Name: Torrecampo, Juan Piolo S.	**Date Performed:** Nov 15, 2022
Course/Section: CPE 232 / CPE31S22	**Date Submitted:** Nov 18, 2022
Instructor: Dr. Jonathan Taylar	**Semester and SY:** 1st Sem, 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.

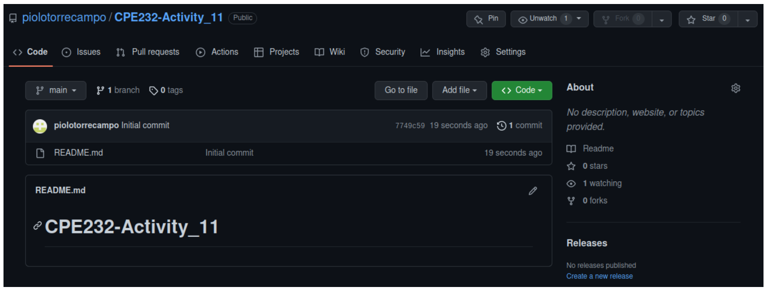
Source: https://docs.microsoft.com/en-us/virtualization/windowscontainers/about/containers-vs-vm

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

Instead of manually installing the docker in the server and configuring it for web server (LAMP Server) through Docker containerization, I have created a fully functional ansible playbook that installs docker and its dependencies, builds an image through Dockerfile and deploys that image.



piolotorrecampo Initial commit 7749c59 19 seconds ago 1 commit

README.md Initial commit 19 seconds ago

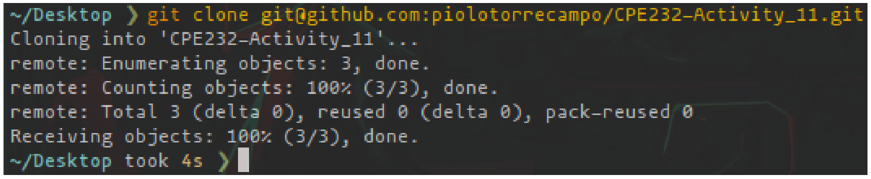
README.md

CPE232-Activity_11

No description, website, or topics provided.

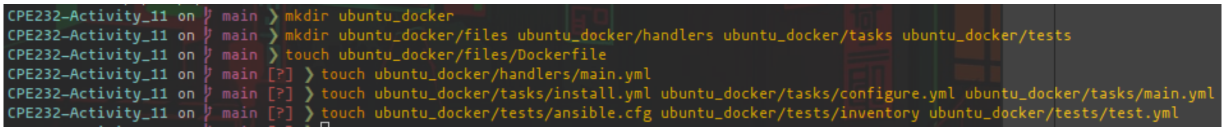
No releases published

Figure 1. The screenshot above shows the page of the newly created repository for this activity.



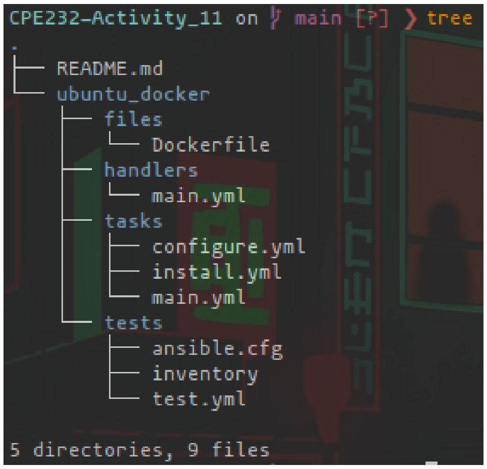
```
~/Desktop ❯ git clone git@github.com:piolotorrecampo/CPE232-Activity_11.git
Cloning into 'CPE232-Activity_11'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
~/Desktop took 4s ❯
```

Figure 2. The image above shows the cloning of the repository in the local machine.



```
CPE232-Activity_11 on  main ❯ mkdir ubuntu_docker
CPE232-Activity_11 on  main ❯ mkdir ubuntu_docker/files ubuntu_docker/handlers ubuntu_docker/tasks ubuntu_docker/tests
CPE232-Activity_11 on  main ❯ touch ubuntu_docker/files/Dockerfile
CPE232-Activity_11 on  main [?] ❯ touch ubuntu_docker/handlers/main.yml
CPE232-Activity_11 on  main [?] ❯ touch ubuntu_docker/tasks/install.yml ubuntu_docker/tasks/configure.yml ubuntu_docker/tasks/main.yml
CPE232-Activity_11 on  main [?] ❯ touch ubuntu_docker/tests/ansible.cfg ubuntu_docker/tests/inventory ubuntu_docker/tests/test.yml
```

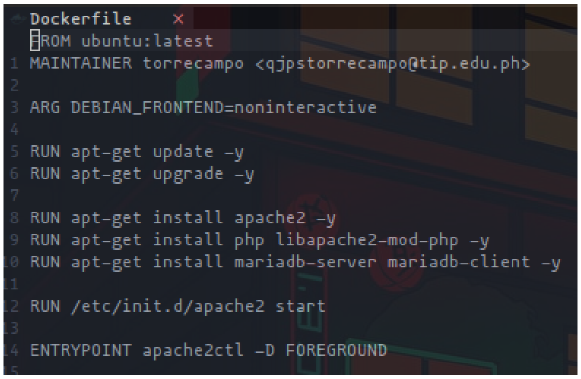
Figure 3. The sequence of commands above creates the structure of our ansible role to create a web server using docker.



```
CPE232-Activity_11 on  main [?] ❯ tree
.
├── README.md
└── ubuntu_docker
    ├── files
    │   └── Dockerfile
    ├── handlers
    │   └── main.yml
    ├── tasks
    │   ├── configure.yml
    │   ├── install.yml
    │   └── main.yml
    └── tests
        ├── ansible.cfg
        ├── inventory
        └── test.yml

5 directories, 9 files
```

Figure 4. The picture above shows the tree structure of the web server ansible role directory where Ubuntu is used as a host operating system.

