Name: Peñas, Issa Victoria H.	**Date Performed:** 10/11/2022
Course/Section: CPE232 - CPE31S22	**Date Submitted:** 10/11/2022
Instructor: Dr. Jonathan V. Taylar	**Semester and SY:** 1st Semester (SY: 2021 - 2022)

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.

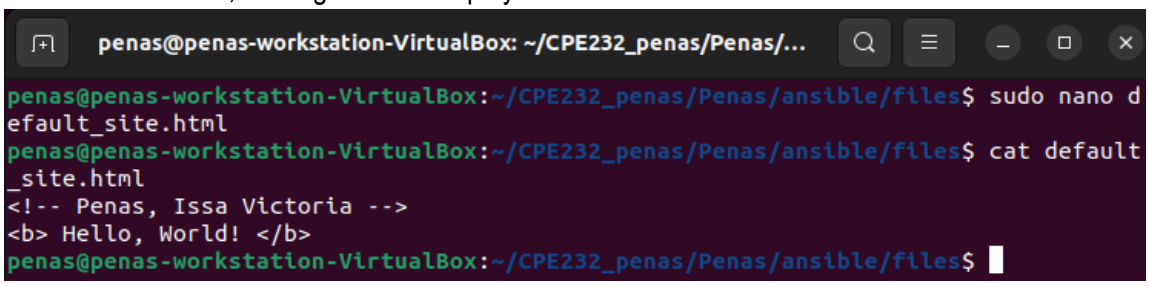


Figure 1.1. Creating a new Directory named *'files'* and a file inside the directory named *'default_site.html'* where its contents are in HTML programming language

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: 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
```

Figure 1.2. Appending a new set of command in the *'site.yml'* Playbook

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

```
penas@penas-workstation-VirtualBox:~/CPE232_penas/Penas/ansible$ ansible-playbook --ask-become-pass site.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.109]
fatal: [192.168.56.102]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: ssh: connect
fatal: [192.168.56.113]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: ssh: connect
ok: [192.168.56.107]

TASK [Install Updates (CentOS)] ************************************************
skipping: [192.168.56.109]
ok: [192.168.56.107]

TASK [Install Updates (Ubuntu)] ************************************************
skipping: [192.168.56.107]
ok: [192.168.56.109]

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.109]
ok: [192.168.56.107]

TASK [copy default html file for site] *****************************************
changed: [192.168.56.107]
ok: [192.168.56.109]

TASK [install apache and php for Ubuntu Servers] *******************************
skipping: [192.168.56.107]
ok: [192.168.56.109]

TASK [install apache and php for CentOS Server] ********************************
skipping: [192.168.56.109]
ok: [192.168.56.107]

TASK [start httpd (CentOS)] ****************************************************
skipping: [192.168.56.109]
changed: [192.168.56.107]

