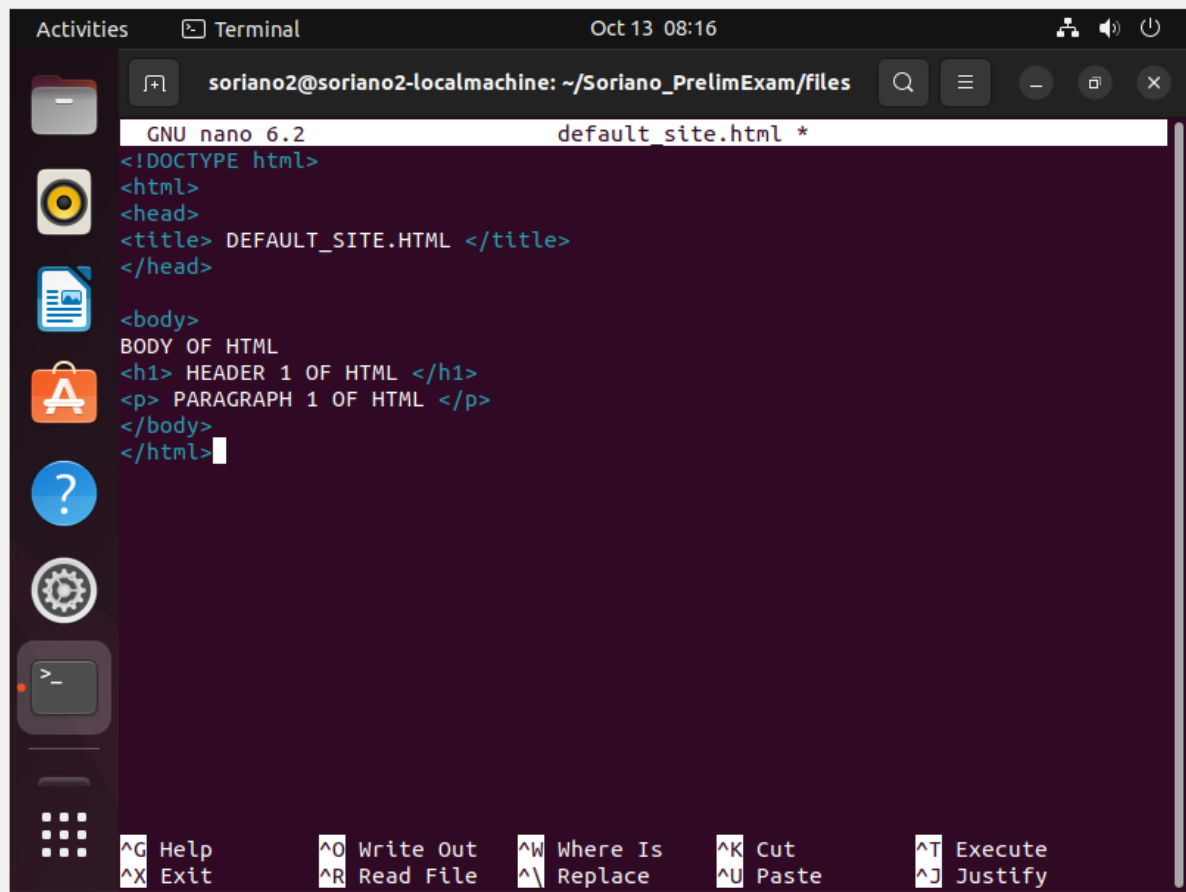| Name: Gabriel Soriano	Date Performed: October 13, 2022
Course/Section: CPE31S23	Date Submitted: October , 2022
Instructor: Engr. Taylar	Semester and SY: 1st sem - SY 2022-2023

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.

SCREENSHOTS:
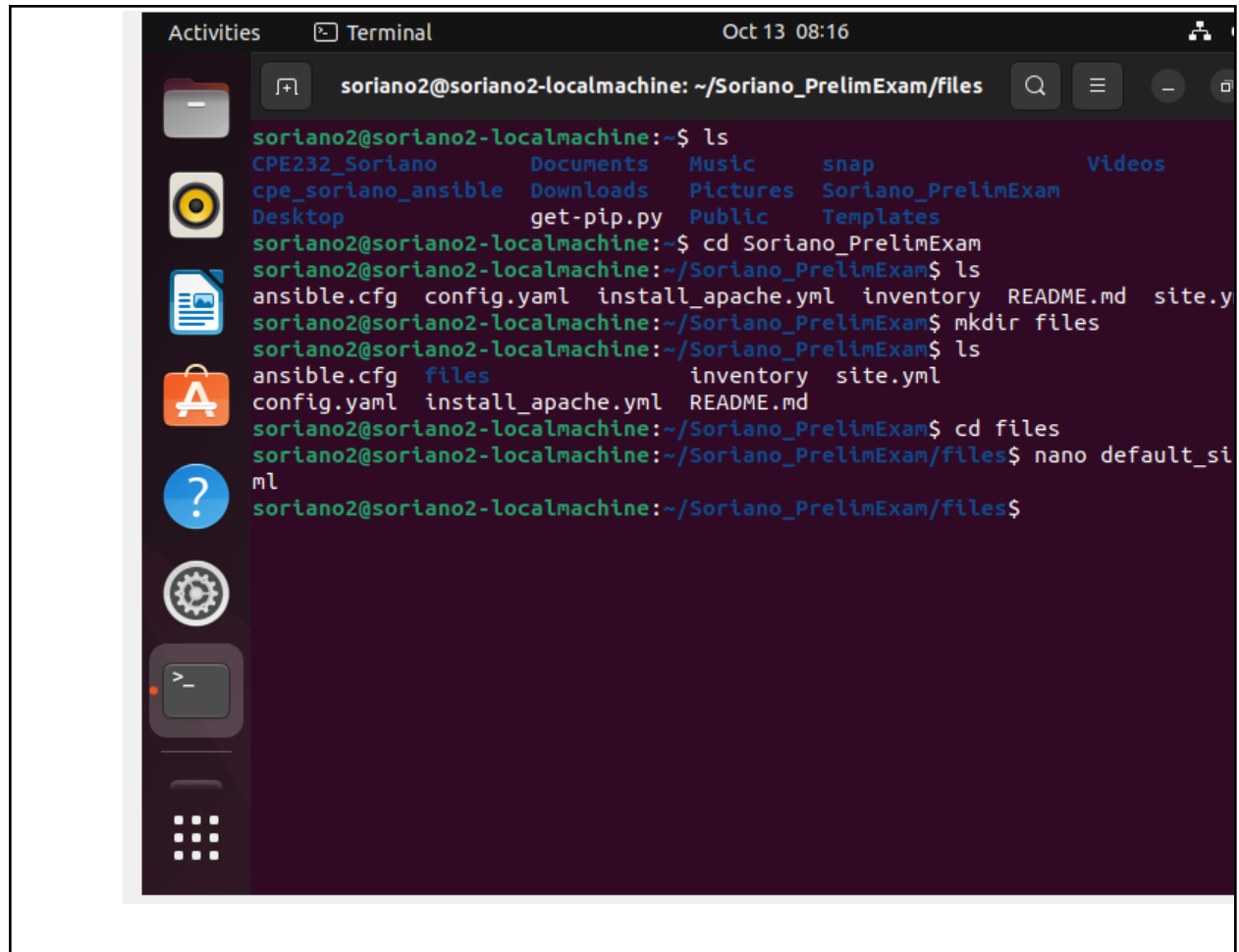
- **This shows the creation of the new directory named "files" and the creation of the new file "default_site.html". This also shows the basic syntax of the html file.**

2. Edit the *site.yml* file and just below the *web_servers* play, create a new file to copy the default html file for site:
   - name: copy default html file for site

     tags: apache, apache2, httpd
     copy:
         src: default_site.html
         dest: /var/www/html/index.html
         owner: root
         group: root
         mode: 0644
3. Run the playbook *site.yml*. Describe the changes.