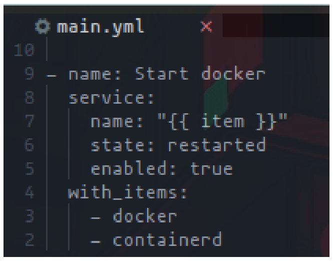
<table>
<tr><th colspan="2">File and its Contents under “files” Directory</th></tr>
<tr><td><b>Dockerfile</b></td><td>



```
Dockerfile ×
FROM ubuntu:latest
MAINTAINER torrecampo <qjpstorrecampo@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
```

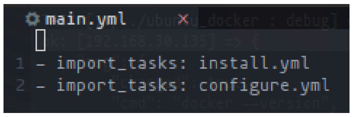
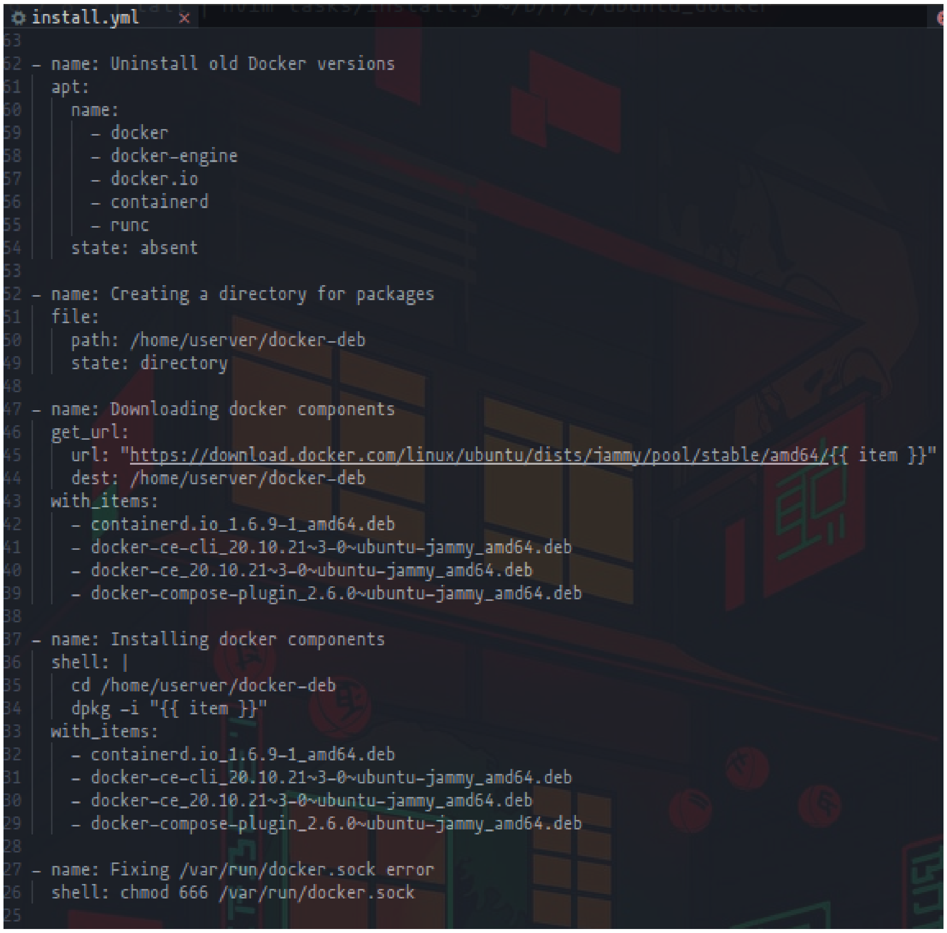
</td></tr>
</table>

Table 1. The table above shows the contents of Dockerfile in the “files” directory.

File and its Contents under ”handlers” Directory	
main.yml	

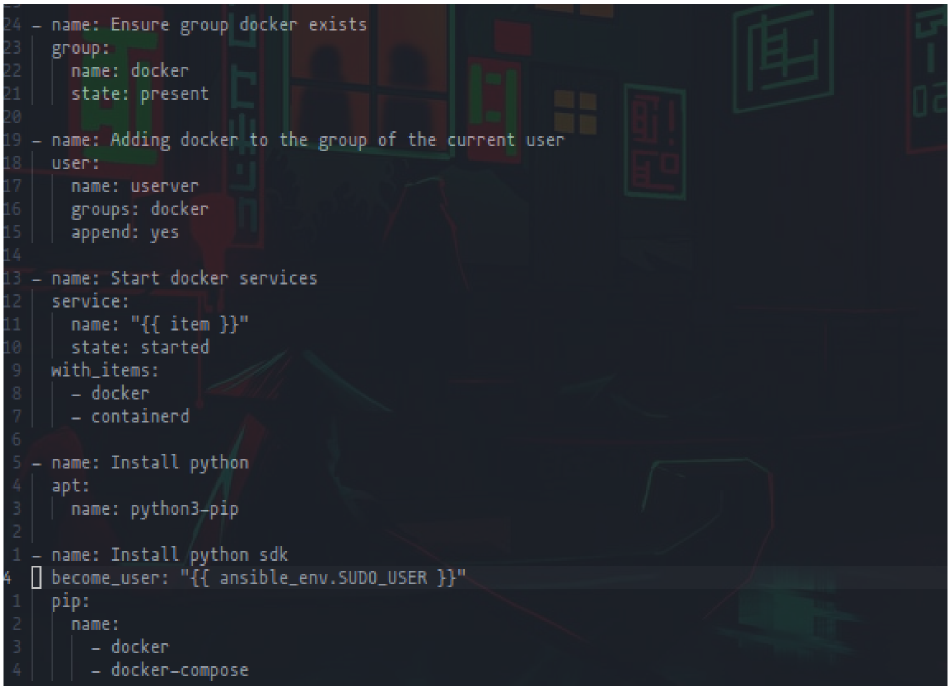
```yaml
- name: Start docker
  service:
    name: "{{ item }}"
    state: restarted
    enabled: true
  with_items:
    - docker
    - containerd
```

Table 2. The table above shows the contents of main.yml in the “handlers” directory.

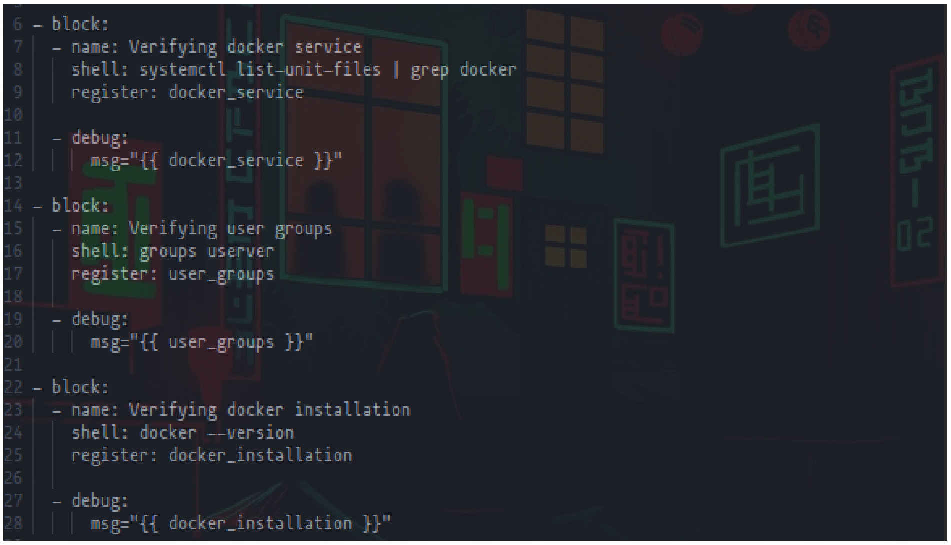
Files and its Contents under “tasks” Directory	
main.yml	
install.yml	

```yaml
- import_tasks: install.yml
- import_tasks: configure.yml
```