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

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

PLAY RECAP *********************************************************************
192.168.56.102             : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0
192.168.56.107             : ok=6    changed=2    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.56.109             : ok=5    changed=0    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0
192.168.56.113             : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0
```

Figure 1.3. After the command the assigned the IP Address in the ***web_server*** executed successfully based on the PLAY RECAP where as Server 1 and CentOS were only Affected on this Play run

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.

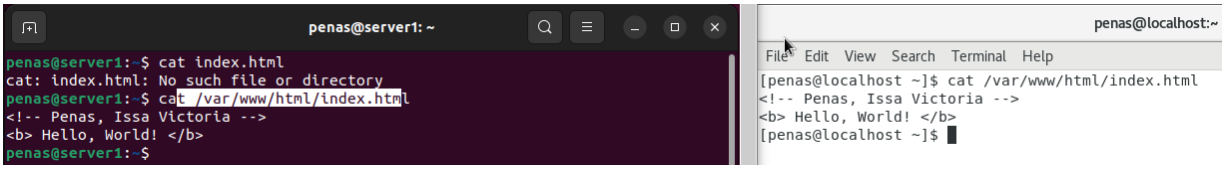


Figure 1.4. Using the cat command if the default_site.html was inserted on the IP Address of the ***web_servers***

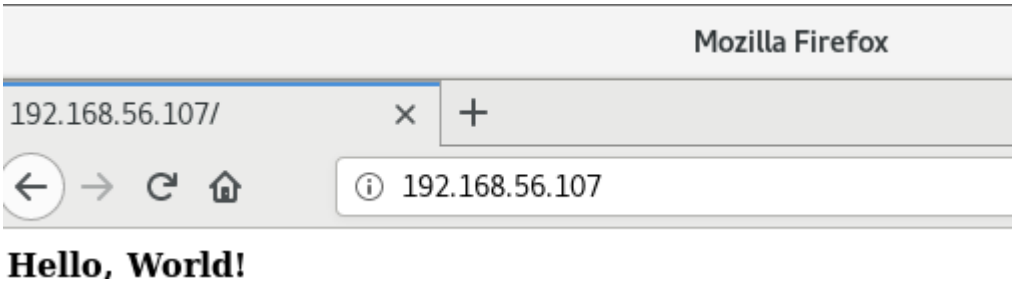


Figure 1.5. Shows the output of the HTML command in the browser - CentOS

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

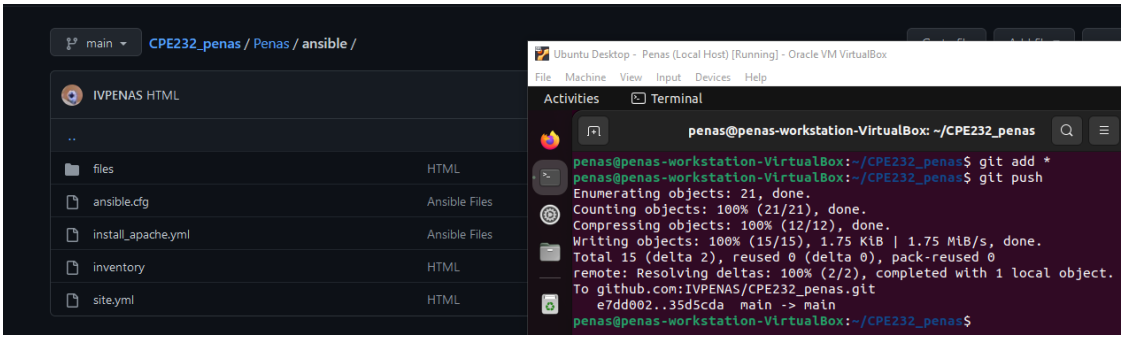


Figure 1.6. Syncing the Server into Github

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
```

Figure 2.1. Appending the new set of commands in the Playbook

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]
#Server - 1
192.168.56.109
#CentOS
192.168.56.107

[db_servers]
#Server - 2
192.168.56.102

[file_servers]
#Server - 3
192.168.56.113

[workstations]
#Server - 1
192.168.56.109
```

Figure 2.2. Adding the IP Address of Server 1 in the workstation group

3. Run the playbook. Describe the output.

```
TASK [Install Updates (CentOS)] ************************************************************************************
skipping: [192.168.56.109]
ok: [192.168.56.107]

TASK [Install Updates (Ubuntu)] ************************************************************************************
skipping: [192.168.56.107]
ok: [192.168.56.109]

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

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

TASK [install unzip] ***********************************************************************************************
ok: [192.168.56.109]

TASK [install terraform] *******************************************************************************************
changed: [192.168.56.109]

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

TASK [Gathering Facts] *********************************************************************************************
ok: [192.168.56.107]
ok: [192.168.56.109]

TASK [copy default html file for site] *****************************************************************************
ok: [192.168.56.109]
ok: [192.168.56.107]

TASK [install apache and php for Ubuntu Servers] *******************************************************************
skipping: [192.168.56.107]
ok: [192.168.56.109]