   SCREENSHOT:

soriano2@soriano2-localmachine: ~/Soriano_PrelimExam

```
  GNU nano 6.2                              site.yml

 - 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
    apt:
      name:
        - apache2
        - libapache2-mod-php
      state: latest
    when: ansible_distribution == "Ubuntu"

 - name: install apache and php for CentOS servers
    dnf:
      name:
        - httpd
        - php
      state: latest
                              [ Wrote 69 lines ]
^G Help        ^O Write Out   ^W Where Is    ^K Cut      ^T Execute
^X Exit        ^R Read File   ^\ Replace     ^U Paste    ^J Justify
```

- **This shows the editing of the file site.yml and adding the command named "copy default html file for site".**

soriano2@soriano2-localmachine: ~

```
*
skipping: [192.168.56.105]
skipping: [192.168.56.106]

TASK [install updates (Ubuntu)] *************************************************
*
fatal: [192.168.56.105]: FAILED! => {"changed": false, "msg": "'/usr/bin/apt-ge
t dist-upgrade ' failed: E: dpkg was interrupted, you must manually run 'sudo d
pkg --configure -a' to correct the problem. \n", "rc": 100, "stdout": "", "stdo
ut_lines": []}
fatal: [192.168.56.106]: FAILED! => {"changed": false, "msg": "'/usr/bin/apt-ge
t dist-upgrade ' failed: E: dpkg was interrupted, you must manually run 'sudo d
pkg --configure -a' to correct the problem. \n", "rc": 100, "stdout": "", "stdo
ut_lines": []}

PLAY RECAP **********************************************************************
*
192.168.56.105             : ok=1     changed=0     unreachable=0     failed=1
skipped=1     rescued=0     ignored=0
192.168.56.106             : ok=1     changed=0     unreachable=0     failed=1
skipped=1     rescued=0     ignored=0

soriano2@soriano2-localmachine:~/Soriano_PrelimExam$ sudo dpkg --configure -a
soriano2@soriano2-localmachine:~/Soriano_PrelimExam$ cd
soriano2@soriano2-localmachine:~$ sudo dpkg --configure -a
soriano2@soriano2-localmachine:~$ ls Soriano_PrelimExam
ansible.cfg   files              inventory  site.yml
config.yaml   install_apache.yml  README.md
soriano2@soriano2-localmachine:~$
```

soriano2@server1-Server1: ~

```
Files initrd image: /boot/initrd.img-5.15.0-46-generic
Found linux image: /boot/vmlinuz-5.15.0-43-generic
Found initrd image: /boot/initrd.img-5.15.0-43-generic
Found memtest86+ image: /boot/memtest86+.elf
Found memtest86+ image: /boot/memtest86+.bin
Warning: os-prober will not be executed to detect other bootable partitions.
Systems on them will not be added to the GRUB boot configuration.
Check GRUB_DISABLE_OS_PROBER documentation entry.
done
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for php8.1-cli (8.1.2-1ubuntu2.5) ...
Processing triggers for dbus (1.12.20-2ubuntu4) ...
Processing triggers for shared-mime-info (2.1-2) ...
Processing triggers for libapache2-mod-php8.1 (8.1.2-1ubuntu2.5) ...
Processing triggers for install-info (6.8-4build1) ...
Processing triggers for mailcap (3.70+nmu1ubuntu1) ...
Setting up libedataserver-1.2-26:amd64 (3.44.4-0ubuntu1) ...
Setting up libecal-2.0-1:amd64 (3.44.4-0ubuntu1) ...
Processing triggers for fontconfig (2.13.1-4.2ubuntu5) ...
Setting up libebook-contacts-1.2-3:amd64 (3.44.4-0ubuntu1) ...
Setting up gnome-control-center (1:41.7-0ubuntu0.22.04.5) ...
Setting up libedataserverui-1.2-3:amd64 (3.44.4-0ubuntu1) ...
Setting up libebackend-1.2-10:amd64 (3.44.4-0ubuntu1) ...
Setting up libedata-cal-2.0-1:amd64 (3.44.4-0ubuntu1) ...
Setting up libedata-book-1.2-26:amd64 (3.44.4-0ubuntu1) ...
Setting up libebook-1.2-20:amd64 (3.44.4-0ubuntu1) ...
Setting up evolution-data-server (3.44.4-0ubuntu1) ...
Setting up gnome-shell (42.4-0ubuntu0.22.04.1) ...
```
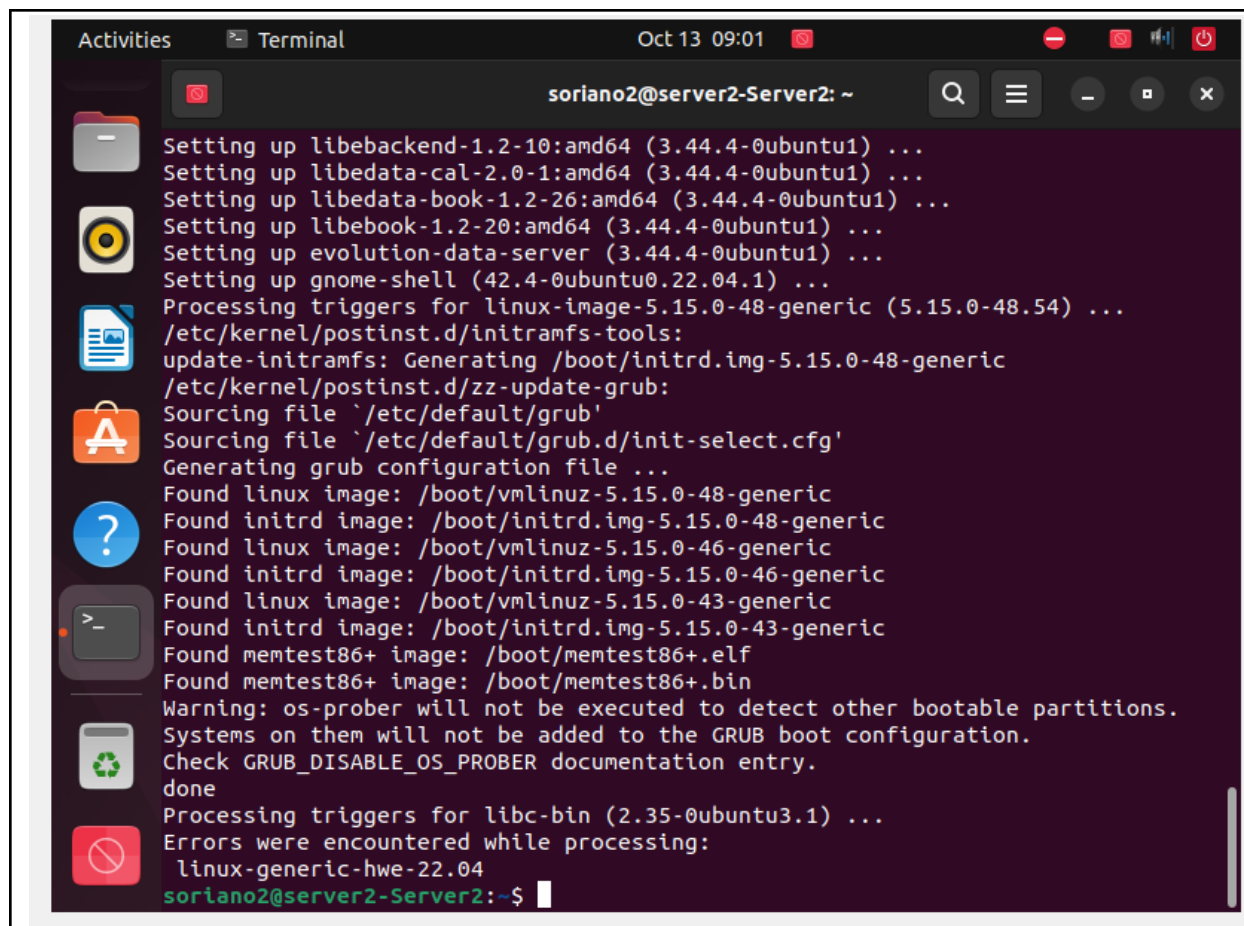
soriano2@server2-Server2: ~

```
Setting up libebackend-1.2-10:amd64 (3.44.4-0ubuntu1) ...
Setting up libedata-cal-2.0-1:amd64 (3.44.4-0ubuntu1) ...
Setting up libedata-book-1.2-26:amd64 (3.44.4-0ubuntu1) ...
Setting up libebook-1.2-20:amd64 (3.44.4-0ubuntu1) ...
Setting up evolution-data-server (3.44.4-0ubuntu1) ...
Setting up gnome-shell (42.4-0ubuntu0.22.04.1) ...
Processing triggers for linux-image-5.15.0-48-generic (5.15.0-48.54) ...
/etc/kernel/postinst.d/initramfs-tools:
update-initramfs: Generating /boot/initrd.img-5.15.0-48-generic
/etc/kernel/postinst.d/zz-update-grub:
Sourcing file `/etc/default/grub'
Sourcing file `/etc/default/grub.d/init-select.cfg'
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-5.15.0-48-generic
Found initrd image: /boot/initrd.img-5.15.0-48-generic
Found linux image: /boot/vmlinuz-5.15.0-46-generic
Found initrd image: /boot/initrd.img-5.15.0-46-generic
Found linux image: /boot/vmlinuz-5.15.0-43-generic
Found initrd image: /boot/initrd.img-5.15.0-43-generic
Found memtest86+ image: /boot/memtest86+.elf
Found memtest86+ image: /boot/memtest86+.bin
Warning: os-prober will not be executed to detect other bootable partitions.
Systems on them will not be added to the GRUB boot configuration.
Check GRUB_DISABLE_OS_PROBER documentation entry.
done
Processing triggers for libc-bin (2.35-0ubuntu3.1) ...
Errors were encountered while processing:
  linux-generic-hwe-22.04
soriano2@server2-Server2:~$
```