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

- name: Creating a directory for packages
  file:
    path: /home/userver/docker-deb
    state: directory

- name: Downloading docker components
  get_url:
    url: "https://download.docker.com/linux/ubuntu/dists/jammy/pool/stable/amd64/{{ item }}"
    dest: /home/userver/docker-deb
  with_items:
    - containerd.io_1.6.9-1_amd64.deb
    - docker-ce-cli_20.10.21~3-0~ubuntu-jammy_amd64.deb
    - docker-ce_20.10.21~3-0~ubuntu-jammy_amd64.deb
    - docker-compose-plugin_2.6.0~ubuntu-jammy_amd64.deb

- name: Installing docker components
  shell: |
    cd /home/userver/docker-deb
    dpkg -i "{{ item }}"
  with_items:
    - containerd.io_1.6.9-1_amd64.deb
    - docker-ce-cli_20.10.21~3-0~ubuntu-jammy_amd64.deb
    - docker-ce_20.10.21~3-0~ubuntu-jammy_amd64.deb
    - docker-compose-plugin_2.6.0~ubuntu-jammy_amd64.deb

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

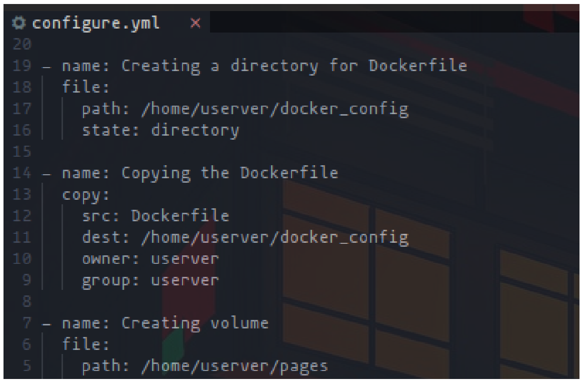


```yaml
- name: Ensure group docker exists
  group:
    name: docker
    state: present

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

- name: Start docker services
  service:
    name: "{{ item }}"
    state: started
  with_items:
    - docker
    - containerd

- name: Install python
  apt:
    name: python3-pip

- name: Install python sdk
  become_user: "{{ ansible_env.SUDO_USER }}"
  pip:
    name:
      - docker
      - docker-compose
```

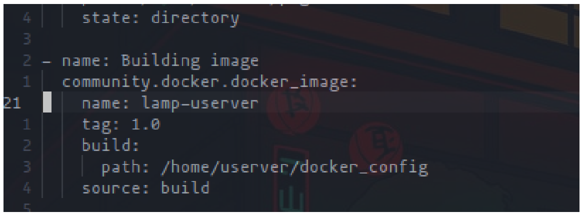


```yaml
- block:
  - name: Verifying docker service
    shell: systemctl list-unit-files | grep docker
    register: docker_service

  - 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 }}"
```

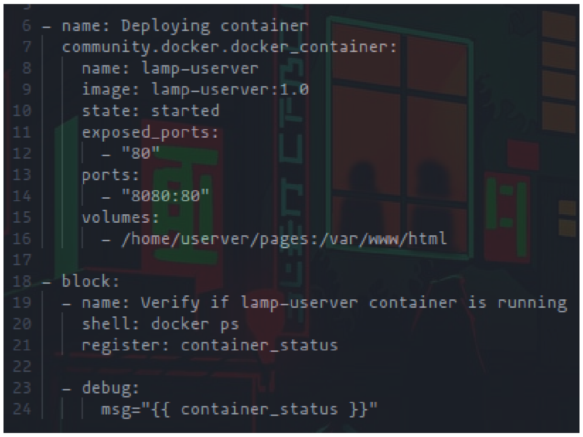
configure.yml



```yaml
- name: Creating a directory for Dockerfile
  file:
    path: /home/userver/docker_config
    state: directory

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

- name: Creating volume
  file:
    path: /home/userver/pages
```



```
    state: directory

- name: Building image
  community.docker.docker_image:
    name: lamp-userver
    tag: 1.0
    build:
      path: /home/userver/docker_config
    source: build
```

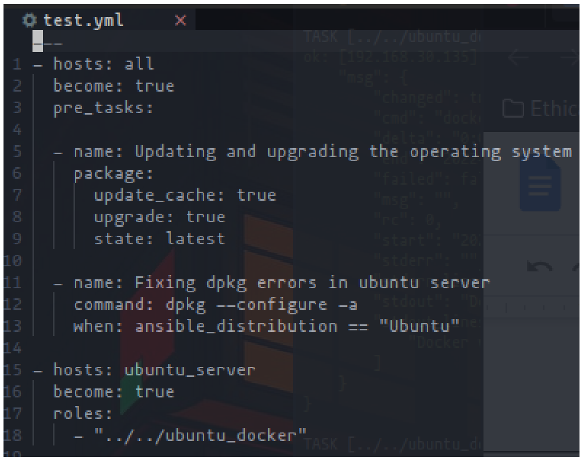


```
- name: Deploying container
  community.docker.docker_container:
    name: lamp-userver
    image: lamp-userver:1.0
    state: started
    exposed_ports:
      - "80"
    ports:
      - "8080:80"
    volumes:
      - /home/userver/pages:/var/www/html

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

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

Table 3. The table above shows the contents of each file in the "tasks" directory namely main.yml install.yml and configure.yml.

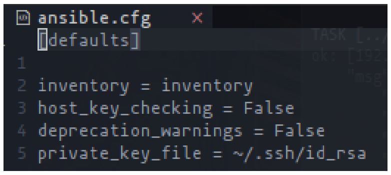
Files and its Contents under "tests" Directory

test.yml



```
---
- 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:
    - "../../ubuntu_docker"
```

ansible.cfg



```
[defaults]

