Name: Peñas, Issa Victoria H.	**Date Performed:** 11/16/2022
Course/Section: CPE31S22	**Date Submitted:** 11/19/2022
Instructor: Dr. Jonathan V. Taylar	**Semester and SY:** 1st Semester (2022-2023)

Activity 11: Containerization

1. Objectives

Create a Dockerfile and form a workflow using Ansible as Infrastructure as Code (IaC) to enable Continuous Delivery process

2. Discussion

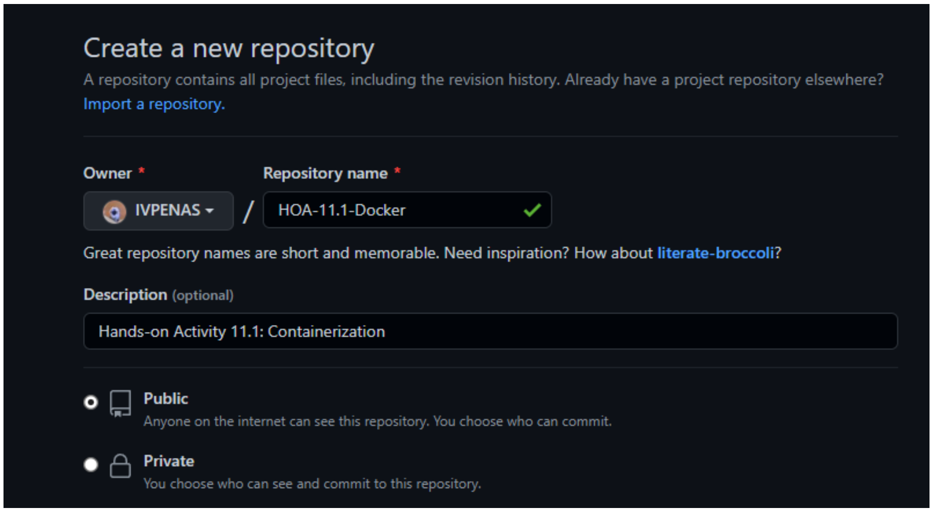
Docker is an open platform for developing, shipping, and running applications. Docker enables you to separate your applications from your infrastructure so you can deliver software quickly. With Docker, you can manage your infrastructure in the same ways you manage your applications. By taking advantage of Docker's methodologies for shipping, testing, and deploying code quickly, you can significantly reduce the delay between writing code and running it in production.
Source: https://docs.docker.com/get-started/overview/
You may also check the difference between containers and virtual machines. Click the link given below.
Source: https://docs.microsoft.com/en-us/virtualization/windowscontainers/about/containers-vs-vm

3. Tasks

1. Create a new repository for this activity.
2. Install Docker and enable the docker socket.
3. Add to Docker group to your current user.
4. Create a Dockerfile to install web and DB server.
5. Install and build the Dockerfile using Ansible.
6. Add, commit and push it to your repository.

4. **Output** (screenshots and explanations)

1. Create a new repository for this activity.



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?
Import a repository.

Owner * Repository name *

IVPENAS ▾ / HOA-11.1-Docker ✓

Great repository names are short and memorable. Need inspiration? How about literate-broccoli?

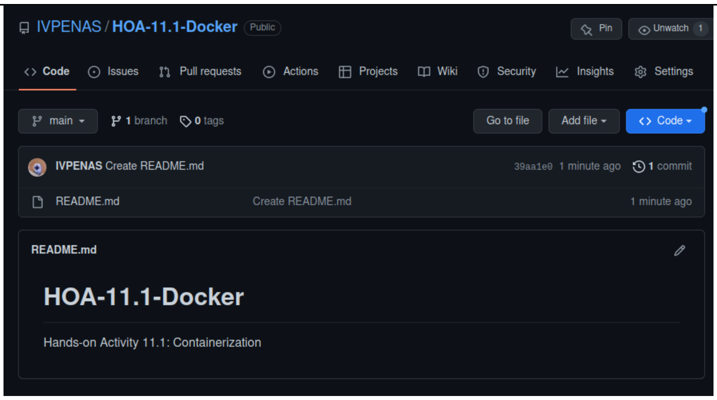
Description (optional)

Hands-on Activity 11.1: Containerization

◉ Public
Anyone on the internet can see this repository. You choose who can commit.

● Private
You choose who can see and commit to this repository.

Figure 1.1. Shows the creation of New Repository in GitHub



IVPENAS Create README.md 39aa1e0 1 minute ago 1 commit

README.md Create README.md 1 minute ago

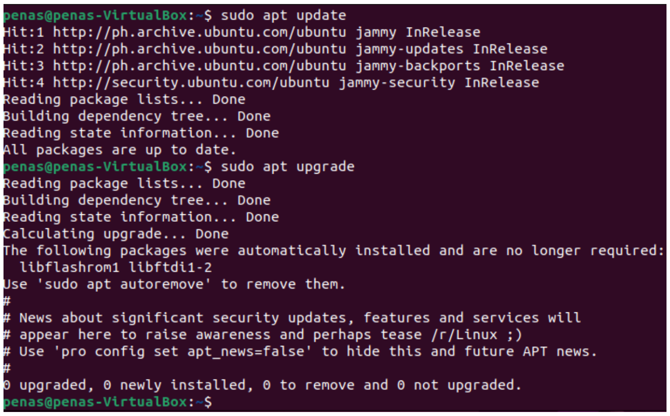
README.md

HOA-11.1-Docker

Hands-on Activity 11.1: Containerization

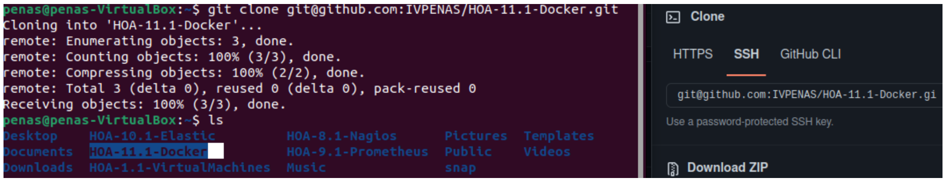
Figure 1.2. Shows the Created Repository named HOA-9.1-Prometheus

2. Install Docker and enable the docker socket.
3. Add to Docker group to your current user.
4. Create a Dockerfile to install web and DB server.
5. Install and build the Dockerfile using Ansible.



```
penas@penas-VirtualBox:~$ sudo apt update
Hit:1 http://ph.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://ph.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://ph.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu jammy-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
penas@penas-VirtualBox:~$ sudo apt upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following packages were automatically installed and are no longer required:
  libflashrom1 libftdi1-2
Use 'sudo apt autoremove' to remove them.
#
# News about significant security updates, features and services will
# appear here to raise awareness and perhaps tease /r/Linux ;)
# Use 'pro config set apt_news=false' to hide this and future APT news.
#
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
penas@penas-VirtualBox:~$
```