- **The command of running the playbook has run into some errors which can be seen on the screenshot above. The said error has been resolved by running the command "sudo dpkg –configure -a" and "sudo apt –fix-broken install" on both of the servers 1 and 2.**

```
soriano2@soriano2-localmachine: ~/Soriano...          Q  ≡  —  □  ✕

soriano2@soriano2-localmachine:~/Soriano_PrelimExam$ nano site.yml
soriano2@soriano2-localmachine:~/Soriano_PrelimExam$ ansible-playbo
ok --ask-become-pass site.yml
BECOME password:

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

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

TASK [install update (CentOS)] **************************************
*************
skipping: [192.168.56.105]
skipping: [192.168.56.106]

TASK [install updates (Ubuntu)] *************************************
```

```
soriano2@soriano2-localmachine: ~/Soriano...          Q  ≡  —  □  ✕

0.46.46 is installed"]}
ok: [192.168.56.105]

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

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

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

TASK [install apache and php for Ubuntu servers] ******************
*************
ok: [192.168.56.105]

TASK [install apache and php for CentOS servers] ******************
*************
skipping: [192.168.56.105]
```

```
soriano2@soriano2-localmachine: ~/Soriano...

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

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

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

TASK [install mariadb package (Ubuntu)] **********************************
*************
fatal: [192.168.56.105]: FAILED! => {"changed": false, "msg": "No p
ackage matching 'mariadb_server' is available"}

PLAY RECAP ***************************************************************
*************
192.168.56.105             : ok=6      changed=0    unreachable=0
failed=1      skipped=3     rescued=0     ignored=0
192.168.56.106             : ok=1      changed=0    unreachable=0
```

```
soriano2@soriano2-localmachine: ~/Soriano...

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

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

TASK [install mariadb package (Ubuntu)] ***************************
*************
fatal: [192.168.56.105]: FAILED! => {"changed": false, "msg": "No p
ackage matching 'mariadb_server' is available"}

PLAY RECAP *******************************************************
*************
192.168.56.105                 : ok=6    changed=0    unreachable=0
failed=1    skipped=3    rescued=0    ignored=0
192.168.56.106                 : ok=1    changed=0    unreachable=0
failed=1    skipped=1    rescued=0    ignored=0

soriano2@soriano2-localmachine:~/Soriano_PrelimExam$
```
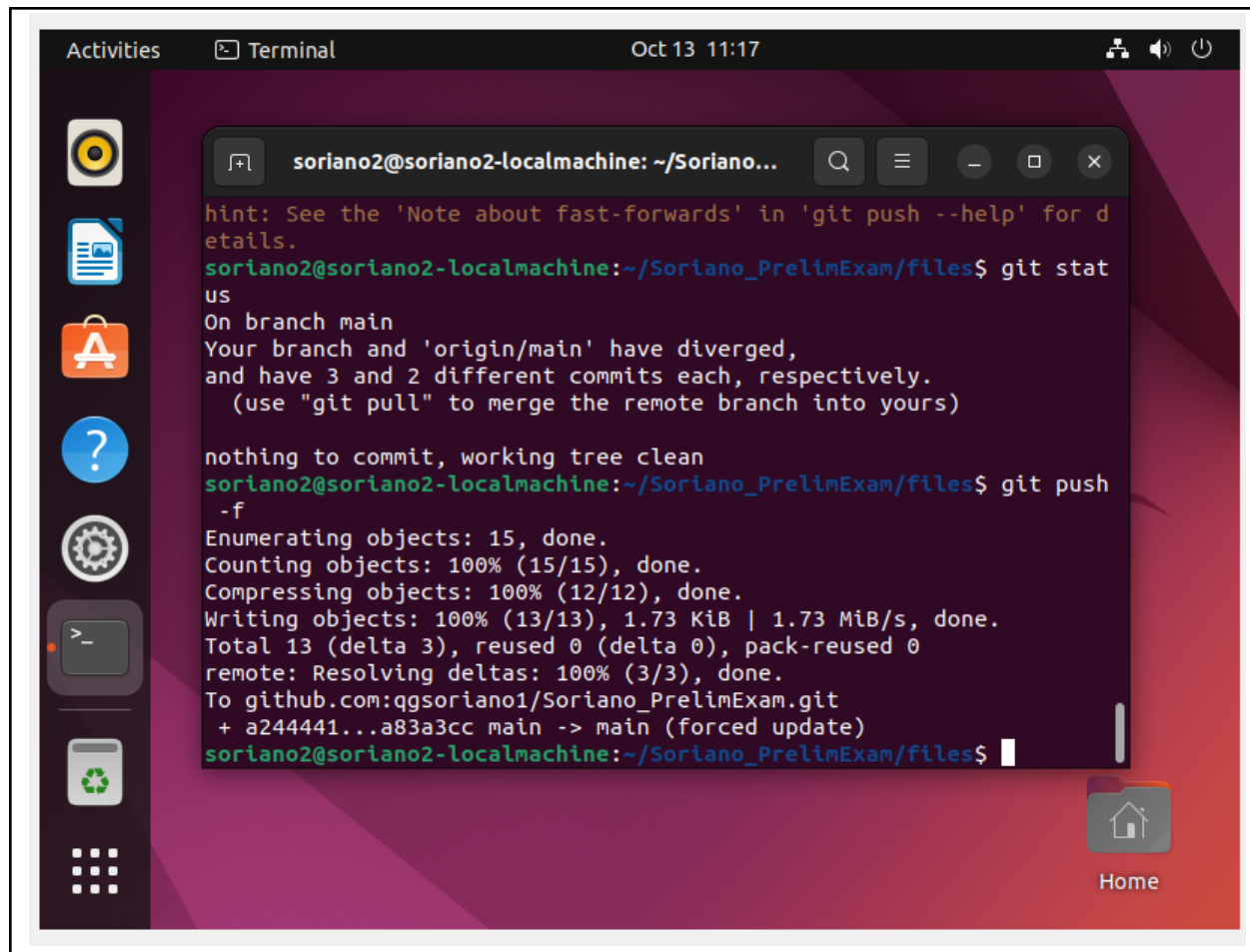
- **The running of the playbook is good because of the correct commands. Only the commands for the CentOS is having an error because I didn't use CentOS, I used another Ubuntu as server 2 instead.**

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.

