE232_HOA6.1$ sudo nano /home/kazuki/CPE232_HOA6.1/roles/web_servers/tasks/main.yml
kazuki@workstation:~/CPE232_HOA6.1$ cat /home/kazuki/CPE232_HOA6.1/roles/web_servers/tasks/main.yml
  - name: install apache and php for Ubuntu servers
    tags: apache,apache2,ubuntu
    apt:
      name:
        - apache2
        - libapache2-mod-php
      state: latest
    when: ansible_distribution == "Ubuntu"
    async: 3600
    poll: 0

  - name: install apache and php for CentOS servers
    tags: apache,centos,httpd
    dnf:
      name:
        - httpd
        - php
      use_backend: dnf
      state: latest
    when: ansible_distribution == "CentOS"
    async: 3600
    poll: 0

  - name: start httpd (CentOS)
    tags: apache, centos,httpd
    service:
      name: httpd
      state: started
      use_backend: dnf
    when: ansible_distribution == "CentOS"
    async: 3600
    poll: 0

  - 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
```

Figure 3.3.3.2 - Contents of main.yml in /roles/web_servers/tasks

```
kazuki@workstation:~/CPE232_HOA6.1$ sudo nano /home/kazuki/CPE232_HOA6.1/roles/file_servers/tasks/main.yml
kazuki@workstation:~/CPE232_HOA6.1$ cat /home/kazuki/CPE232_HOA6.1/roles/file_servers/tasks/main.yml
  - name: isntall samba package
    tags: samba
    package:
      name: samba
      state: latest
      use_backend: dnf
    async: 3600
    poll: 0
```

Figure 3.3.3.3 - Contents of main.yml in /roles/file_servers/tasks

```
kazuki@workstation:~/CPE232_HOA6.1$ sudo nano /home/kazuki/CPE232_HOA6.1/roles/db_servers/tasks/main.yml
kazuki@workstation:~/CPE232_HOA6.1$ cat /home/kazuki/CPE232_HOA6.1/roles/db_servers/tasks/main.yml
  - name: install mariadb package (CentOS)
    tags: centos, db,mariadb
    yum:
      name: mariadb-server
      state: latest
    when: ansible_distribution == "CentOS"
    async: 3600
    poll: 0

  - name: "Mariadb- Restarting/Enabling"
    service:
      name: mariadb
      state: restarted
      enabled: true
    async: 3600
    poll: 0

  - name: install mariadb package (Ubuntu)
    tags: db, mariadb,ubuntu
    apt:
      name: mariadb-server
      state: latest
    when: ansible_distribution == "Ubuntu"
    async: 3600
    poll: 0
```

Figure 3.3.3.4 - Contents of main.yml in /roles/db_servers/tasks

```
kazuki@workstation:~/CPE232_HOA6.1$ sudo nano /home/kazuki/CPE232_HOA6.1/roles/workstations/tasks/main.yml
kazuki@workstation:~/CPE232_HOA6.1$ cat /home/kazuki/CPE232_HOA6.1/roles/workstations/tasks/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
```

Figure 3.3.3.5 - Contents of main.yml in /roles/workstations/tasks

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

```
kazuki@workstation:~/CPE232_HOA6.1$ ansible-playbook --ask-become-pass site.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.126]
ok: [192.168.56.125]
ok: [192.168.56.123]
ok: [192.168.56.127]

TASK [update repository index (CentOS)] ****************************************
skipping: [192.168.56.126]
skipping: [192.168.56.125]
skipping: [192.168.56.123]
ok: [192.168.56.127]

TASK [install updates (Ubuntu)] ************************************************
skipping: [192.168.56.127]
ok: [192.168.56.123]
ok: [192.168.56.125]
ok: [192.168.56.126]

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.123]
ok: [192.168.56.127]
ok: [192.168.56.125]
ok: [192.168.56.126]
```

Figure 3.3.4.1 - Output of the Modified site.yml Playbook File

It successfully executes the pre_tasks in my playbook.

```
TASK [base : install updates (CentOS)] *************************************
skipping: [192.168.56.126]
skipping: [192.168.56.125]
skipping: [192.168.56.123]
changed: [192.168.56.127]

TASK [base : install updates (Ubuntu)] *************************************
skipping: [192.168.56.127]
changed: [192.168.56.123]
changed: [192.168.56.126]
changed: [192.168.56.125]
```

Figure 3.3.4.2 - Output of the Modified site.yml Playbook File (base)

When using roles we can see a difference in the output of the playbook, as we can in the picture the task is labeled as “base : <tasks>” which means it successfully calls the role for that grouped host.

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

TASK [Gathering Facts] *****************************************************
ok: [192.168.56.125]

TASK [workstations : install unzip] ****************************************
ok: [192.168.56.125]

TASK [workstations : install terraform] ************************************
ok: [192.168.56.125]
```

Figure 3.3.4.3 - Output of the Modified site.yml Playbook File (workstations)

When using roles we can see a difference in the output of the playbook, as we can in the picture the task is labeled as “workstations : <tasks>” which means it successfully calls the role for that grouped host.