Figure 1.3. Shows the Local Server was up-to-date

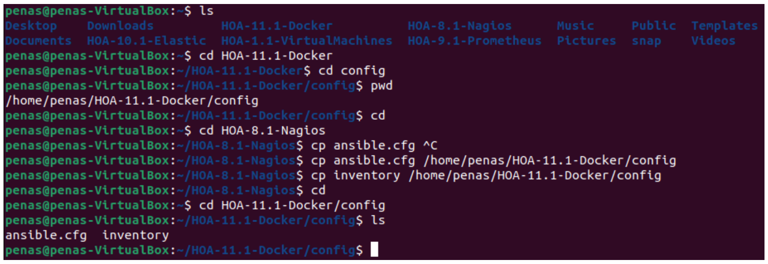


```
penas@penas-VirtualBox:~$ git clone git@github.com:IVPENAS/HOA-11.1-Docker.git
Cloning into 'HOA-11.1-Docker'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
penas@penas-VirtualBox:~$ ls
Desktop     HOA-10.1-Elastic          HOA-8.1-Nagios       Pictures  Templates
Documents   HOA-11.1-Docker           HOA-9.1-Prometheus   Public    Videos
Downloads   HOA-1.1-VirtualMachines   Music                snap
```

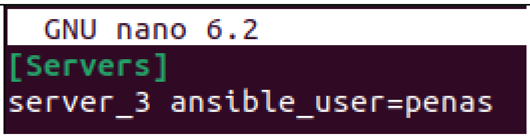
git@github.com:IVPENAS/HOA-11.1-Docker.gi

Use a password-protected SSH key.

Figure 1.4. Cloning the created Repository from GitHub to the Local Host

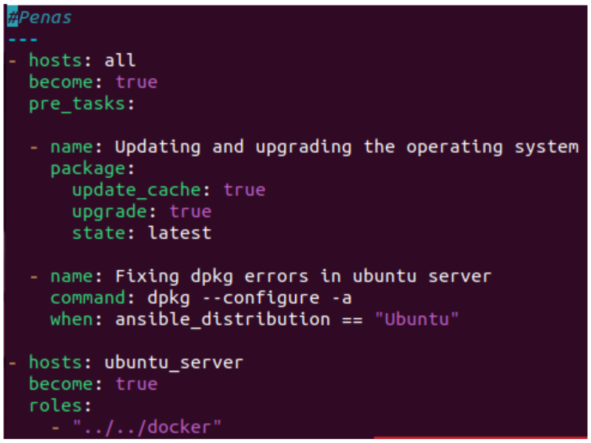


```
penas@penas-VirtualBox:~$ ls
Desktop    Downloads         HOA-11.1-Docker          HOA-8.1-Nagios      Music     Public  Templates
Documents  HOA-10.1-Elastic  HOA-1.1-VirtualMachines  HOA-9.1-Prometheus  Pictures  snap    Videos
penas@penas-VirtualBox:~$ cd HOA-11.1-Docker
penas@penas-VirtualBox:~/HOA-11.1-Docker$ cd config
penas@penas-VirtualBox:~/HOA-11.1-Docker/config$ pwd
/home/penas/HOA-11.1-Docker/config
penas@penas-VirtualBox:~/HOA-11.1-Docker/config$ cd
penas@penas-VirtualBox:~$ cd HOA-8.1-Nagios
penas@penas-VirtualBox:~/HOA-8.1-Nagios$ cp ansible.cfg ^C
penas@penas-VirtualBox:~/HOA-8.1-Nagios$ cp ansible.cfg /home/penas/HOA-11.1-Docker/config
penas@penas-VirtualBox:~/HOA-8.1-Nagios$ cp inventory /home/penas/HOA-11.1-Docker/config
penas@penas-VirtualBox:~/HOA-8.1-Nagios$ cd
penas@penas-VirtualBox:~$ cd HOA-11.1-Docker/config
penas@penas-VirtualBox:~/HOA-11.1-Docker/config$ ls
ansible.cfg  inventory
penas@penas-VirtualBox:~/HOA-11.1-Docker/config$ 
```

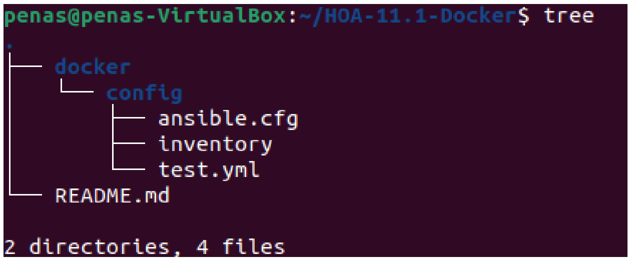


```
  GNU nano 6.2
[Servers]
server_3 ansible_user=penas
```

Figure 1.5. Copying the inventory, and ansible.cfg files

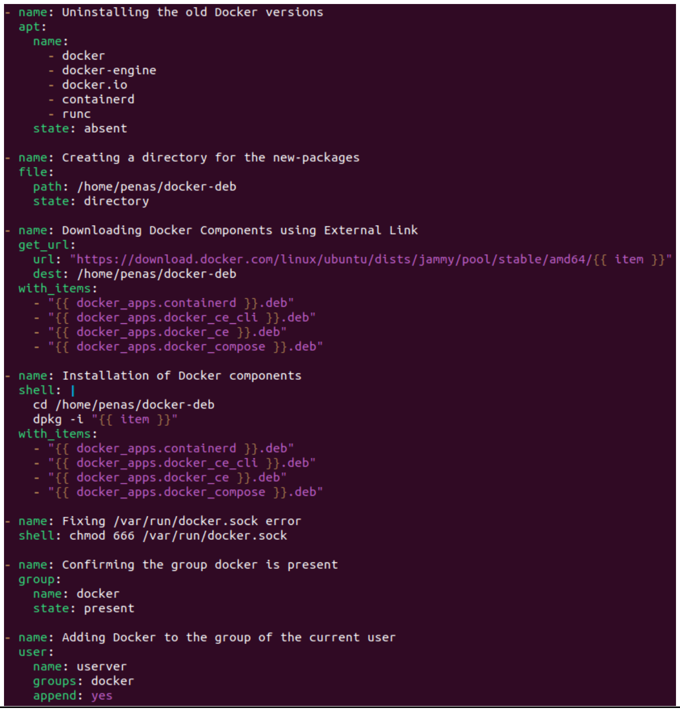


```
#Penas
---
- hosts: all
  become: true
  pre_tasks:

  - name: Updating and upgrading the operating system
    package:
      update_cache: true
      upgrade: true
      state: latest

  - name: Fixing dpkg errors in ubuntu server
    command: dpkg --configure -a
    when: ansible_distribution == "Ubuntu"

- hosts: ubuntu_server
  become: true
  roles:
    - "../../docker"
```