   SCREENSHOTS:

192.168.56.105

BODY OF HTML

HEADER 1 OF HTML

PARAGRAPH 1 OF HTML

BODY OF HTML

HEADER 1 OF HTML

PARAGRAPH 1 OF HTML

- **These screenshots show that the typing of the appropriate ip addresses in their browsers will result in the created html file earlier. The two ip addresses with successful execution in the browser are 192.168.56.105 and 192.168.56.106. There were some errors encountered but it was resolved properly. The screenshot is proof that there are no unreachable and failed tasks to do.**

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

   SCREENSHOT:

soriano2@soriano2-localmachine: ~/Soriano...

```
hint: See the 'Note about fast-forwards' in 'git push --help' for d
etails.
soriano2@soriano2-localmachine:~/Soriano_PrelimExam/files$ git stat
us
On branch main
Your branch and 'origin/main' have diverged,
and have 3 and 2 different commits each, respectively.
  (use "git pull" to merge the remote branch into yours)

nothing to commit, working tree clean
soriano2@soriano2-localmachine:~/Soriano_PrelimExam/files$ git push
 -f
Enumerating objects: 15, done.
Counting objects: 100% (15/15), done.
Compressing objects: 100% (12/12), done.
Writing objects: 100% (13/13), 1.73 KiB | 1.73 MiB/s, done.
Total 13 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), done.
To github.com:qgsoriano1/Soriano_PrelimExam.git
 + a244441...a83a3cc main -> main (forced update)
soriano2@soriano2-localmachine:~/Soriano_PrelimExam/files$
```

Home

![profile] qgsoriano1 commit on oct 13, 2022 11:08am ...		12 minutes ago	⟲ 6
📁 files	commit on oct 13, 2022 11:08am	12 minutes ago	
📄 .config.yaml.swp	committetetjeltkj	7 days ago	
📄 README.md	Initial commit	21 days ago	
📄 ansible.cfg	last commit PrelimExam at 12:13pm	21 days ago	
📄 config.yaml	last commit PrelimExam at 12:13pm	21 days ago	
📄 install_apache.yml	last commit PrelimExam at 12:13pm	21 days ago	
📄 inventory	commit on oct 13, 2022 11:08am	12 minutes ago	
📄 site.yml	commit on oct 13, 2022 11:08am	12 minutes ago	

- These screenshots show that the uploading of the files in the repository is good and successful.

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:
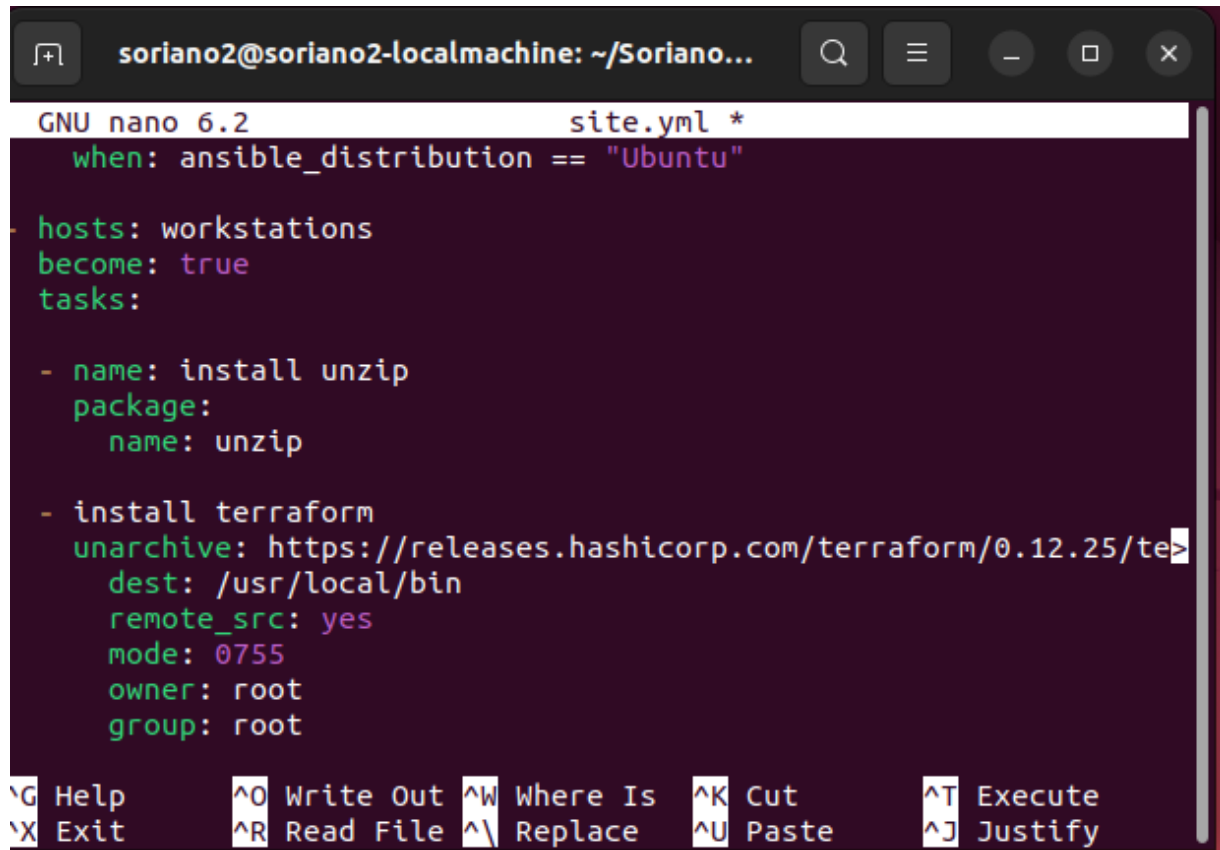
                                                                                 src:
     https://releases.hashicorp.com/terraform/0.12.28/terraform_0.12.28_linux_amd64.zip
         dest: /usr/local/bin
         remote_src: yes
         mode: 0755
         owner: root

group: root

SCREENSHOT:

```
soriano2@soriano2-localmachine: ~/Soriano...        Q   ≡   —   □   ✕

  GNU nano 6.2                        site.yml *
    when: ansible_distribution == "Ubuntu"

- hosts: workstations
  become: true
  tasks:

  - name: install unzip
    package:
      name: unzip

  - install terraform
    unarchive: https://releases.hashicorp.com/terraform/0.12.25/te>
      dest: /usr/local/bin
      remote_src: yes
      mode: 0755
      owner: root
      group: root

^G Help        ^O Write Out ^W Where Is  ^K Cut      ^T Execute
^X Exit        ^R Read File ^\ Replace    ^U Paste    ^J Justify
```

- **This shows the editing of the site.yml and adding of the task of installing the terraform file from an remote source with the url address provided.**

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

SCREENSHOT:
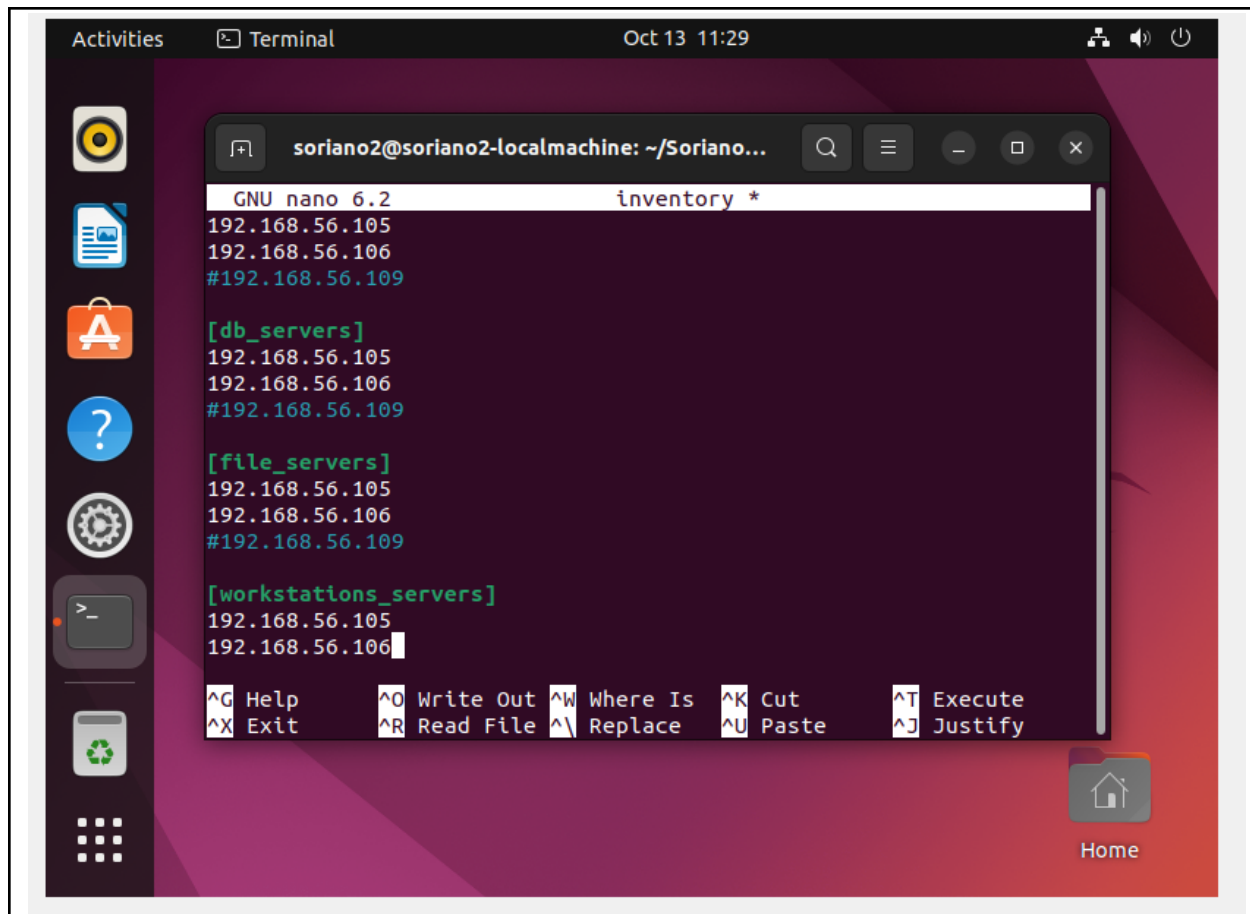
- **This shows the adding of the workstations servers ip addresses.**