inventory = inventory
host_key_checking = False
deprecation_warnings = False
private_key_file = ~/.ssh/id_rsa
```

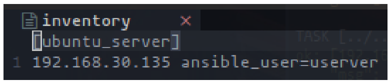
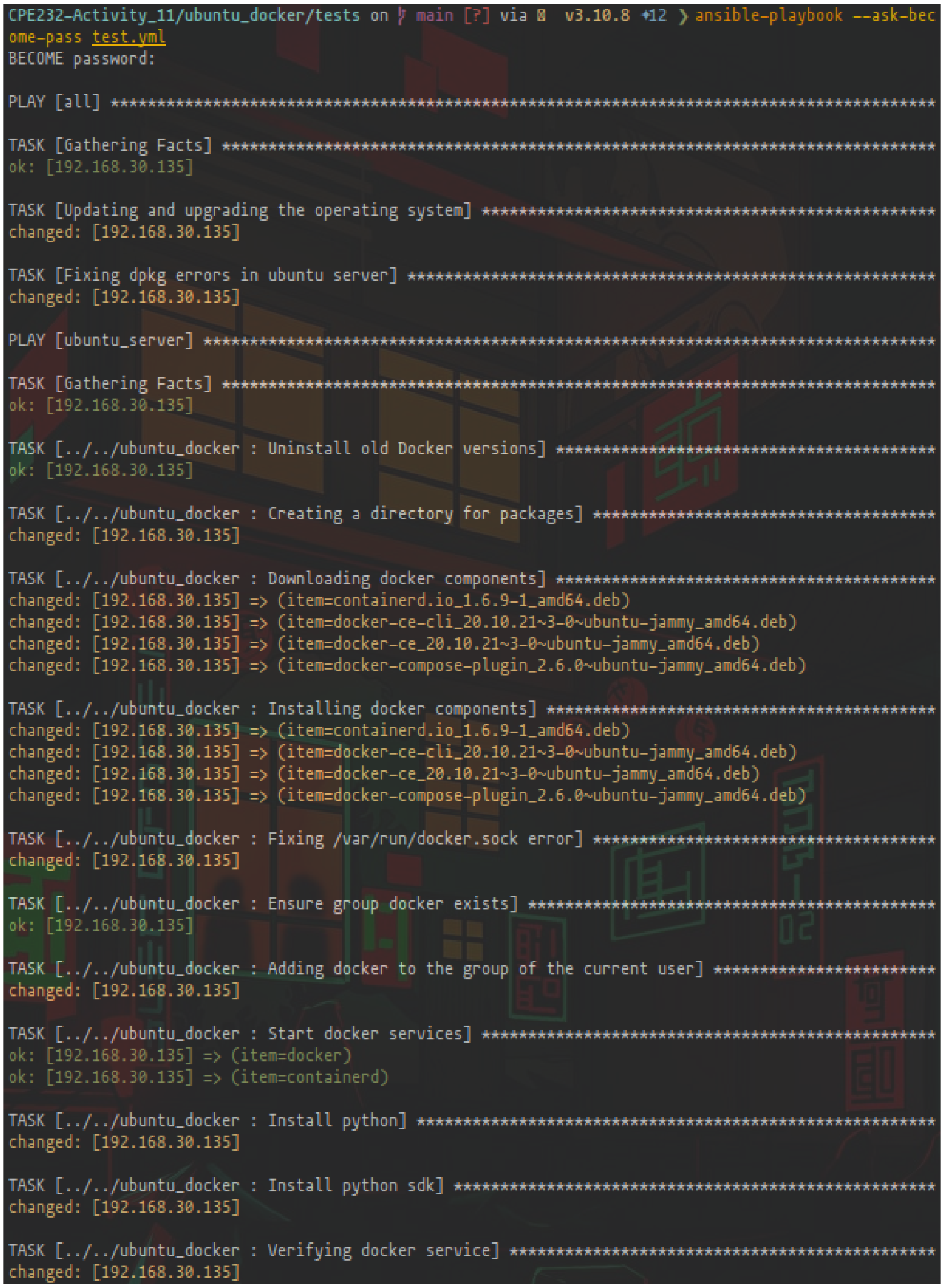
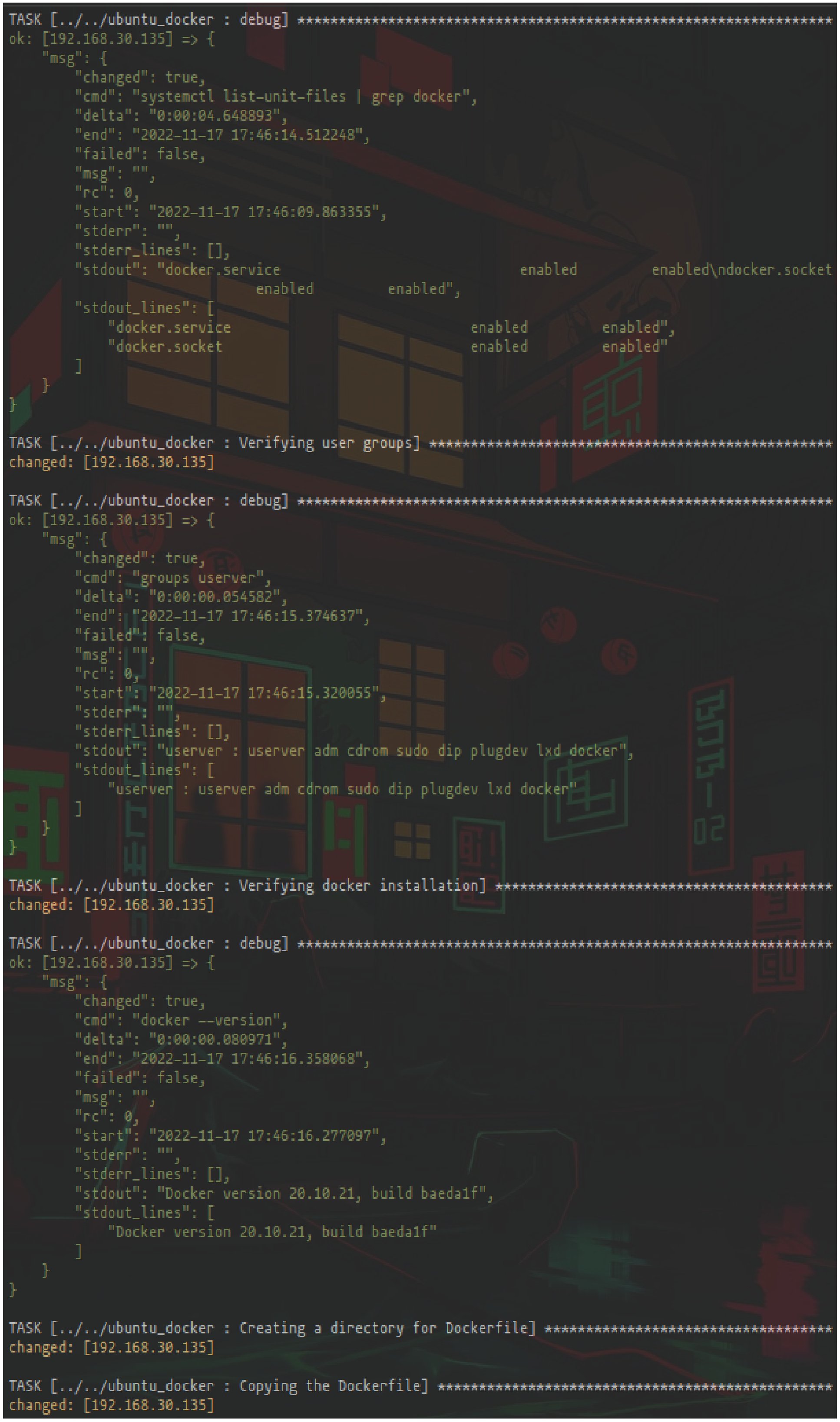
inventory	 inventory [ubuntu_server] 192.168.30.135 ansible_user=userver

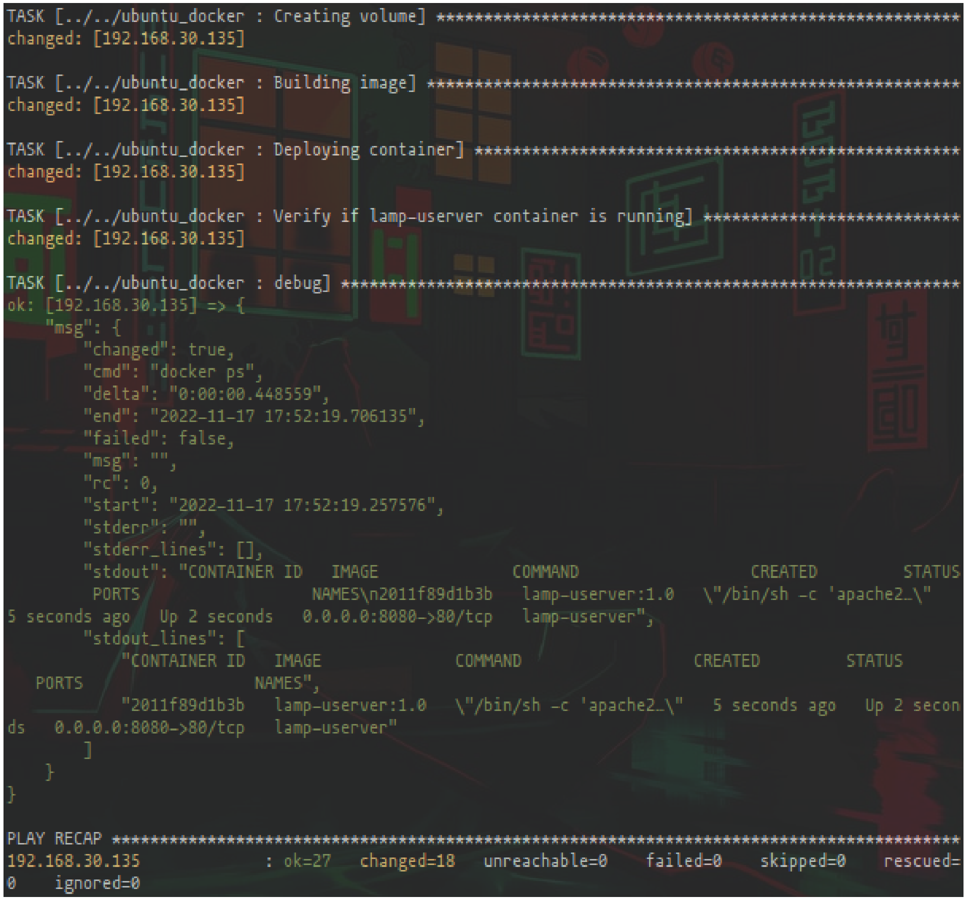
Table 4. The table above shows the contents of each file in the "tests" directory namely test.yml, inventory and ansible.cfg.