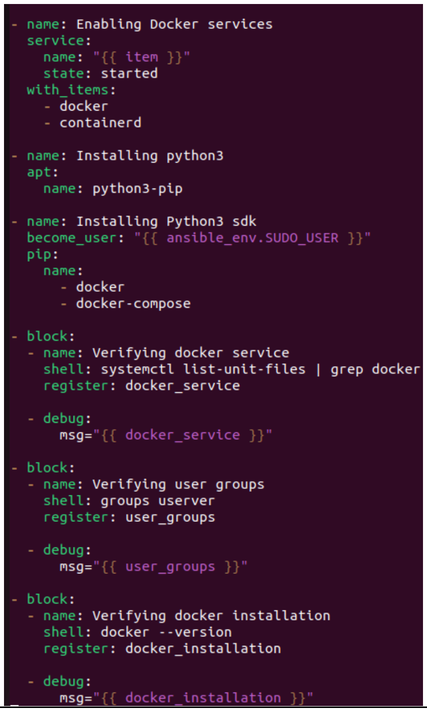


```
penas@penas-VirtualBox:~/HOA-11.1-Docker$ tree
.
├── docker
│   └── config
│       ├── ansible.cfg
│       ├── inventory
│       └── test.yml
└── README.md

2 directories, 4 files
```

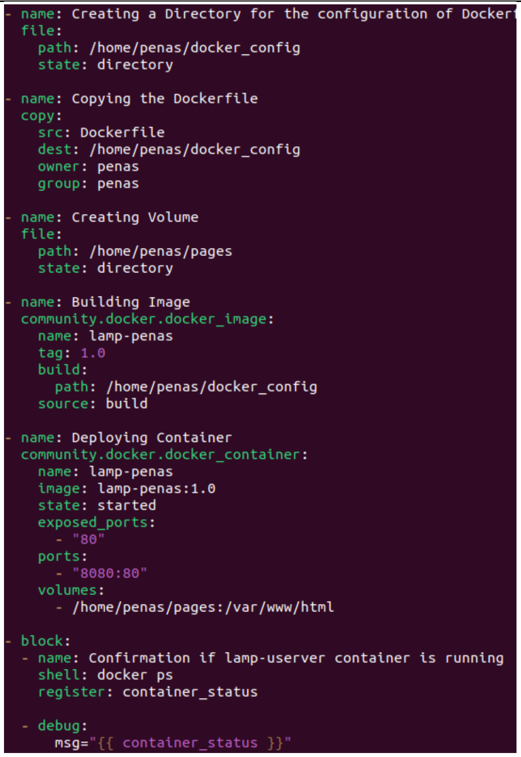
Figure 1.6-7. Creating a new .yml file for update and upgrading the OS Configuring and Dpkg in Ubuntu Servers as a partial requirement inside the docker primary folder



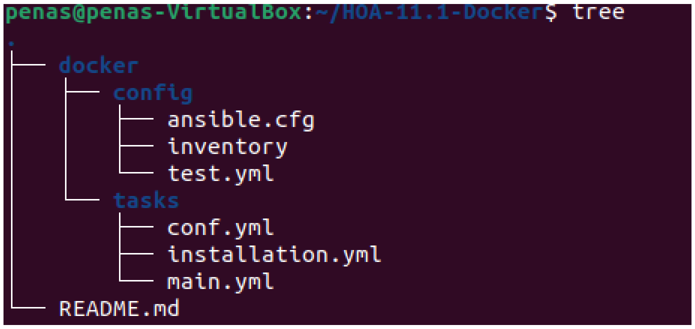
```
- name: Uninstalling the old Docker versions
  apt:
    name:
      - docker
      - docker-engine
      - docker.io
      - containerd
      - runc
    state: absent

- name: Creating a directory for the new-packages
  file:
    path: /home/penas/docker-deb
    state: directory

- name: Downloading Docker Components using External Link
  get_url:
    url: "https://download.docker.com/linux/ubuntu/dists/jammy/pool/stable/amd64/{{ item }}"
    dest: /home/penas/docker-deb
  with_items:
    - "{{ docker_apps.containerd }}.deb"
    - "{{ docker_apps.docker_ce_cli }}.deb"
    - "{{ docker_apps.docker_ce }}.deb"
    - "{{ docker_apps.docker_compose }}.deb"

- name: Installation of Docker components
  shell: |
    cd /home/penas/docker-deb
    dpkg -i "{{ item }}"
  with_items:
    - "{{ docker_apps.containerd }}.deb"
    - "{{ docker_apps.docker_ce_cli }}.deb"
    - "{{ docker_apps.docker_ce }}.deb"
    - "{{ docker_apps.docker_compose }}.deb"

- name: Fixing /var/run/docker.sock error
  shell: chmod 666 /var/run/docker.sock

- name: Confirming the group docker is present
  group:
    name: docker
    state: present

- name: Adding Docker to the group of the current user
  user:
    name: userver
    groups: docker
    append: yes
```



```
- name: Enabling Docker services
  service:
    name: "{{ item }}"
    state: started
  with_items:
    - docker
    - containerd

- name: Installing python3
  apt:
    name: python3-pip

- name: Installing Python3 sdk
  become_user: "{{ ansible_env.SUDO_USER }}"
  pip:
    name:
      - docker
      - docker-compose

- block:
  - name: Verifying docker service
    shell: systemctl list-unit-files | grep docker
    register: docker_service

  - debug:
      msg="{{ docker_service }}"

- block:
  - name: Verifying user groups
    shell: groups userver
    register: user_groups

  - debug:
      msg="{{ user_groups }}"

- block:
  - name: Verifying docker installation
    shell: docker --version
    register: docker_installation

  - debug:
      msg="{{ docker_installation }}"
```

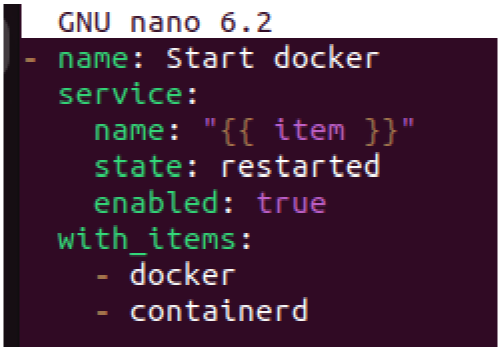


```
- name: Creating a Directory for the configuration of Docker
  file:
    path: /home/penas/docker_config
    state: directory

- name: Copying the Dockerfile
  copy:
    src: Dockerfile
    dest: /home/penas/docker_config
    owner: penas
    group: penas

- name: Creating Volume
  file:
    path: /home/penas/pages
    state: directory

- name: Building Image
  community.docker.docker_image:
    name: lamp-penas
    tag: 1.0
    build:
      path: /home/penas/docker_config
    source: build

- name: Deploying Container
  community.docker.docker_container:
    name: lamp-penas
    image: lamp-penas:1.0
    state: started
    exposed_ports:
      - "80"
    ports:
      - "8080:80"
    volumes:
      - /home/penas/pages:/var/www/html

- block:
  - name: Confirmation if lamp-userver container is running
    shell: docker ps
    register: container_status

  - debug:
      msg="{{ container_status }}"
```