3. Run the playbook. Describe the output.

   SCREENSHOT:

- **This shows the successful running of the ansible file site.yml. There were errors encountered during the process but the errors were soon resolved.**

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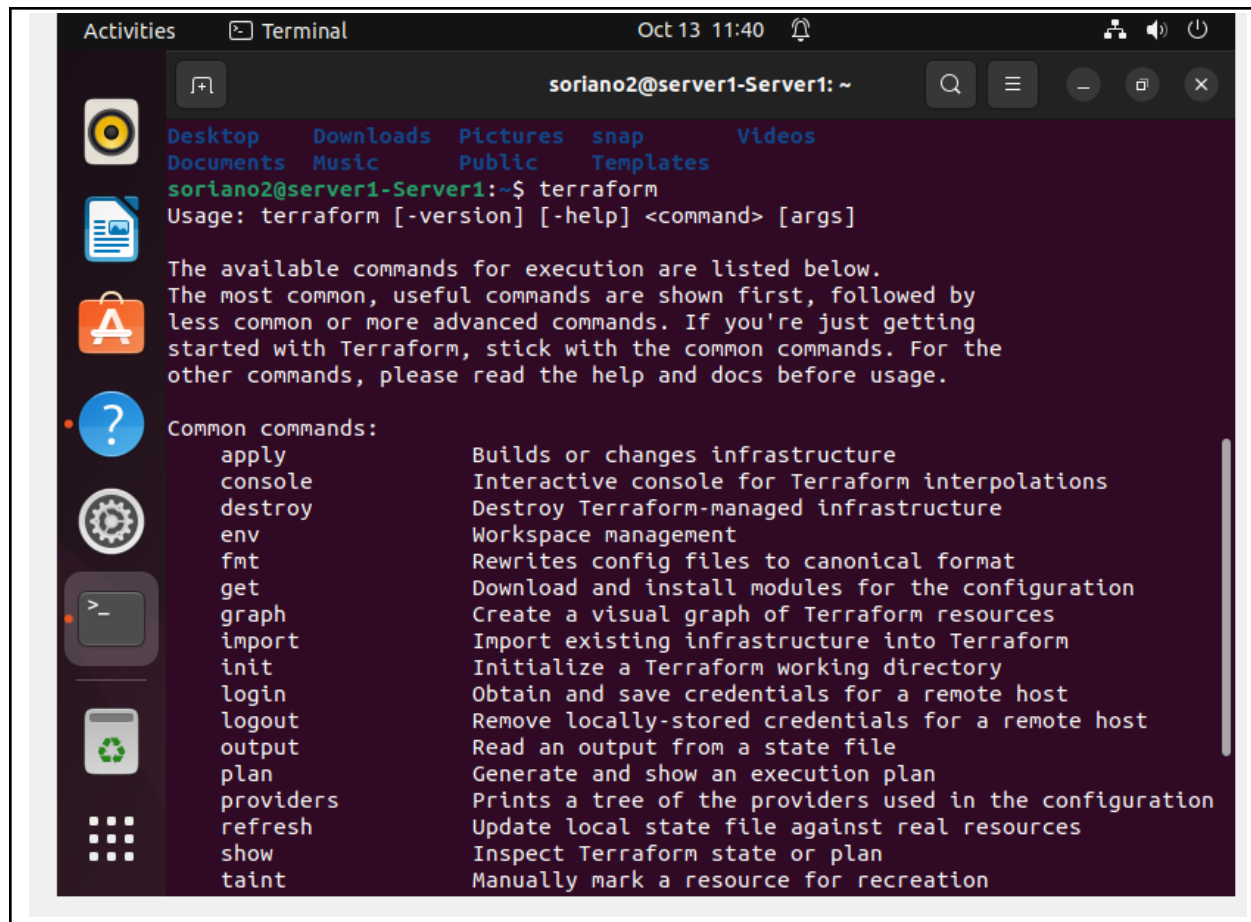
SCREENSHOT:

soriano2@server1-Server1: ~

```
Desktop     Downloads    Pictures   snap       Videos
Documents   Music        Public     Templates
soriano2@server1-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
```

- **These images show the successful installation of terraform in each of the specified servers.**