```
CPE232-Activity_11/ubuntu_docker/tests on  main [?] via   v3.10.8 +12 ❯ ansible-playbook --ask-bec
ome-pass test.yml
BECOME password:

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

TASK [Gathering Facts] *****************************************************************************
ok: [192.168.30.135]

TASK [Updating and upgrading the operating system] *************************************************
changed: [192.168.30.135]

TASK [Fixing dpkg errors in ubuntu server] *********************************************************
changed: [192.168.30.135]

PLAY [ubuntu_server] *******************************************************************************

TASK [Gathering Facts] *****************************************************************************
ok: [192.168.30.135]

TASK [../../ubuntu_docker : Uninstall old Docker versions] *****************************************
ok: [192.168.30.135]

TASK [../../ubuntu_docker : Creating a directory for packages] *************************************
changed: [192.168.30.135]

TASK [../../ubuntu_docker : Downloading docker components] *****************************************
changed: [192.168.30.135] => (item=containerd.io_1.6.9-1_amd64.deb)
changed: [192.168.30.135] => (item=docker-ce-cli_20.10.21~3-0~ubuntu-jammy_amd64.deb)
changed: [192.168.30.135] => (item=docker-ce_20.10.21~3-0~ubuntu-jammy_amd64.deb)
changed: [192.168.30.135] => (item=docker-compose-plugin_2.6.0~ubuntu-jammy_amd64.deb)

TASK [../../ubuntu_docker : Installing docker components] ******************************************
changed: [192.168.30.135] => (item=containerd.io_1.6.9-1_amd64.deb)
changed: [192.168.30.135] => (item=docker-ce-cli_20.10.21~3-0~ubuntu-jammy_amd64.deb)
changed: [192.168.30.135] => (item=docker-ce_20.10.21~3-0~ubuntu-jammy_amd64.deb)
changed: [192.168.30.135] => (item=docker-compose-plugin_2.6.0~ubuntu-jammy_amd64.deb)

TASK [../../ubuntu_docker : Fixing /var/run/docker.sock error] *************************************
changed: [192.168.30.135]

TASK [../../ubuntu_docker : Ensure group docker exists] ********************************************
ok: [192.168.30.135]

TASK [../../ubuntu_docker : Adding docker to the group of the current user] ************************
changed: [192.168.30.135]

TASK [../../ubuntu_docker : Start docker services] *************************************************
ok: [192.168.30.135] => (item=docker)
ok: [192.168.30.135] => (item=containerd)

TASK [../../ubuntu_docker : Install python] ********************************************************
changed: [192.168.30.135]

TASK [../../ubuntu_docker : Install python sdk] ****************************************************
changed: [192.168.30.135]

TASK [../../ubuntu_docker : Verifying docker service] **********************************************
changed: [192.168.30.135]
```



```
TASK [../../ubuntu_docker : debug] *****************************************************************
ok: [192.168.30.135] => {
    "msg": {
        "changed": true,
        "cmd": "systemctl list-unit-files | grep docker",
        "delta": "0:00:04.648893",
        "end": "2022-11-17 17:46:14.512248",
        "failed": false,
        "msg": "",
        "rc": 0,
        "start": "2022-11-17 17:46:09.863355",
        "stderr": "",
        "stderr_lines": [],
        "stdout": "docker.service                             enabled         enabled\ndocker.socket
                              enabled         enabled",
        "stdout_lines": [
            "docker.service                             enabled         enabled",
            "docker.socket                              enabled         enabled"
        ]
    }
}

TASK [../../ubuntu_docker : Verifying user groups] *************************************************
changed: [192.168.30.135]

TASK [../../ubuntu_docker : debug] *****************************************************************
ok: [192.168.30.135] => {
    "msg": {
        "changed": true,
        "cmd": "groups userver",
        "delta": "0:00:00.054582",
        "end": "2022-11-17 17:46:15.374637",
        "failed": false,
        "msg": "",
        "rc": 0,
        "start": "2022-11-17 17:46:15.320055",
        "stderr": "",
        "stderr_lines": [],
        "stdout": "userver : userver adm cdrom sudo dip plugdev lxd docker",
        "stdout_lines": [
            "userver : userver adm cdrom sudo dip plugdev lxd docker"
        ]
    }
}