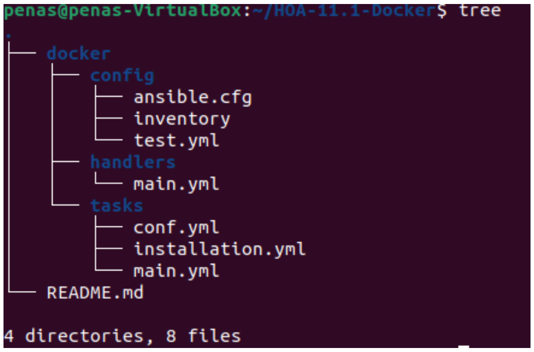


```
penas@penas-VirtualBox:~/HOA-11.1-Docker$ tree
.
├── docker
│   ├── config
│   │   ├── ansible.cfg
│   │   ├── inventory
│   │   └── test.yml
│   └── tasks
│       ├── conf.yml
│       ├── installation.yml
│       └── main.yml
└── README.md
```

Figure 1.8-11. Under the new directory the commands shown above is the Installation of the Application of Docker adding it to the current user under the configuration files of installation.yml and conf.yml

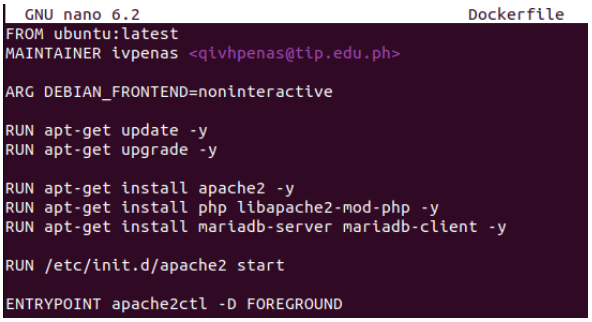


```
GNU nano 6.2
- name: Start docker
  service:
    name: "{{ item }}"
    state: restarted
    enabled: true
  with_items:
    - docker
    - containerd
```

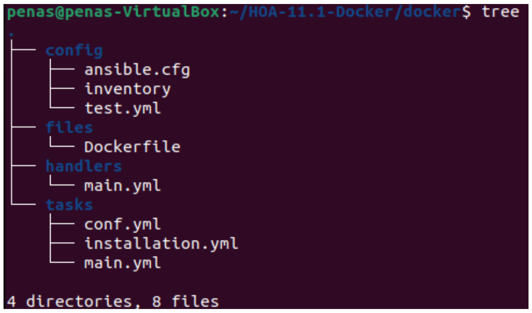


```
penas@penas-VirtualBox:~/HOA-11.1-Docker$ tree
.
├── docker
│   ├── config
│   │   ├── ansible.cfg
│   │   ├── inventory
│   │   └── test.yml
│   ├── handlers
│   │   └── main.yml
│   └── tasks
│       ├── conf.yml
│       ├── installation.yml
│       └── main.yml
└── README.md

4 directories, 8 files
```

Figure 1.12-13. Under the new directory named handlers the config file allows to run the application docker

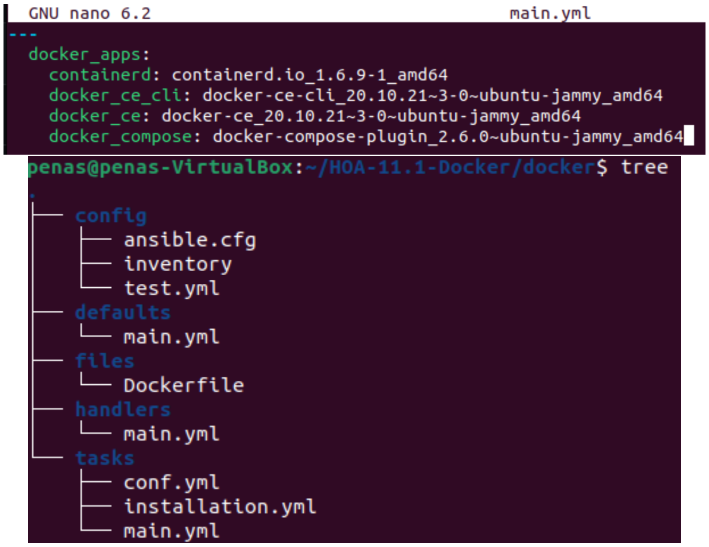


```
GNU nano 6.2                                    Dockerfile
FROM ubuntu:latest
MAINTAINER ivpenas <qivhpenas@tip.edu.ph>

ARG DEBIAN_FRONTEND=noninteractive

RUN apt-get update -y
RUN apt-get upgrade -y

RUN apt-get install apache2 -y
RUN apt-get install php libapache2-mod-php -y
RUN apt-get install mariadb-server mariadb-client -y

RUN /etc/init.d/apache2 start

ENTRYPOINT apache2ctl -D FOREGROUND
```



```
penas@penas-VirtualBox:~/HOA-11.1-Docker/docker$ tree
.
├── config
│   ├── ansible.cfg
│   ├── inventory
│   └── test.yml
├── files
│   └── Dockerfile
├── handlers
│   └── main.yml
└── tasks
    ├── conf.yml
    ├── installation.yml
    └── main.yml

4 directories, 8 files
```

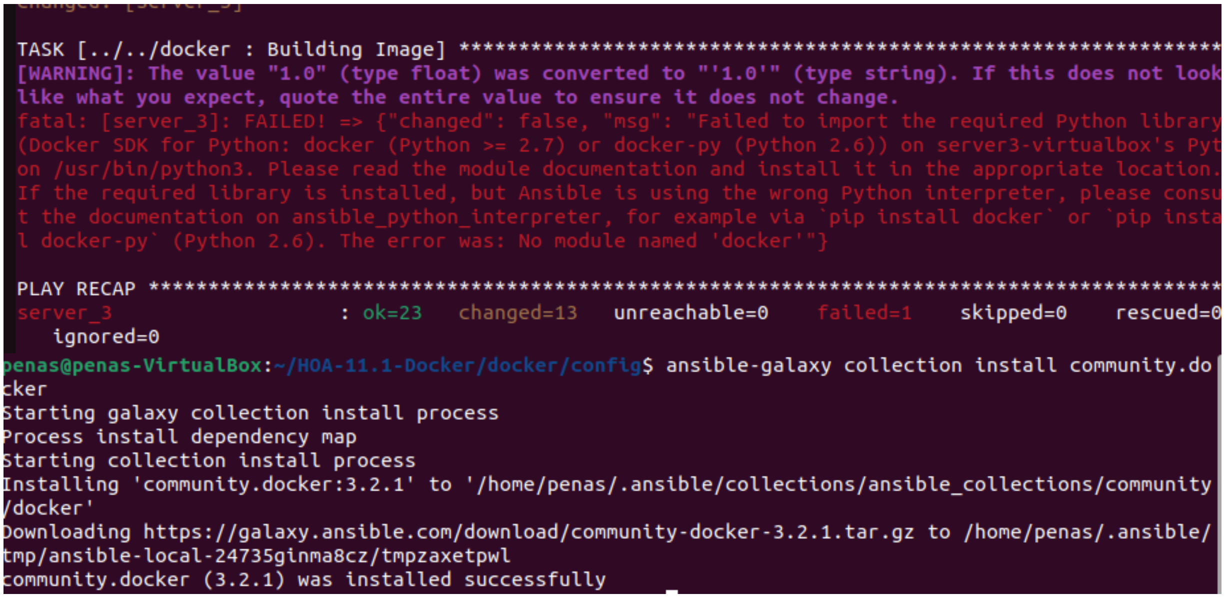
Figure 1.14-15. Under the new directory named files the config file allows to install apache and mariadb-server using the Dockerfile



```
 GNU nano 6.2                                        main.yml
---
  docker_apps:
    containerd: containerd.io_1.6.9-1_amd64
    docker_ce_cli: docker-ce-cli_20.10.21~3-0~ubuntu-jammy_amd64
    docker_ce: docker-ce_20.10.21~3-0~ubuntu-jammy_amd64
    docker_compose: docker-compose-plugin_2.6.0~ubuntu-jammy_amd64
```