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)

```yaml
---
- 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
```
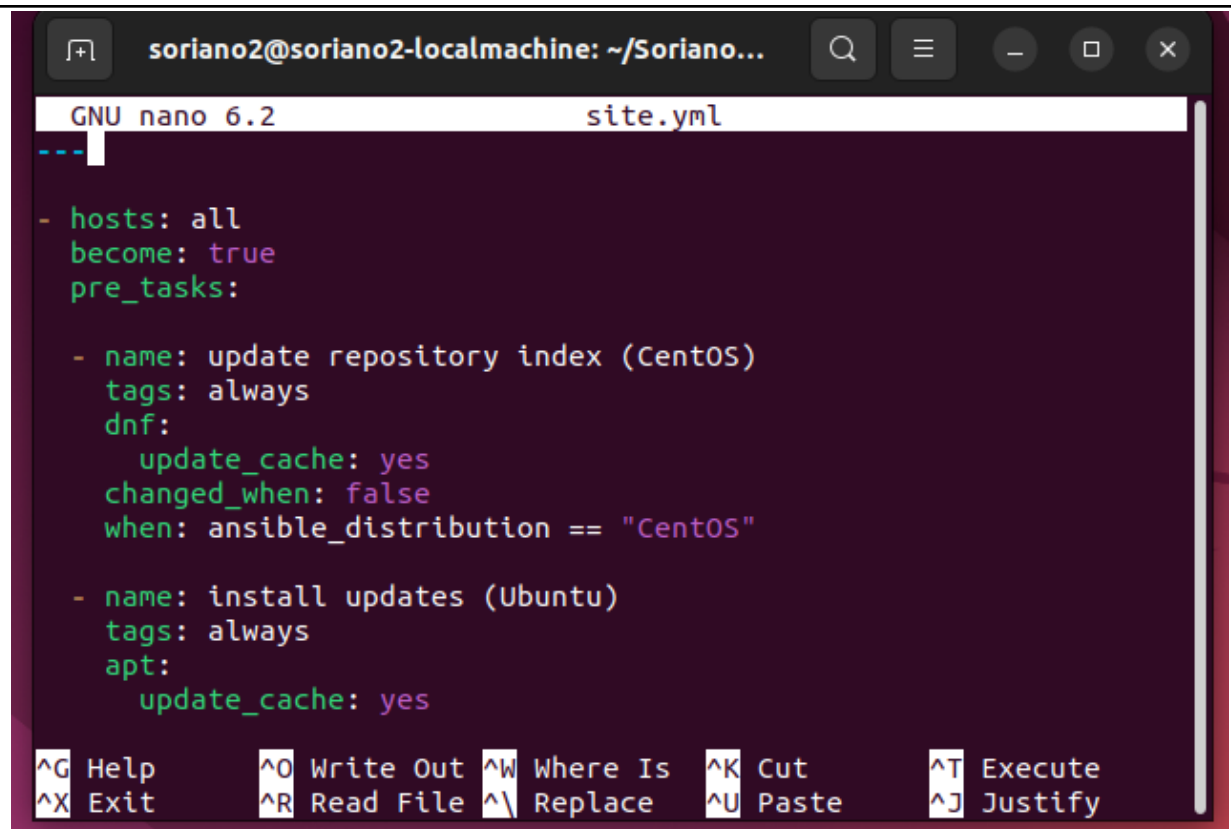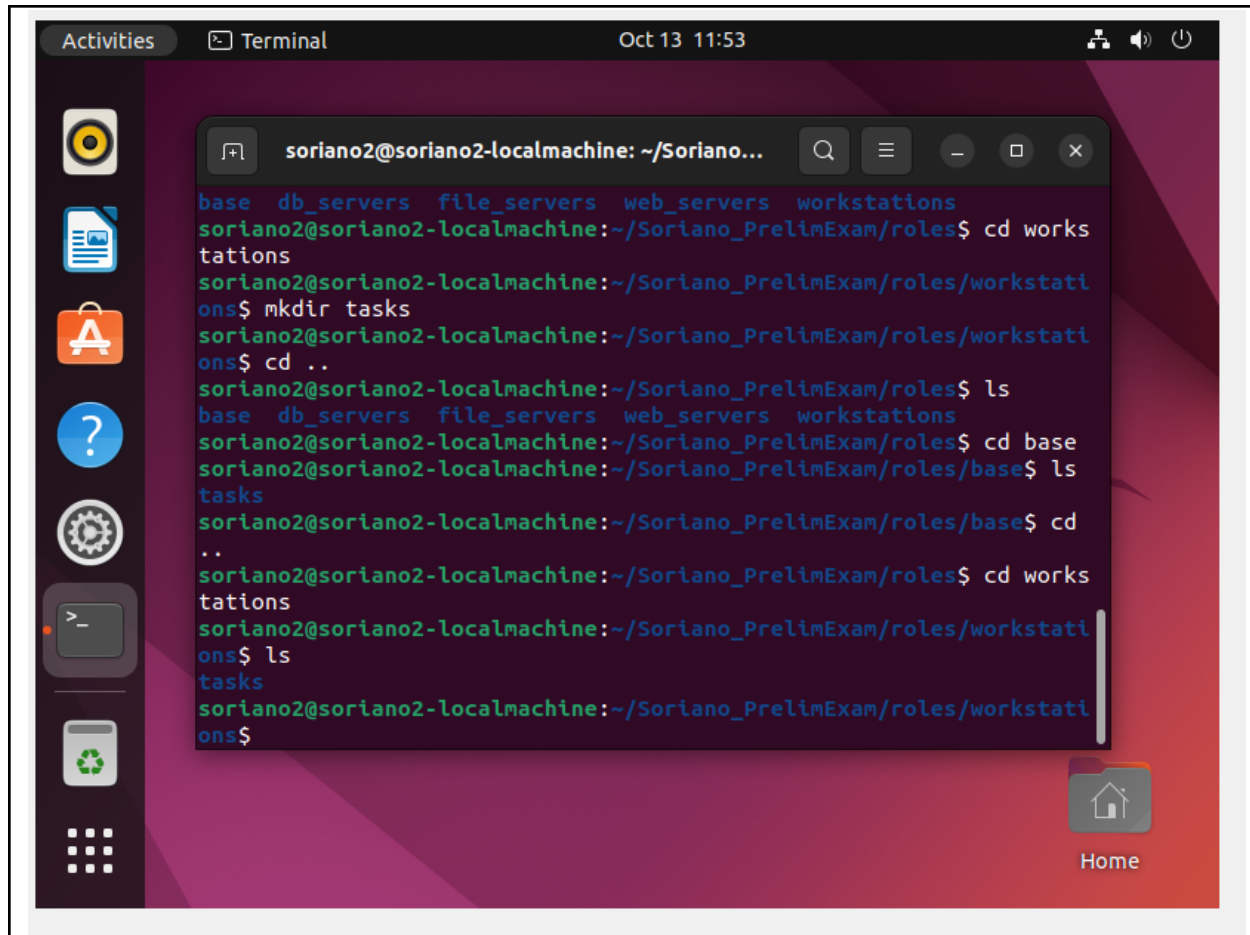
Save the file and exit.

SCREENSHOT:

```
GNU nano 6.2                          site.yml
---

- 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

^G Help        ^O Write Out ^W Where Is  ^K Cut       ^T Execute
^X Exit        ^R Read File ^\ Replace   ^U Paste     ^J Justify
```

- **This shows the adding of the roles on each of the group servers, and also the updating of the repository command.**

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.

   SCREENSHOT:
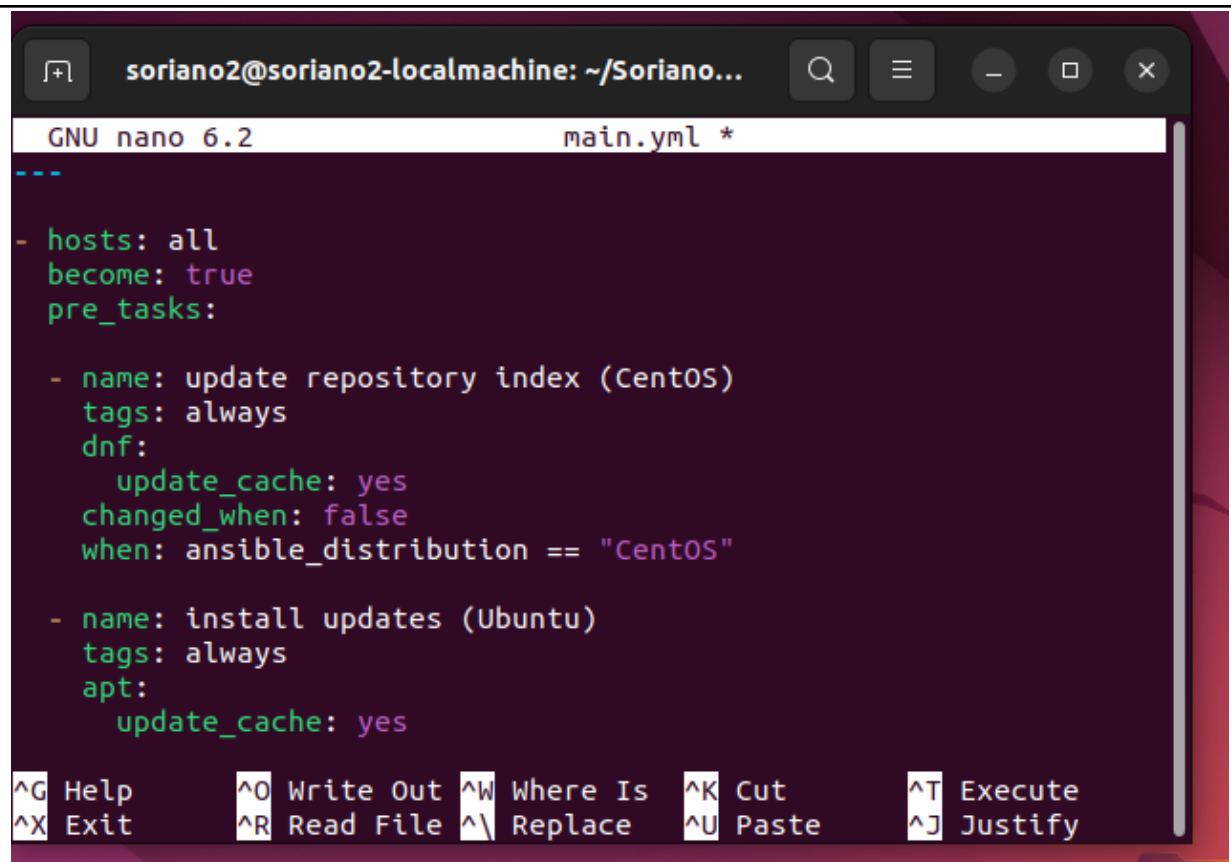
- **This shows the creation of multiple directories under the directory Soriano_PrelimExam. The directories created are seen above, which is base, db_servers, file_servers, web_servers, and workstations directory. Each of these directory has a single directory inside them, which is name tasks.**

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.