TASK [../../ubuntu_docker : Verifying docker installation] *****************************************
changed: [192.168.30.135]

TASK [../../ubuntu_docker : debug] *****************************************************************
ok: [192.168.30.135] => {
    "msg": {
        "changed": true,
        "cmd": "docker --version",
        "delta": "0:00:00.080971",
        "end": "2022-11-17 17:46:16.358068",
        "failed": false,
        "msg": "",
        "rc": 0,
        "start": "2022-11-17 17:46:16.277097",
        "stderr": "",
        "stderr_lines": [],
        "stdout": "Docker version 20.10.21, build baeda1f",
        "stdout_lines": [
            "Docker version 20.10.21, build baeda1f"
        ]
    }
}

TASK [../../ubuntu_docker : Creating a directory for Dockerfile] ***********************************
changed: [192.168.30.135]

TASK [../../ubuntu_docker : Copying the Dockerfile] ************************************************
changed: [192.168.30.135]
```



```
TASK [../../ubuntu_docker : Creating volume] *******************************************************
changed: [192.168.30.135]

TASK [../../ubuntu_docker : Building image] ********************************************************
changed: [192.168.30.135]

TASK [../../ubuntu_docker : Deploying container] ***************************************************
changed: [192.168.30.135]

TASK [../../ubuntu_docker : Verify if lamp-userver container is running] ***************************
changed: [192.168.30.135]

TASK [../../ubuntu_docker : debug] *****************************************************************
ok: [192.168.30.135] => {
    "msg": {
        "changed": true,
        "cmd": "docker ps",
        "delta": "0:00:00.448559",
        "end": "2022-11-17 17:52:19.706135",
        "failed": false,
        "msg": "",
        "rc": 0,
        "start": "2022-11-17 17:52:19.257576",
        "stderr": "",
        "stderr_lines": [],
        "stdout": "CONTAINER ID   IMAGE                COMMAND                  CREATED         STATUS
         PORTS                  NAMES\n2011f89d1b3b   lamp-userver:1.0   \"/bin/sh -c 'apache2…\"
5 seconds ago   Up 2 seconds   0.0.0.0:8080->80/tcp   lamp-userver",
        "stdout_lines": [
            "CONTAINER ID   IMAGE                COMMAND                  CREATED         STATUS
   PORTS                  NAMES",
            "2011f89d1b3b   lamp-userver:1.0   \"/bin/sh -c 'apache2…\"   5 seconds ago   Up 2 secon
ds   0.0.0.0:8080->80/tcp   lamp-userver"
        ]
    }
}

PLAY RECAP *****************************************************************************************
192.168.30.135             : ok=27   changed=18   unreachable=0    failed=0    skipped=0    rescued=
0    ignored=0
```

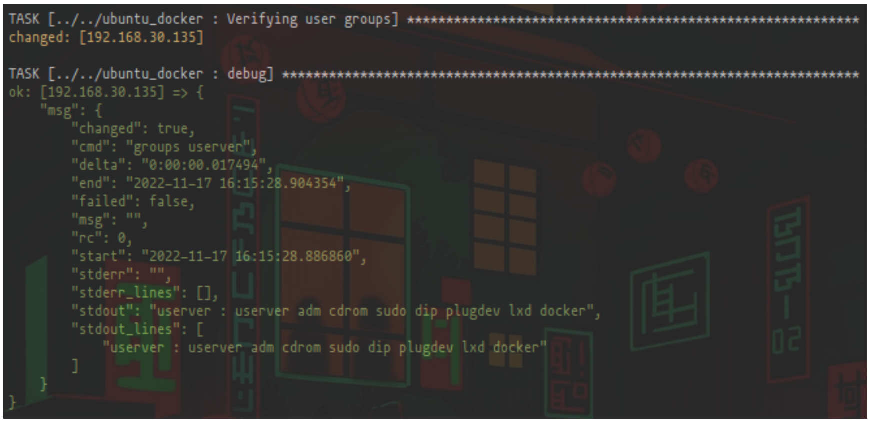
Figure 5. The screenshot above shows the output after running the playbook. It also shows the verification for the user's groups, docker installation and container/s status using debug messages.

<table>
<tr><th colspan="2">Summary of Debug Messages from the Output and
Screenshot of its Equivalent in the Server</th></tr>
<tr><td rowspan="2">

Adding Docker to Group

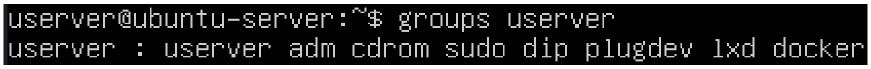
The debug message shows the docker services by executing the command "systemctl list-unit-files".

</td><td>



```
TASK [../../ubuntu_docker : Verifying user groups] ************************************************************
changed: [192.168.30.135]

TASK [../../ubuntu_docker : debug] ****************************************************************************
ok: [192.168.30.135] => {
    "msg": {
        "changed": true,
        "cmd": "groups userver",
        "delta": "0:00:00.017494",
        "end": "2022-11-17 16:15:28.904354",
        "failed": false,
        "msg": "",
        "rc": 0,
        "start": "2022-11-17 16:15:28.886860",
        "stderr": "",
        "stderr_lines": [],
        "stdout": "userver : userver adm cdrom sudo dip plugdev lxd docker",
        "stdout_lines": [
            "userver : userver adm cdrom sudo dip plugdev lxd docker"
        ]
    }
}
```

</td></tr>
<tr><td>

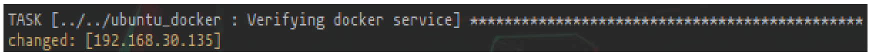


```
userver@ubuntu-server:~$ groups userver
userver : userver adm cdrom sudo dip plugdev lxd docker
```

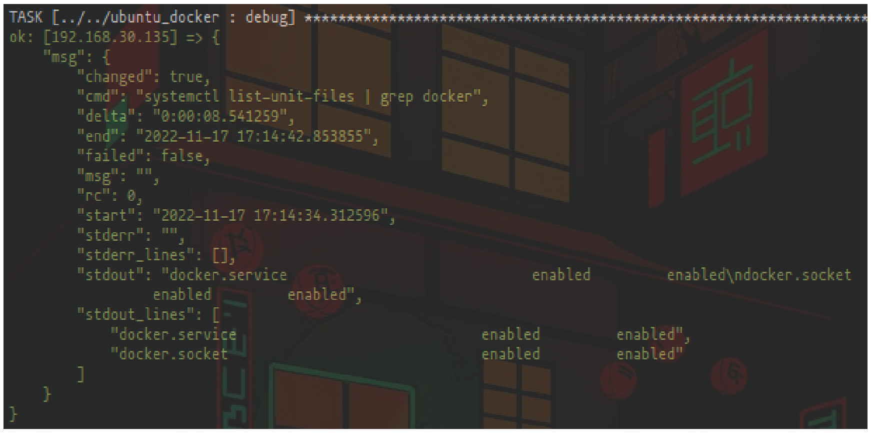
</td></tr>
</table>

Docker Service

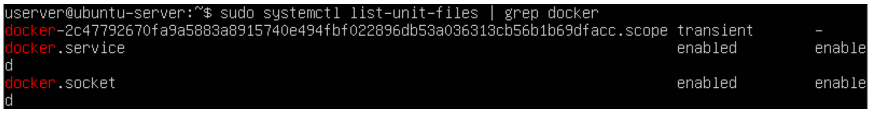
The debug message shows the docker services by executing the command “systemctl list-unit-files”.