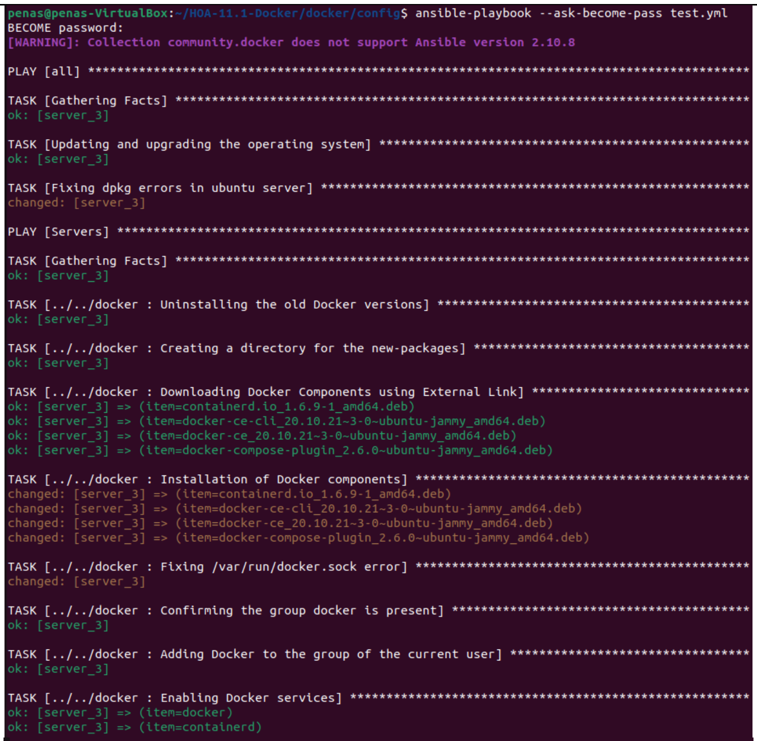
```
penas@penas-VirtualBox:~/HOA-11.1-Docker/docker$ tree
.
├── config
│   ├── ansible.cfg
│   ├── inventory
│   └── test.yml
├── defaults
│   └── main.yml
├── files
│   └── Dockerfile
├── handlers
│   └── main.yml
└── tasks
    ├── conf.yml
    ├── installation.yml
    └── main.yml
```

Figure 1.16-17. Under the new directory named defaults the config file allows to install the docker packages to the system

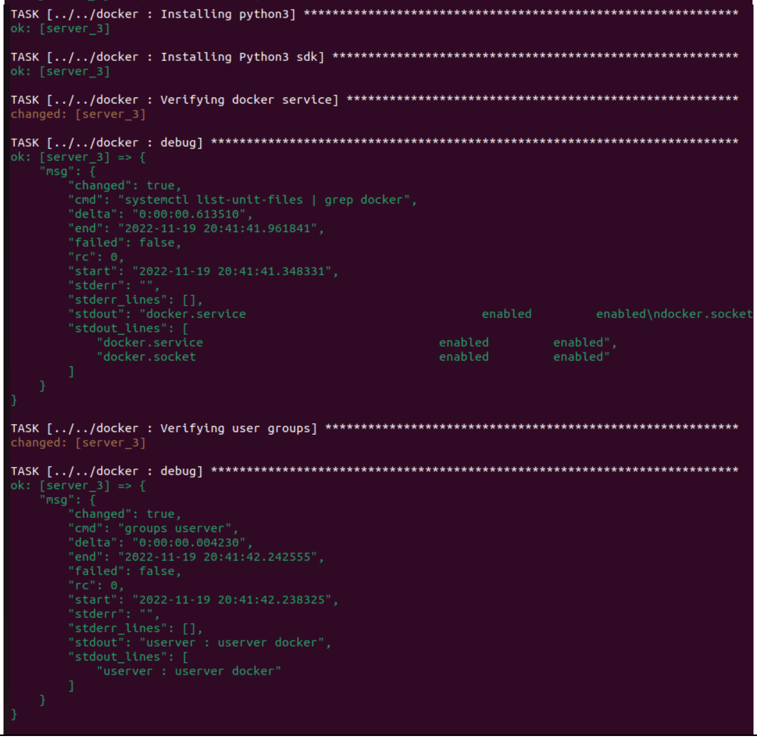


```
changed: [server_3]

TASK [../../docker : Building Image] ********************************************************************
[WARNING]: The value "1.0" (type float) was converted to "'1.0'" (type string). If this does not look
like what you expect, quote the entire value to ensure it does not change.
fatal: [server_3]: FAILED! => {"changed": false, "msg": "Failed to import the required Python library
(Docker SDK for Python: docker (Python >= 2.7) or docker-py (Python 2.6)) on server3-virtualbox's Pyt
on /usr/bin/python3. Please read the module documentation and install it in the appropriate location.
If the required library is installed, but Ansible is using the wrong Python interpreter, please consu
t the documentation on ansible_python_interpreter, for example via `pip install docker` or `pip insta
l docker-py` (Python 2.6). The error was: No module named 'docker'"}

PLAY RECAP **********************************************************************************************
server_3                   : ok=23   changed=13   unreachable=0    failed=1    skipped=0    rescued=0
   ignored=0
penas@penas-VirtualBox:~/HOA-11.1-Docker/docker/config$ ansible-galaxy collection install community.do
cker
Starting galaxy collection install process
Process install dependency map
Starting collection install process
Installing 'community.docker:3.2.1' to '/home/penas/.ansible/collections/ansible_collections/community
/docker'
Downloading https://galaxy.ansible.com/download/community-docker-3.2.1.tar.gz to /home/penas/.ansible/
tmp/ansible-local-24735ginma8cz/tmpzaxetpwl
community.docker (3.2.1) was installed successfully
```