SCREENSHOTS:

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

- 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
```

^G Help   ^O Write Out ^W Where Is ^K Cut   ^T Execute
^X Exit   ^R Read File ^\ Replace   ^U Paste   ^J Justify

- **This the main.yml inside the tasks directory in the base directory.**

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

- 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
                          [ Wrote 232 lines ]
^G Help       ^O Write Out ^W Where Is   ^K Cut       ^T Execute
^X Exit       ^R Read File ^\ Replace    ^U Paste     ^J Justify
```

- **This the main.yml inside the tasks directory in the db_servers directory.**
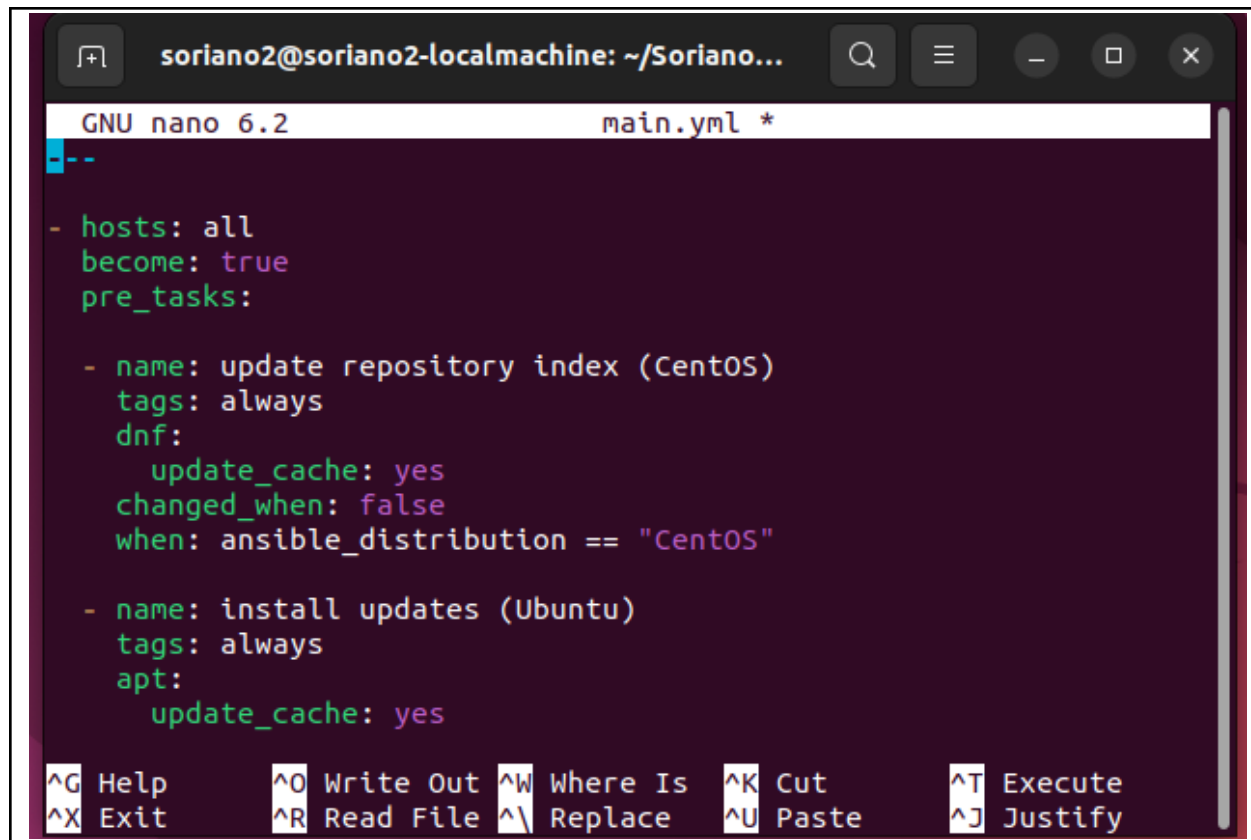
```
GNU nano 6.2                          main.yml *
---

- 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


^G Help         ^O Write Out ^W Where Is   ^K Cut       ^T Execute
^X Exit         ^R Read File ^\ Replace    ^U Paste     ^J Justify
```

- **This the main.yml inside the tasks directory in the file_servers directory.**

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


- 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

^G Help       ^O Write Out ^W Where Is  ^K Cut      ^T Execute
^X Exit       ^R Read File ^\ Replace   ^U Paste    ^J Justify
```

- **This the main.yml inside the tasks directory in the web_servers directory.**

```
 ┌─┐
 │+│   soriano2@soriano2-localmachine: ~/Soriano...    🔍  ☰   –  ▢  ✕
 └─┘
  GNU nano 6.2                        main.yml *
---