TASK [install apache and php for CentOS Server] ********************************************************************
skipping: [192.168.56.109]
ok: [192.168.56.107]

TASK [start httpd (CentOS)] ****************************************************************************************
skipping: [192.168.56.109]
ok: [192.168.56.107]

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

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

PLAY RECAP *********************************************************************************************************
192.168.56.102             : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0
192.168.56.107             : ok=6    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0
192.168.56.109             : ok=8    changed=1    unreachable=0    failed=0    skipped=3    rescued=0    ignored=0
192.168.56.113             : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0
```

Figure 2.3. The Installation of the terraform was successfully executed in Server 1 based on the Play Recap

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

```
penas@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:
    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
penas@server1:~$
```

Figure 2.4. Shows the manual or the commands available in 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.

```
penas@penas-workstation-VirtualBox:~/CPE232_penas/Penas/ansible$ cp site.yml old_s
ite.yml
penas@penas-workstation-VirtualBox:~/CPE232_penas/Penas/ansible$ ls
ansible.cfg  files  install_apache.yml  inventory  old_site.yml  site.yml
penas@penas-workstation-VirtualBox:~/CPE232_penas/Penas/ansible$
```

Figure 3.1. Copying the old site.yml commands to backup the original playbook

```
  GNU nano 6.2
---
#Penas

- hosts: all
  become: true
  pre_tasks:

  - name: Update repository index (CentOS)
    tags: always
    dnf:
      update_cache: yes
    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
```

Figure 3.2. The new *'site.yml'* and its commands

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.

```
penas@penas-workstation-VirtualBox:~/CPE232_penas/Penas/ansible/roles$ tree
.
├── base
│   └── tasks
├── db_servers
│   └── tasks
├── file_servers
│   └── tasks
├── web_servers
│   └── tasks
└── workstations
    └── tasks

10 directories, 0 files
```

Figure 3.3. Making new Directories of the Group Servers respectively

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.

```
penas@penas-workstation-VirtualBox:~/CPE232_penas/Penas/ansible/roles$ touch base/tasks/main.yml
penas@penas-workstation-VirtualBox:~/CPE232_penas/Penas/ansible/roles$ touch db_servers/tasks/main.yml
penas@penas-workstation-VirtualBox:~/CPE232_penas/Penas/ansible/roles$ touch web_servers/tasks/main.yml
penas@penas-workstation-VirtualBox:~/CPE232_penas/Penas/ansible/roles$ touch file_servers/tasks/main.yml
penas@penas-workstation-VirtualBox:~/CPE232_penas/Penas/ansible/roles$ touch workstations/tasks/main.yml
penas@penas-workstation-VirtualBox:~/CPE232_penas/Penas/ansible/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
```

Figure 3.4. Making new playbooks named *'main.yml'* of the Group Servers respectively

```
  GNU nano 6.2                                    main
---
#Penas - base

- 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"
```

Figure 3.5. The playbook of the ***base*** directory

```
GNU nano 6.2                                    main.
---
#Penas - db_servers

- hosts: db_servers
  become: true
  tasks:

  - name: Install mariadb package (CentOS)
    tags: centos,db.mariadb
    yum:
      name: mariadb-server
      state: latest
    when: ansible_distribution == "CentOS"

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

  - name: Install mariadb package (Ubuntu)
    tags: db,mariadb,ubuntu
    apt:
      name: mariadb-server
      state: latest
    when: ansible_distribution == "Ubuntu"
```

Figure 3.6. The playbook of the ***db_servers*** directory

```
---
#Penas - file_servers

- hosts: file_servers
  become: true
  tasks:

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

Figure 3.7. The playbook of the ***file_servers*** directory

```
GNU nano 6.2
---
#Penas - web_servers

- hosts: web_servers
  become: true
  tasks:

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

  - 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 Server
    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
      enabled: true
    when: ansible_distribution == "CentOS"
```