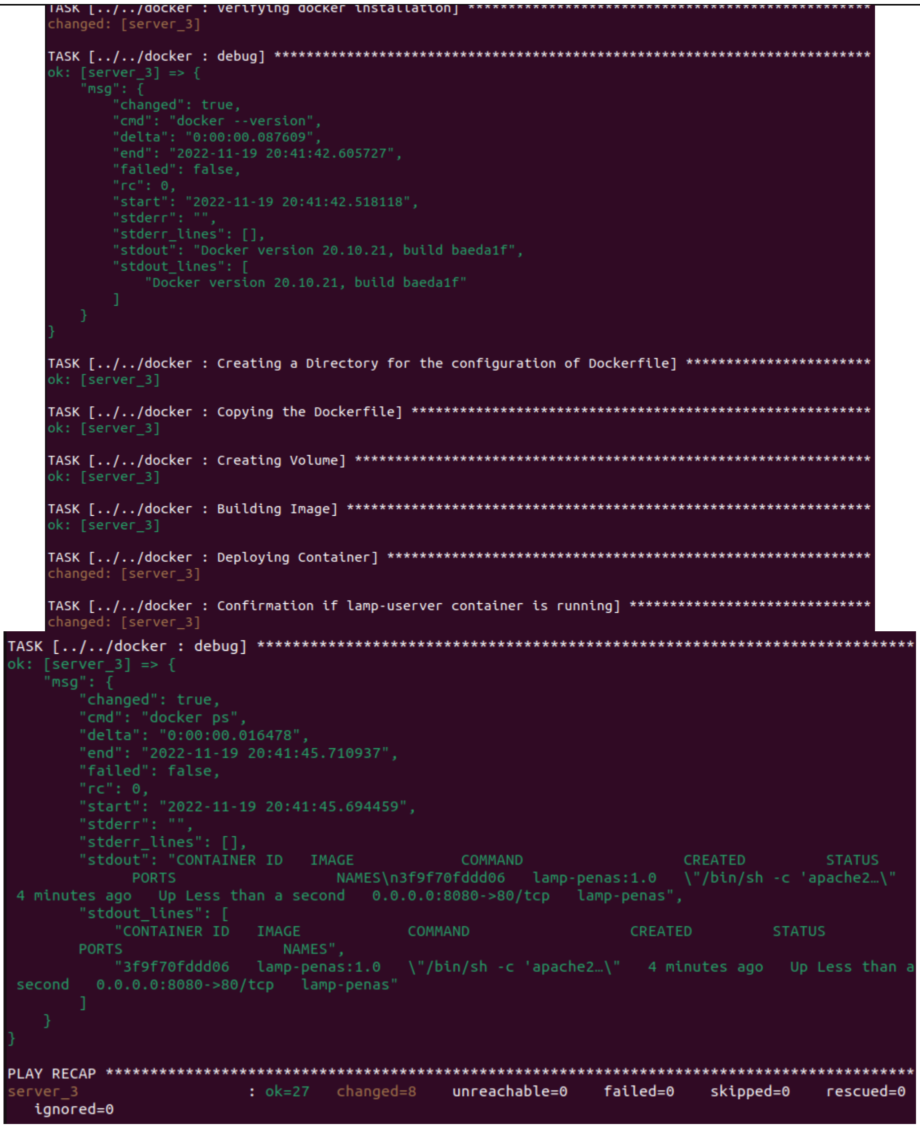
Figure 1.18-19. In order for the ansible to work with the Docker application in the server the user must input the command ‘ansible-galaxy collection install



```
penas@penas-VirtualBox:~/HOA-11.1-Docker/docker/config$ ansible-playbook --ask-become-pass test.yml
BECOME password:
[WARNING]: Collection community.docker does not support Ansible version 2.10.8

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

TASK [Gathering Facts] *******************************************************************************
ok: [server_3]

TASK [Updating and upgrading the operating system] ***************************************************
ok: [server_3]

TASK [Fixing dpkg errors in ubuntu server] ***********************************************************
changed: [server_3]

PLAY [Servers] ***************************************************************************************

TASK [Gathering Facts] *******************************************************************************
ok: [server_3]

TASK [../../docker : Uninstalling the old Docker versions] *******************************************
ok: [server_3]

TASK [../../docker : Creating a directory for the new-packages] **************************************
ok: [server_3]

TASK [../../docker : Downloading Docker Components using External Link] ******************************
ok: [server_3] => (item=containerd.io_1.6.9-1_amd64.deb)
ok: [server_3] => (item=docker-ce-cli_20.10.21~3-0~ubuntu-jammy_amd64.deb)
ok: [server_3] => (item=docker-ce_20.10.21~3-0~ubuntu-jammy_amd64.deb)
ok: [server_3] => (item=docker-compose-plugin_2.6.0~ubuntu-jammy_amd64.deb)

TASK [../../docker : Installation of Docker components] **********************************************
changed: [server_3] => (item=containerd.io_1.6.9-1_amd64.deb)
changed: [server_3] => (item=docker-ce-cli_20.10.21~3-0~ubuntu-jammy_amd64.deb)
changed: [server_3] => (item=docker-ce_20.10.21~3-0~ubuntu-jammy_amd64.deb)
changed: [server_3] => (item=docker-compose-plugin_2.6.0~ubuntu-jammy_amd64.deb)

TASK [../../docker : Fixing /var/run/docker.sock error] **********************************************
changed: [server_3]

TASK [../../docker : Confirming the group docker is present] *****************************************
ok: [server_3]

TASK [../../docker : Adding Docker to the group of the current user] *********************************
ok: [server_3]

TASK [../../docker : Enabling Docker services] *******************************************************
ok: [server_3] => (item=docker)
ok: [server_3] => (item=containerd)
```