- 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

^G Help        ^O Write Out ^W Where Is  ^K Cut     ^T Execute
^X Exit        ^R Read File ^\ Replace   ^U Paste   ^J Justify
```
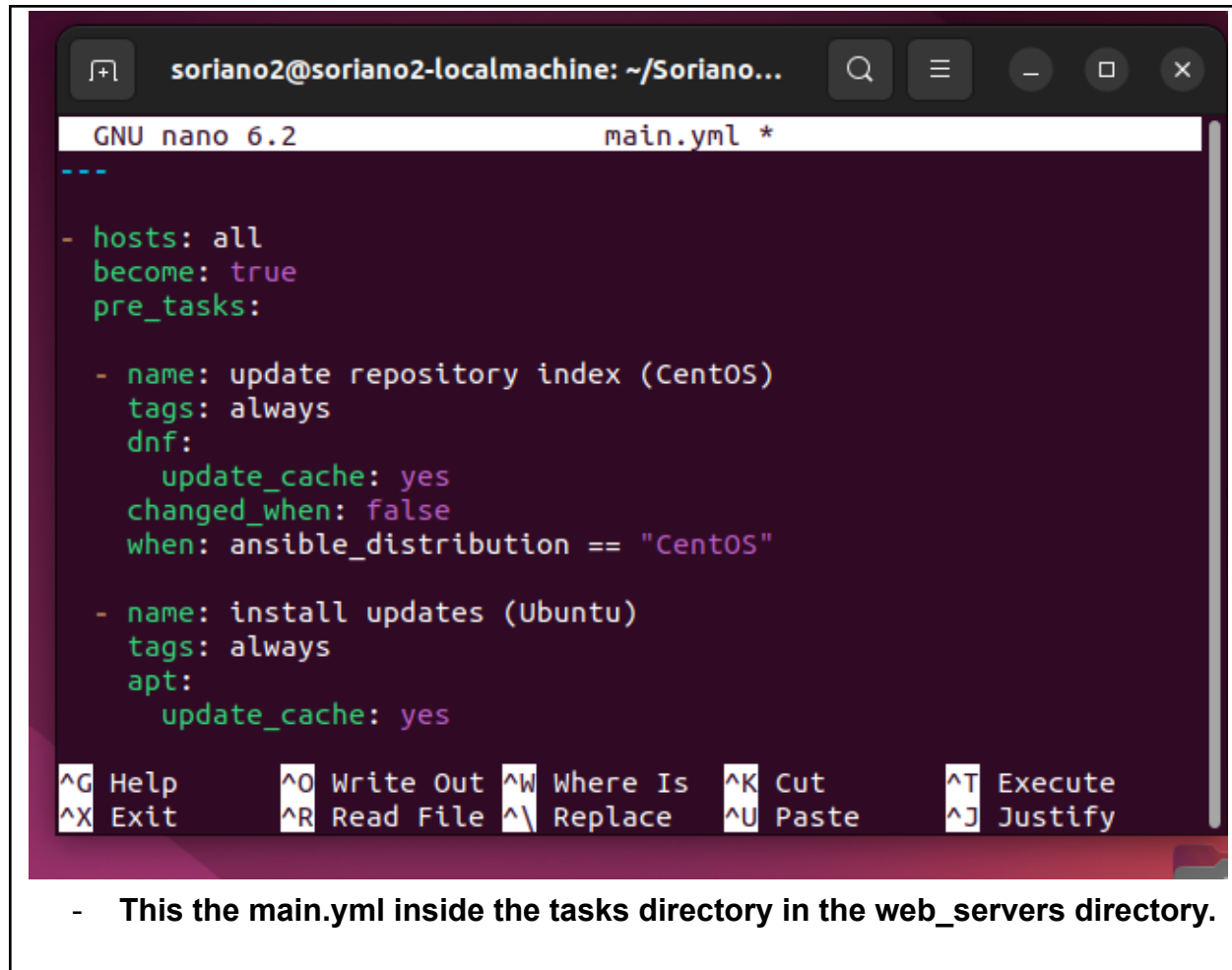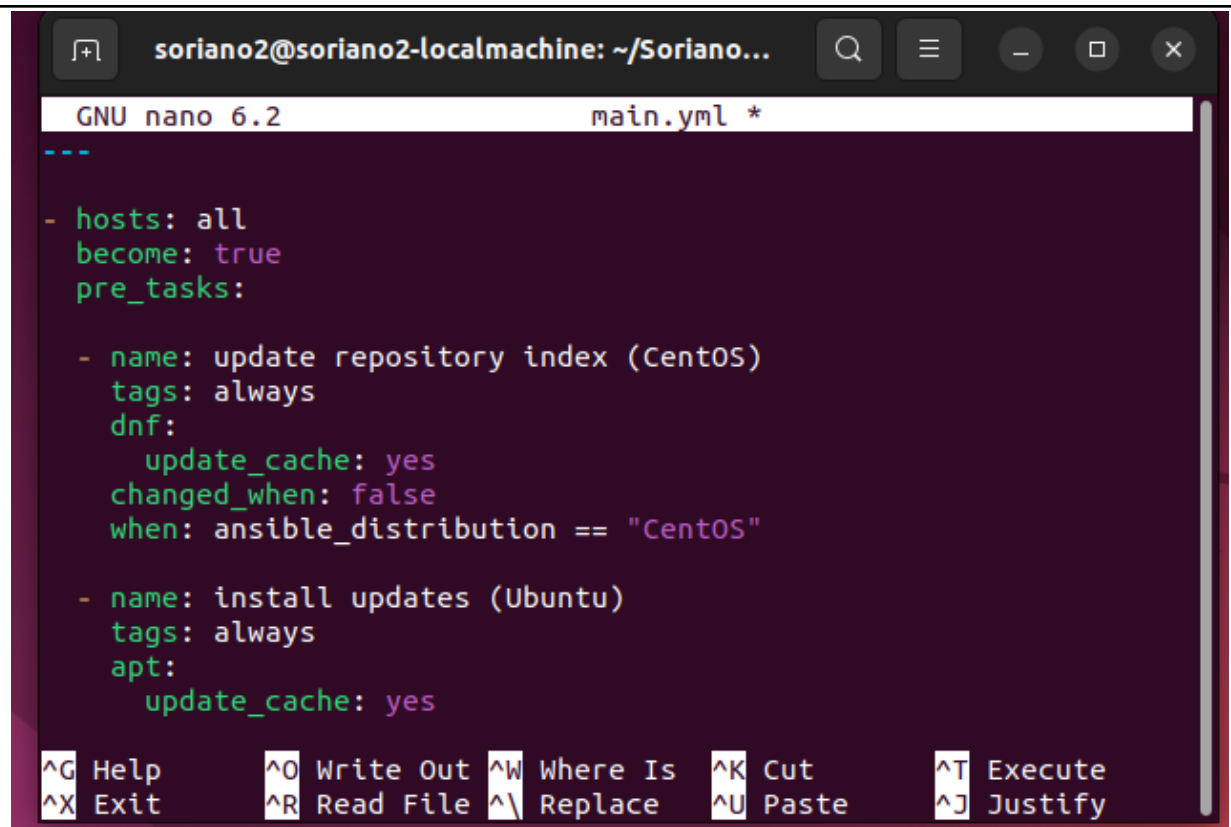
- **This the main.yml inside the tasks directory in the workstations directory.**

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

SCREENSHOTS:

```
soriano2@soriano2-localmachine: ~/Soriano...          Q    ≡    —   □   ✕

soriano2@soriano2-localmachine:~/Soriano_PrelimExam$ ls
ansible.cfg  files                inventory   README.md  site.yml
config.yaml  install_apache.yml  oldsite.yml  roles
soriano2@soriano2-localmachine:~/Soriano_PrelimExam$ ansible-playbo
ok --ask-become-pass site.yml
BECOME password:

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

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

TASK [install updates (CentOS)] **************************************
*************
skipping: [192.168.56.105]
skipping: [192.168.56.106]

TASK [install updates (Ubuntu)] **************************************
```

```
soriano2@soriano2-localmachine: ~/Soriano...          Q    ≡    —    □    ✕

skipping: [192.168.56.106]

TASK [install updates (Ubuntu)] ********************************
*************
ok: [192.168.56.106]
ok: [192.168.56.105]

PLAY [workstations_servers] ****************************************
*************

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

TASK [install unzip] ***********************************************
*************
ok: [192.168.56.106]
ok: [192.168.56.105]

TASK [install terraform] *******************************************
```

```
soriano2@soriano2-localmachine: ~/Soriano...          Q  ≡  —  □  ×

ok: [192.168.56.105]

TASK [install terraform] ***************************************
*************
ok: [192.168.56.105]
ok: [192.168.56.106]

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

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

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

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

```
ok: [192.168.56.105]

TASK [install apache and php for Ubuntu servers] ******************
*************
ok: [192.168.56.106]
ok: [192.168.56.105]

TASK [install apache and php for CentOS servers] ******************
*************
skipping: [192.168.56.105]
skipping: [192.168.56.106]

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

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

TASK [install mariadb package (CentOS)] **************************
```

ok: [192.168.56.105]

TASK [install mariadb package (CentOS)] ***************************
****
skipping: [192.168.56.105]
skipping: [192.168.56.106]

TASK [install mariadb package (Ubuntu)] ***************************
****
ok: [192.168.56.105]
ok: [192.168.56.106]

TASK [Mariadb- Restarting/Enabling] *******************************
****
changed: [192.168.56.106]
changed: [192.168.56.105]

PLAY [file_servers] ***
****

TASK [Gathering Facts] **

```
soriano2@soriano2-localmachine: ~/Soriano...          Q   ☰   —   □   ✕

TASK [Mariadb- Restarting/Enabling] ******************************
*************
changed: [192.168.56.106]
changed: [192.168.56.105]


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


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


PLAY RECAP *******************************************************
*************
192.168.56.105              : ok=12    changed=1    unreachable=0
failed=0    skipped=3    rescued=0    ignored=0
192.168.56.106              : ok=12    changed=1    unreachable=0
failed=0    skipped=3    rescued=0    ignored=0

soriano2@soriano2-localmachine:~/Soriano_PrelimExam$ █
```

- **These images show the whole result of the last running of the site.yml. So far, so good. All of the encoded tasks inside the site.yml are properly running. Also the main.yml files inside each of the tasks directories are having the old codes of the site.yml. With this being said, this activity is successfully completed. The last image shows that the files are successfully uploaded to the repository which seen above with the comment "LAST COMMIT ON ACT 7 12PM OCT 13"**

Reflections:
Answer the following:

1. What is the importance of creating roles?
   - The use of Ansible roles, which group and encapsulate related automation artifacts like configuration files, templates, tasks, and handlers, enables you to create reusable automation components. These components are simpler to reuse and share with others because roles isolate them.

2. What is the importance of managing files?

   - File organization strategies are crucial because they can keep your computer organized and make it simpler to find and access the files you require. You can free up space on your hard drive and keep your computer running smoothly by using a good file management strategy.