Figure 3.8. The playbook of the ***web_servers*** directory

```
GNU nano 6.2
---
#Penas - workstations

- 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
```

Figure 3.9. The playbook of the ***workstations*** directory

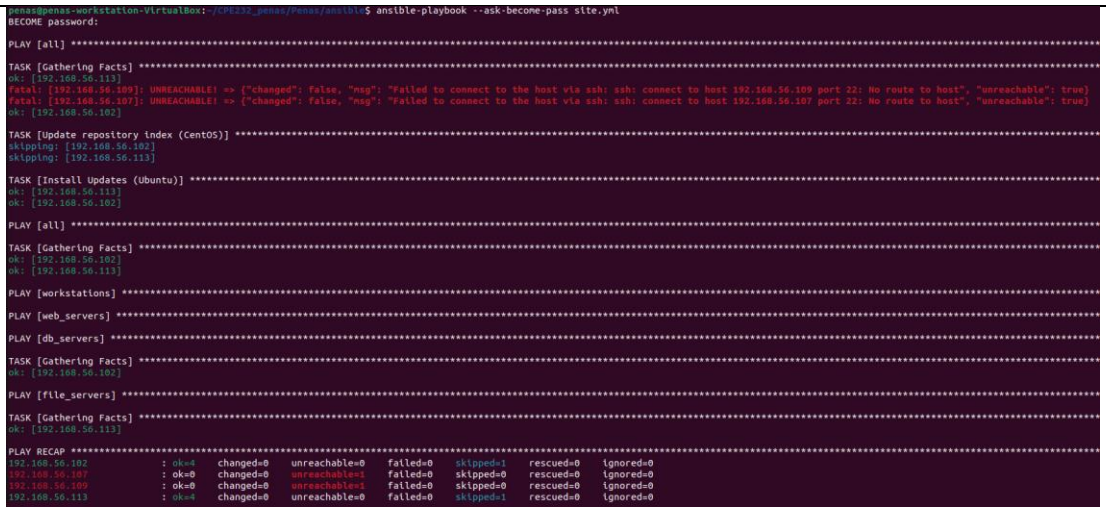
```
penas@penas-workstation-VirtualBox:~/CPE232_penas/Penas/ansible/roles$ mv base/tasks base/task
penas@penas-workstation-VirtualBox:~/CPE232_penas/Penas/ansible/roles$ mv db_servers/tasks db_servers/task
penas@penas-workstation-VirtualBox:~/CPE232_penas/Penas/ansible/roles$ mv file_servers/tasks file_servers/task
penas@penas-workstation-VirtualBox:~/CPE232_penas/Penas/ansible/roles$ mv web_servers/tasks web_servers/task
penas@penas-workstation-VirtualBox:~/CPE232_penas/Penas/ansible/roles$ mv workstation/tasks workstation/task
mv: cannot stat 'workstation/tasks': No such file or directory
penas@penas-workstation-VirtualBox:~/CPE232_penas/Penas/ansible/roles$ mv workstations/tasks workstation/task
mv: cannot move 'workstations/tasks' to 'workstation/task': No such file or directory
penas@penas-workstation-VirtualBox:~/CPE232_penas/Penas/ansible/roles$ mv workstations/tasks workstations/task
penas@penas-workstation-VirtualBox:~/CPE232_penas/Penas/ansible/roles$ tree
.
├── base
│   └── task
│       └── main.yml
├── db_servers
│   └── task
│       └── main.yml
├── file_servers
│   └── task
│       └── main.yml
├── web_servers
│   └── task
│       └── main.yml
└── workstations
    └── task
        └── main.yml

10 directories, 5 files
```