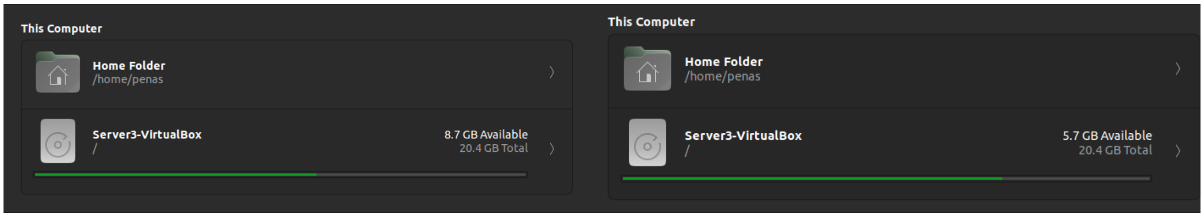


```
TASK [../../docker : Installing python3] ***********************************************************
ok: [server_3]

TASK [../../docker : Installing Python3 sdk] *******************************************************
ok: [server_3]

TASK [../../docker : Verifying docker service] *****************************************************
changed: [server_3]

TASK [../../docker : debug] ************************************************************************
ok: [server_3] => {
    "msg": {
        "changed": true,
        "cmd": "systemctl list-unit-files | grep docker",
        "delta": "0:00:00.613510",
        "end": "2022-11-19 20:41:41.961841",
        "failed": false,
        "rc": 0,
        "start": "2022-11-19 20:41:41.348331",
        "stderr": "",
        "stderr_lines": [],
        "stdout": "docker.service                               enabled         enabled\ndocker.socket
        "stdout_lines": [
            "docker.service                               enabled         enabled",
            "docker.socket                                enabled         enabled"
        ]
    }
}

TASK [../../docker : Verifying user groups] ********************************************************
changed: [server_3]

TASK [../../docker : debug] ************************************************************************
ok: [server_3] => {
    "msg": {
        "changed": true,
        "cmd": "groups userver",
        "delta": "0:00:00.004230",
        "end": "2022-11-19 20:41:42.242555",
        "failed": false,
        "rc": 0,
        "start": "2022-11-19 20:41:42.238325",
        "stderr": "",
        "stderr_lines": [],
        "stdout": "userver : userver docker",
        "stdout_lines": [
            "userver : userver docker"
        ]
    }
}
```



```
TASK [../../docker : Verifying docker installation] ****************************************************
changed: [server_3]

TASK [../../docker : debug] *****************************************************************************
ok: [server_3] => {
    "msg": {
        "changed": true,
        "cmd": "docker --version",
        "delta": "0:00:00.087609",
        "end": "2022-11-19 20:41:42.605727",
        "failed": false,
        "rc": 0,
        "start": "2022-11-19 20:41:42.518118",
        "stderr": "",
        "stderr_lines": [],
        "stdout": "Docker version 20.10.21, build baeda1f",
        "stdout_lines": [
            "Docker version 20.10.21, build baeda1f"
        ]
    }
}

TASK [../../docker : Creating a Directory for the configuration of Dockerfile] **********************
ok: [server_3]

TASK [../../docker : Copying the Dockerfile] ***********************************************************
ok: [server_3]

TASK [../../docker : Creating Volume] ******************************************************************
ok: [server_3]

TASK [../../docker : Building Image] *******************************************************************
ok: [server_3]

TASK [../../docker : Deploying Container] **************************************************************
changed: [server_3]

TASK [../../docker : Confirmation if lamp-userver container is running] ******************************
changed: [server_3]
TASK [../../docker : debug] ******************************************************************************
ok: [server_3] => {
    "msg": {
        "changed": true,
        "cmd": "docker ps",
        "delta": "0:00:00.016478",
        "end": "2022-11-19 20:41:45.710937",
        "failed": false,
        "rc": 0,
        "start": "2022-11-19 20:41:45.694459",
        "stderr": "",
        "stderr_lines": [],
        "stdout": "CONTAINER ID   IMAGE            COMMAND                  CREATED         STATUS
           PORTS                  NAMES\n3f9f70fddd06   lamp-penas:1.0   \"/bin/sh -c 'apache2…\"
 4 minutes ago   Up Less than a second   0.0.0.0:8080->80/tcp   lamp-penas",
        "stdout_lines": [
            "CONTAINER ID   IMAGE            COMMAND                  CREATED         STATUS
        PORTS                  NAMES",
            "3f9f70fddd06   lamp-penas:1.0   \"/bin/sh -c 'apache2…\"   4 minutes ago   Up Less than a
 second   0.0.0.0:8080->80/tcp   lamp-penas"
        ]
    }
}

PLAY RECAP ***********************************************************************************************
server_3                   : ok=27   changed=8    unreachable=0    failed=0    skipped=0    rescued=0
   ignored=0
```

Figure 1.20-24. Shows the Play Recap of the compiled Ansible Playbook after running the task.yml



This Computer

Home Folder
/home/penas

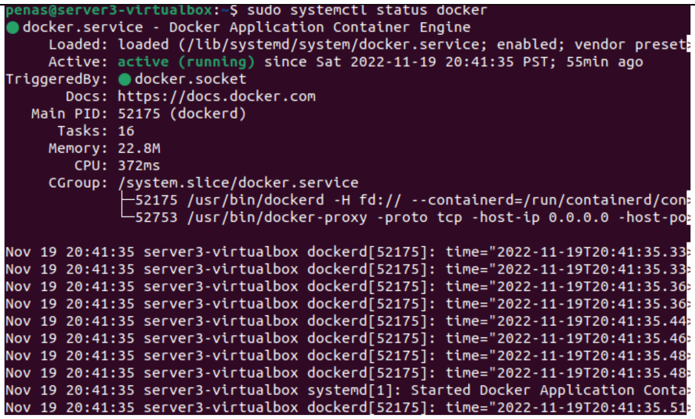
Server3-VirtualBox
/
8.7 GB Available
20.4 GB Total

This Computer

Home Folder
/home/penas

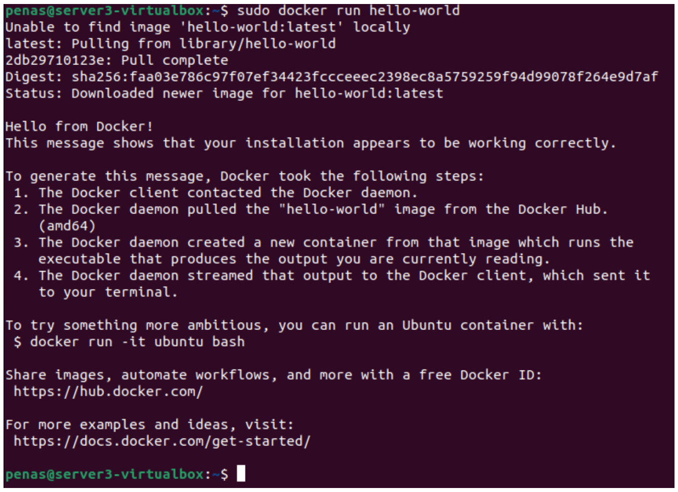
Server3-VirtualBox
/
5.7 GB Available
20.4 GB Total

Figure 1.25-26. Shows the Disk Storage of the Server 3 before and after installing the Docker Application and Updating/Upgrading the OS



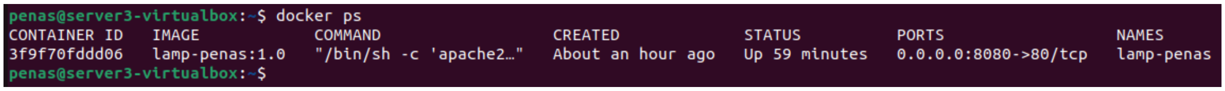
```
penas@server3-virtualbox:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
     Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset>
     Active: active (running) since Sat 2022-11-19 20:41:35 PST; 55min ago
TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
   Main PID: 52175 (dockerd)
      Tasks: 16
     Memory: 22.8M
        CPU: 372ms
     CGroup: /system.slice/docker.service
             ├─52175 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/con>
             └─52753 /usr/bin/docker-proxy -proto tcp -host-ip 0.0.0.0 -host-po>

Nov 19 20:41:35 server3-virtualbox dockerd[52175]: time="2022-11-19T20:41:35.33>
Nov 19 20:41:35 server3-virtualbox dockerd[52175]: time="2022-11-19T20:41:35.33>
Nov 19 20:41:35 server3-virtualbox dockerd[52175]: time="2022-11-19T20:41:35.36>
Nov 19 20:41:35 server3-virtualbox dockerd[52175]: time="2022-11-19T20:41:35.36>
Nov 19 20:41:35 server3-virtualbox dockerd[52175]: time="2022-11-19T20:41:35.44>
Nov 19 20:41:35 server3-virtualbox dockerd[52175]: time="2022-11-19T20:41:35.46>
Nov 19 20:41:35 server3-virtualbox dockerd[52175]: time="2022-11-19T20:41:35.48>
Nov 19 20:41:35 server3-virtualbox dockerd[52175]: time="2022-11-19T20:41:35.48>
Nov 19 20:41:35 server3-virtualbox systemd[1]: Started Docker Application Conta>
Nov 19 20:41:35 server3-virtualbox dockerd[52175]: time="2022-11-19T20:41:35.51>
```