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

TASK [Gathering Facts] *****************************************************
ok: [192.168.56.127]
ok: [192.168.56.126]

TASK [web_servers : install apache and php for Ubuntu servers] *************
skipping: [192.168.56.127]
changed: [192.168.56.126]

TASK [web_servers : install apache and php for CentOS servers] *************
skipping: [192.168.56.126]
changed: [192.168.56.127]

TASK [web_servers : start httpd (CentOS)] **********************************
skipping: [192.168.56.126]
changed: [192.168.56.127]

TASK [web_servers : copy default html file for site] ***********************
ok: [192.168.56.127]
ok: [192.168.56.126]
```

Figure 3.3.4 .4- Output of the Modified site.yml Playbook File (web_servers)

When using roles we can see a difference in the output of the playbook, as we can in the picture the task is labeled as “web_servers : <tasks>” which means it successfully calls the role for that grouped host.

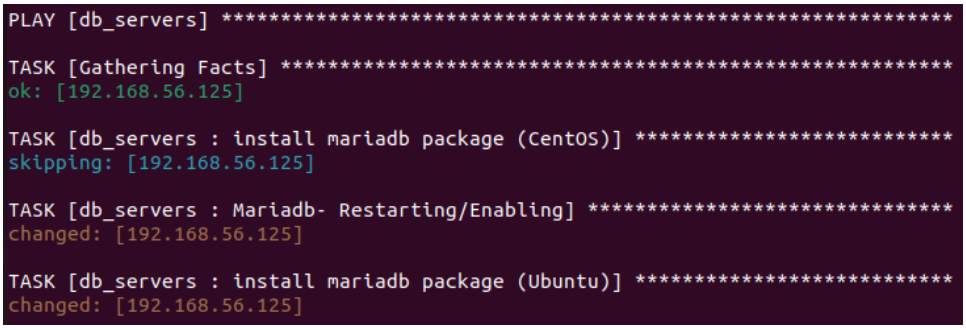


Figure 3.3.4.5 - Output of the Modified site.yml Playbook File (db_servers)

When using roles we can see a difference in the output of the playbook, as we can in the picture the task is labeled as “db_servers : <tasks>” which means it successfully calls the role for that grouped host.

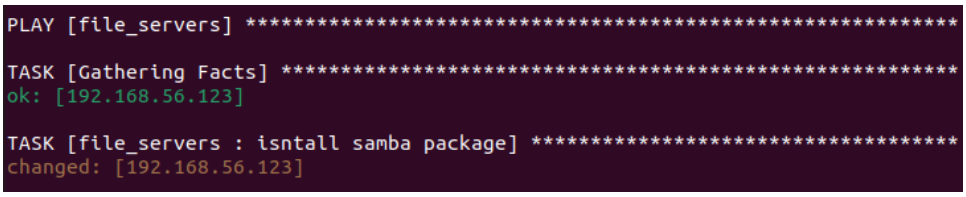


Figure 3.3.4.6 - Output of the Modified site.yml Playbook File (file_servers)

When using roles we can see a difference in the output of the playbook, as we can in the picture the task is labeled as “file_servers : <tasks>” which means it successfully calls the role for that grouped host.

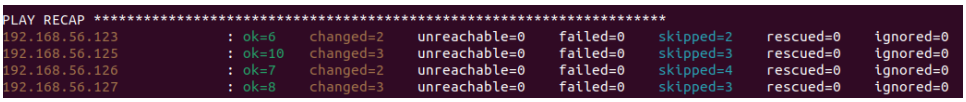


Figure 3.3.4 - Output of the Modified site.yml Playbook File (RECAP)

This shows the recap of all plays in the playbook file and there is no error,failed, or unreachable = 1, meaning there is no error encountered when executing the playbook.

Reflections:

Answer the following:

1. What is the importance of creating roles?
    - The importance of creating roles is it groups multiple tasks together into one role meaning it can be reused and share the codes efficiently. Also, when they are a group (role) they will be easier to call (fewer code lines) in our main playbook. For example, there is a DevOps Engineer who has a job for managing different web servers and databases in their company. With this, creating roles is very necessary and can make the DevOps Engineer job easier. Like it will be reusable, for instance, the DevOps Engineer has multiple playbooks for different operations for web servers and databases, by creating roles he/she can avoid code duplications and make sure that his/her work is consistent. Overall, roles in Ansible helps us in managing our playbook effectively promoting code organization, reusability, and collaboration.

2. What is the importance of managing files?
    - The importance of managing files is it makes our job easier to easily and quickly locate and access our files, especially if the files are important. Personally, on my laptop, I separate my files from the files of my siblings who are also using my laptop. With this, it promotes the organization of files, and when you want to find a specific file you can just easily go to the folder of it and access it. For example, in my first time experience in an online class in 2019, we had lots of files to download and create and submit it but once I submitted it, I'm already deleting it so I can save space for my device that time, but little did I know that one time I did not double check if I successfully submitted the file, I did not submit my file that time and receive a failing grade for that activity, starting from that day, I learned to manage my files like putting them in USB or Good drive so I can still access them in case this happens again.