```
TASK [../../ubuntu_docker : Verifying docker service] *************************************************
changed: [192.168.30.135]
```



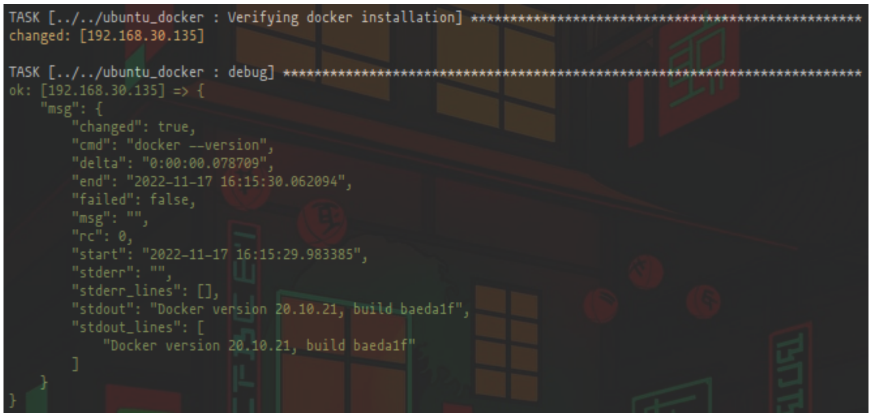
```
TASK [../../ubuntu_docker : debug] ***********************************************************************
ok: [192.168.30.135] => {
    "msg": {
        "changed": true,
        "cmd": "systemctl list-unit-files | grep docker",
        "delta": "0:00:08.541259",
        "end": "2022-11-17 17:14:42.853855",
        "failed": false,
        "msg": "",
        "rc": 0,
        "start": "2022-11-17 17:14:34.312596",
        "stderr": "",
        "stderr_lines": [],
        "stdout": "docker.service                             enabled         enabled\ndocker.socket
                  enabled         enabled",
        "stdout_lines": [
            "docker.service                             enabled         enabled",
            "docker.socket                              enabled         enabled"
        ]
    }
}
```



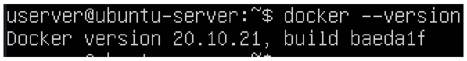
```
userver@ubuntu-server:~$ sudo systemctl list-unit-files | grep docker
docker-2c47792670fa9a5883a8915740e494fbf022896db53a036313cb56b1b69dfacc.scope transient       -
docker.service                                                                enabled         enable
d
docker.socket                                                                 enabled         enable
d
```

Verify Docker Installation

The debug message get the output of the command “docker –version” to verify that docker is installed successfully.



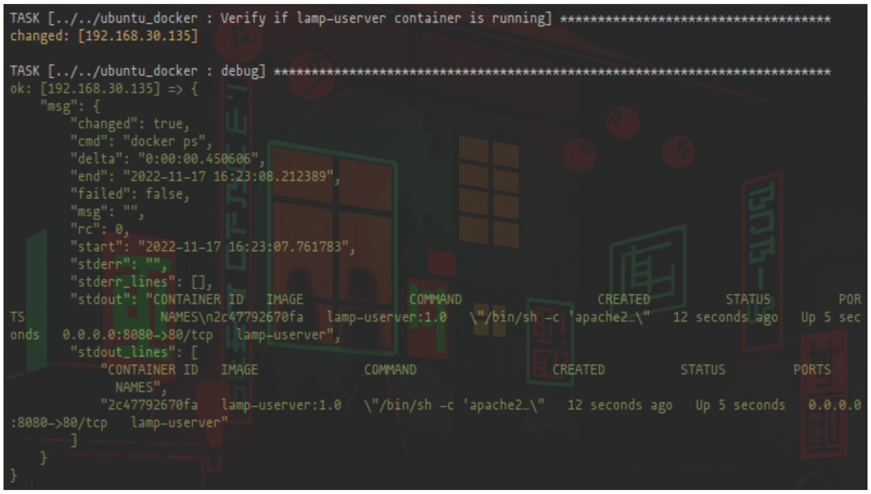
```
TASK [../../ubuntu_docker : Verifying docker installation] ********************************************
changed: [192.168.30.135]

TASK [../../ubuntu_docker : debug] ***********************************************************************
ok: [192.168.30.135] => {
    "msg": {
        "changed": true,
        "cmd": "docker --version",
        "delta": "0:00:00.078709",
        "end": "2022-11-17 16:15:30.062094",
        "failed": false,
        "msg": "",
        "rc": 0,
        "start": "2022-11-17 16:15:29.983385",
        "stderr": "",
        "stderr_lines": [],
        "stdout": "Docker version 20.10.21, build baeda1f",
        "stdout_lines": [
            "Docker version 20.10.21, build baeda1f"
        ]
    }
}
```



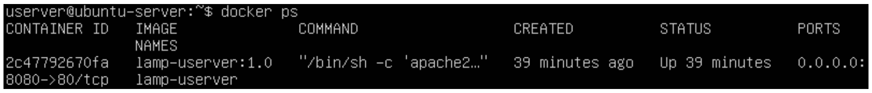
```
userver@ubuntu-server:~$ docker --version
Docker version 20.10.21, build baeda1f
```

Verify if the Container is Running

The debug message shows the running containers. It is achieved by running the command “dcoker ps”.

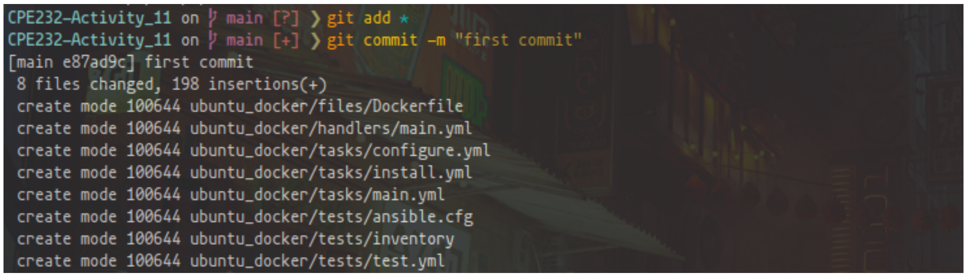


```
TASK [../../ubuntu_docker : Verify if lamp-userver container is running] ************************************
changed: [192.168.30.135]