Figure 3.10. Renaming the tasks directory to task

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

```
penas@penas-workstation-VirtualBox:~/CPE232_penas/Penas/ansible$ ansible-playbook --ask-become-pass site.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************************************************************
ok: [192.168.56.109]
ok: [192.168.56.107]
fatal: [192.168.56.113]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: ssh: connect to host 192.168.56.113 port 22: Connection timed out", "unreachable": true}
fatal: [192.168.56.102]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: ssh: connect to host 192.168.56.102 port 22: Connection timed out", "unreachable": true}

TASK [Update repository index (CentOS)] ****************************************************************************************
skipping: [192.168.56.109]
ok: [192.168.56.107]

TASK [Install Updates (Ubuntu)] ************************************************************************************************
skipping: [192.168.56.107]
ok: [192.168.56.109]

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

TASK [Gathering Facts] *********************************************************************************************************
ok: [192.168.56.109]
ok: [192.168.56.107]

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

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

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

TASK [Gathering Facts] *********************************************************************************************************
ok: [192.168.56.109]
ok: [192.168.56.107]

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

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

PLAY RECAP *********************************************************************************************************************
192.168.56.102             : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0
192.168.56.107             : ok=4    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
192.168.56.109             : ok=5    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
192.168.56.113             : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0
```

Figure 3.11.1. Shows a successful output on Servers 1 and CentOS



```
penas@penas-workstation-VirtualBox:~/CPE232_penas/Penas/ansible$ ansible-playbook --ask-become-pass site.yml
BECOME password:

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.113]
fatal: [192.168.56.109]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: ssh: connect to host 192.168.56.109 port 22: No route to host", "unreachable": true}
fatal: [192.168.56.107]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: ssh: connect to host 192.168.56.107 port 22: No route to host", "unreachable": true}
ok: [192.168.56.102]

TASK [Update repository index (CentOS)] ****************************************
skipping: [192.168.56.102]
skipping: [192.168.56.113]

TASK [Install Updates (Ubuntu)] ************************************************
ok: [192.168.56.113]
ok: [192.168.56.102]

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.102]
ok: [192.168.56.113]

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

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

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.102]

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

TASK [Gathering Facts] *********************************************************
ok: [192.168.56.113]

PLAY RECAP *********************************************************************
192.168.56.102             : ok=4    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
192.168.56.107             : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0
192.168.56.109             : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0
192.168.56.113             : ok=4    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
```

Figure 3.11.1. Shows a successful output on Servers 2 and 3

GitHub: https://github.com/IVPENAS/CPE232_penas.git

Reflections:

Answer the following:

1. What is the importance of creating roles?
   - It might seem troublesome for the System Administrator to add roles among the servers one by one but, i observed managing roles made reading the playbook much easier than putting all of the set commands on one main playbook in which *debugging an error will be more easier* as error outputs will and can pinpoint which of the specific roles has an error. Playing the playbook will be more faster as commands from the playbook are separated, if a said command dictates only the ***[web_servers]*** will make changes and all of the servers that are included in the ***[web_servers]*** are opened then the directory of ***[web_servers]*** that contains their respective command will **only** run, avoiding all the unnecessary time on running other roles.

2. What is the importance of managing files?
   - First and foremost, it makes the Admin more concise and organized, having a bad trait of not managing files will lead to [1] ***misconception*** as all important files were either not labelled or were not in the right folder. [2] It can **create more errors**, where it occurred to the student unabling to sync the local repository and the servers. Having files all organized can help the System Admin to determine its management for the day in the company finishing the job efficiently.

Conclusion:

When handling a network of servers it is important to manage files to ***reduce unnecessary time*** whenever playing the Playbook, and whenever errors occur it can be ***easily debugged*** by the System Administrator in the Local Host by **Implementing Roles** in Ansible which comes hand in hand with **Managing Files**. The student was able to accomplish all the Intended Learning Outcomes (ILO's) of this Hands-On Activity, by managing the Old Site Playbook splicing its own commands on their respective Servers and Folders

whereas understanding how important its functions and how it can affect the servers if not done properly.
desc