Figure 1.27. To Further verify the installation of the Docker Application, use the command 'sudo systemctl status docker' which shows a status of Active meaning the Application is currently running



```
penas@server3-virtualbox:~$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
2db29710123e: Pull complete
Digest: sha256:faa03e786c97f07ef34423fccceeec2398ec8a5759259f94d99078f264e9d7af
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/

penas@server3-virtualbox:~$ 
```

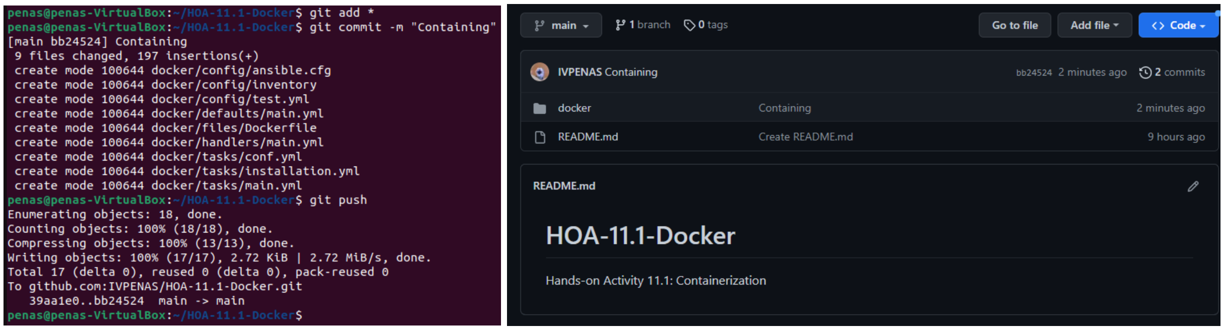
Figure 1.28. Another way to verify the installation was to execute the command 'sudo docker run hello-world' which pulls the image if the said image was not installed yet within the system



```
penas@server3-virtualbox:~$ docker ps
CONTAINER ID   IMAGE            COMMAND                  CREATED          STATUS          PORTS                  NAMES
3f9f70fddd06   lamp-penas:1.0   "/bin/sh -c 'apache2…"   About an hour ago   Up 59 minutes   0.0.0.0:8080->80/tcp   lamp-penas
penas@server3-virtualbox:~$
```

Figure 1.29. Remember the installation.yml codes where the student initiated the creation of an image namely lamp-penas:1.0 which was currently running and containerized within the system

6. Add, commit and push it to your repository.



```
penas@penas-VirtualBox:~/HOA-11.1-Docker$ git add *
penas@penas-VirtualBox:~/HOA-11.1-Docker$ git commit -m "Containing"
[main bb24524] Containing
 9 files changed, 197 insertions(+)
 create mode 100644 docker/config/ansible.cfg
 create mode 100644 docker/config/inventory
 create mode 100644 docker/config/test.yml
 create mode 100644 docker/defaults/main.yml
 create mode 100644 docker/files/Dockerfile
 create mode 100644 docker/handlers/main.yml
 create mode 100644 docker/tasks/conf.yml
 create mode 100644 docker/tasks/installation.yml
 create mode 100644 docker/tasks/main.yml
penas@penas-VirtualBox:~/HOA-11.1-Docker$ git push
Enumerating objects: 18, done.
Counting objects: 100% (18/18), done.
Compressing objects: 100% (13/13), done.
Writing objects: 100% (17/17), 2.72 KiB | 2.72 MiB/s, done.
Total 17 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:IVPENAS/HOA-11.1-Docker.git
   39aa1e0..bb24524  main -> main
penas@penas-VirtualBox:~/HOA-11.1-Docker$
```

IVPENAS Containing bb24524 2 minutes ago 2 commits

docker	Containing	2 minutes ago
README.md	Create README.md	9 hours ago

README.md

HOA-11.1-Docker

Hands-on Activity 11.1: Containerization

Figure 1.30. Pushing the changes to the connected repository

Reflections:

Answer the following:

1. What are the benefits of implementing containerizations?

   Before we get through to its benefits, first we describe the concept of containerization which is a Form of Operating System Virtualization or can be called a 'Portable Computing Environment' that allows Developers to deploy and create applications securely as the contained application is isolated user space-sharing in an OS and to transfer its data from one environment to another without having any issues, here are some of the benefits of implementing containerizations:

   [1] **Portability**, whenever an application container creates an executable software package, it'll be abstracted away from the host operating system, meaning the container does not depend to the host OS as a result, it allows the developer to run the contained application in any platform or in the cloud without hassle as long the OS supports the containerization tools. [2] **Efficiency**, as application is containerized it allows a developer to transfer among various application layer, as containers have smaller capacity compared to virtual machines resulting minimal startup times and the ability to run multiple containers in the same computer capacity in one virtual machine making it efficient on companies as it reduces licensing cost and associated servers. [3] **Agility**, as containers have minimal capacities, it can incorporate its creation rapidly and be deployed in any environment then the admins can execute the *orchestration* or the ability to shut down the container and use it again until it was needed. [4] **Improved Security**, on any occasion that a malicious code executes to the system the isolated or containerized application prevents its devastating effects to spread to the system, as application is running on their respective self-contained environment. And [5] **Scalability**, along with the Efficiency and Agility that containers provide it can handle increasing workload by reconfiguring the existing architecture to enable the resources using service-oriented application design with minimum resource usage.

Conclusions:

First we describe the concept of containerization which is a Form of Operating System Virtualization or can be called a 'Portable Computing Environment' due to its ability to *execute applications including their dependencies in an isolated user space-sharing in the operating system* which is the containers, whereas the Virtual Machines (VM's) has the same concept yet it mostly relies on the virtualized OS and the hypervisor software layer, while Containerizations allows an application to have a *direct access to the computing resources without using extra software layers* as they are operated on an abstracted layer beyond the host operating system meaning, it uses a more efficient method of virtualization compared to the Virtual Machines as the result of Containers doesn't use the virtual hardware, kernel, or operating system in which consumes the resources in order to run the applications.

It allows Software Developers, System and Computer Network Administrator to *securely create and deploy applications* whenever it was being transferred to a new location *without encountering on errors* due to the changes of the environment of the application or codes, as Containerization bundles the supplication code including to its configuration files, dependencies and libraries. This concept of the evolution of cloud computing also benefits the Admins and Developers in other ways namely as:

[1] **Portability**, allows the developer to run the contained application in any environment or in the cloud without hassle as long the OS supports the containerization tools. [2] **Efficiency**, it allows a developer to transfer among various application layer, as containers have smaller capacity resulting for a minimal startup time and the ability to run multiple containers. [3] **Agility**, incorporates the creation of containers rapidly and be deployed in any environment then the admins can execute the *orchestration* or the ability to shut down the container and use it again until it was needed. [4] **Improved Security**, prevents the malicious code to spread to the system, as application is running on their respective self-contained environment. And [5] **Scalability**, containerization can handle increased workload by reconfiguring the existing architecture to enable the resources using service-oriented application design with minimum resource usage.