TASK [../../ubuntu_docker : debug] ***************************************************************************
ok: [192.168.30.135] => {
    "msg": {
        "changed": true,
        "cmd": "docker ps",
        "delta": "0:00:00.450606",
        "end": "2022-11-17 16:23:08.212389",
        "failed": false,
        "msg": "",
        "rc": 0,
        "start": "2022-11-17 16:23:07.761783",
        "stderr": "",
        "stderr_lines": [],
        "stdout": "CONTAINER ID   IMAGE               COMMAND                  CREATED          STATUS         POR
TS             NAMES\n2c47792670fa   lamp-userver:1.0   \"/bin/sh -c 'apache2…\"   12 seconds ago   Up 5 sec
onds   0.0.0.0:8080->80/tcp   lamp-userver",
        "stdout_lines": [
            "CONTAINER ID   IMAGE               COMMAND                  CREATED          STATUS         PORTS
              NAMES",
            "2c47792670fa   lamp-userver:1.0   \"/bin/sh -c 'apache2…\"   12 seconds ago   Up 5 seconds   0.0.0.0
:8080->80/tcp   lamp-userver"
        ]
    }
}
```

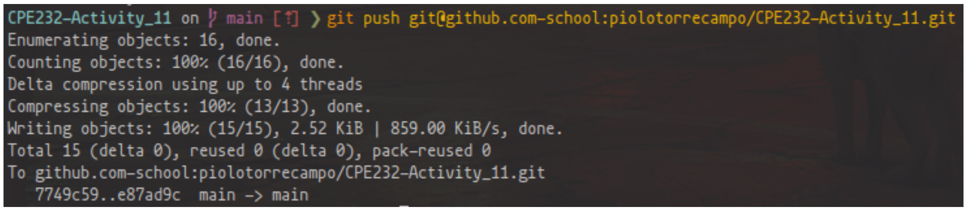


```
userver@ubuntu-server:~$ docker ps
CONTAINER ID   IMAGE              COMMAND                  CREATED          STATUS          PORTS
               NAMES
2c47792670fa   lamp-userver:1.0   "/bin/sh -c 'apache2…"   39 minutes ago   Up 39 minutes   0.0.0.0:
8080->80/tcp   lamp-userver
```

Table 5. The table above shows the summary of debug messages with the screenshot of its equivalent in the server.

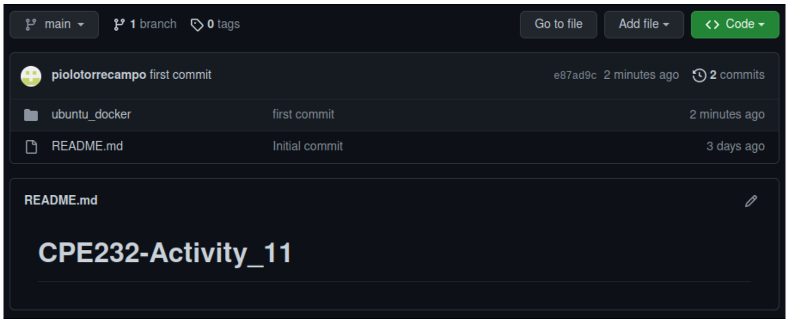


```
CPE232-Activity_11 on ᚠ main [?] ❯ git add *
CPE232-Activity_11 on ᚠ main [+] ❯ git commit -m "first commit"
[main e87ad9c] first commit
 8 files changed, 198 insertions(+)
 create mode 100644 ubuntu_docker/files/Dockerfile
 create mode 100644 ubuntu_docker/handlers/main.yml
 create mode 100644 ubuntu_docker/tasks/configure.yml
 create mode 100644 ubuntu_docker/tasks/install.yml
 create mode 100644 ubuntu_docker/tasks/main.yml
 create mode 100644 ubuntu_docker/tests/ansible.cfg
 create mode 100644 ubuntu_docker/tests/inventory
 create mode 100644 ubuntu_docker/tests/test.yml
```



```
CPE232-Activity_11 on ᚠ main [⇡] ❯ git push git@github.com-school:piolotorrecampo/CPE232-Activity_11.git
Enumerating objects: 16, done.
Counting objects: 100% (16/16), done.
Delta compression using up to 4 threads
Compressing objects: 100% (13/13), done.
Writing objects: 100% (15/15), 2.52 KiB | 859.00 KiB/s, done.
Total 15 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com-school:piolotorrecampo/CPE232-Activity_11.git
   7749c59..e87ad9c  main -> main
```

Figure 6. The image above shows the process of pushing the contents of the repository to Github.



piolotorrecampo first commit e87ad9c 2 minutes ago 2 commits

ubuntu_docker	first commit	2 minutes ago
README.md	Initial commit	3 days ago

README.md

CPE232-Activity_11

Figure 7. The picture above shows the pages of the repository.

Note: Before proceeding in deploying the playbook, I have first installed the module for docker that helps to create, build and deploy containers. The command **[ansible-galaxy collection install community.docker]** is used to download the community.docker module. Also, there is another dependency named "Docker SDK" that should be installed on the server. The installation is already included in the playbook.

5. Links

References:

- https://docs.ansible.com/ansible/latest/collections/community/docker/docker_image_module.html
- https://docs.ansible.com/ansible/2.9/modules/docker_container_module.html
- https://www.youtube.com/watch?v=CQk9AOPh5pw&t=600s
- https://docs.docker.com/engine/install/ubuntu/

Github Repository: https://github.com/piolotorrecampo/CPE232-Activity_11.git

6. Reflections:

Answer the following:

1. What are the benefits of implementing containerizations?

   - The benefits of implementing containerizations in a server is that it enables it to isolate applications in a lightweight virtual machine rather than providing a separate fully functional operating system. The operating system installed in a container is a portion of the full operating system which only uses the needed file systems to run. Since a container utilizes a smaller amount of memory, then we can deploy multiple containers that are running simultaneously. The other benefits of containerization is managing and building the containers by the use of Dockerfile. Dockerfile is a set of instructions that instructs the container to upgrade, install some applications and many more. We can visualize these instructions using a terminal but different in syntax. Containerization also implements persistent storage and provides an isolated view of virtual network adapters.

7. Conclusions:

The activity achieved its objective to practice the student in creating a Dockerfile and using the ansible to build an image with it and deploy it as a container which enables the continuous delivery process for an enterprise. This activity gives me a clear understanding of how Docker works in a live environment. Docker reduces the resources in a computing system that enables it to replace a full working operating system to minimal installation by using docker containerization. Containerization works by isolating a group of applications in the host operating system. It is not a fully operational operating system but it has the dependencies and is capable of launching an application.

In performing this activity, I have decided to make all of the steps regarding installation and enabling docker, building a Dockerfile, and deploying the container in one ansible playbook. In installing and enabling the services of docker, I relied on the steps in the official page of docker and convert into ansible syntax. In building the Dockerfile and deploying it as a container, I utilized the given code in the ansible official documentation for docker support. Overall, this activity gives me confidence in writing Dockerfile and as well as building an image and deploying a container.

8. Honor Pledge:

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