Name: Pastrana, Mark Laurenz S.	Date Performed: Nov 21, 2023
Course/Section: CPE31S5	Date Submitted: Nov 22, 2023
Instructor: Engr. Richard Roman	Semester and SY: 1st, 2023-2024

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.

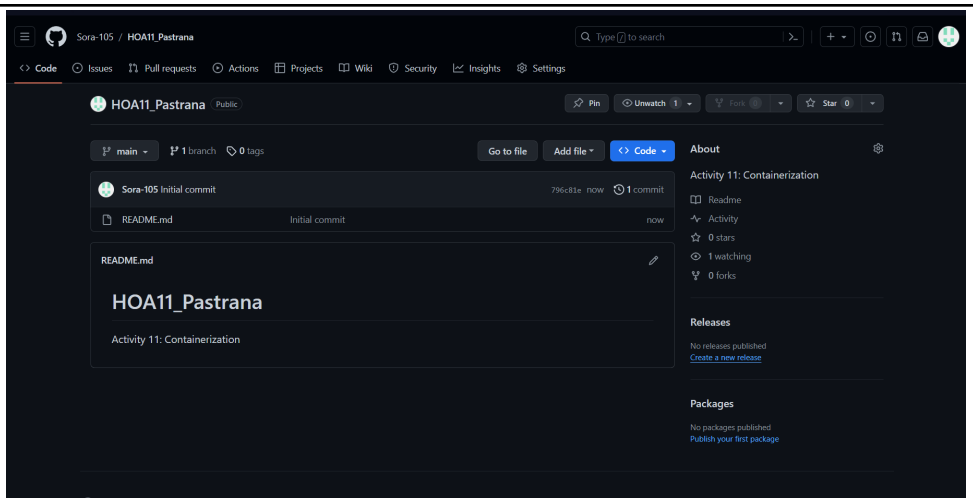
Source: https://docs.microsoft.com/en-us/virtualization/windowscontainers/about/containers-vs-vm

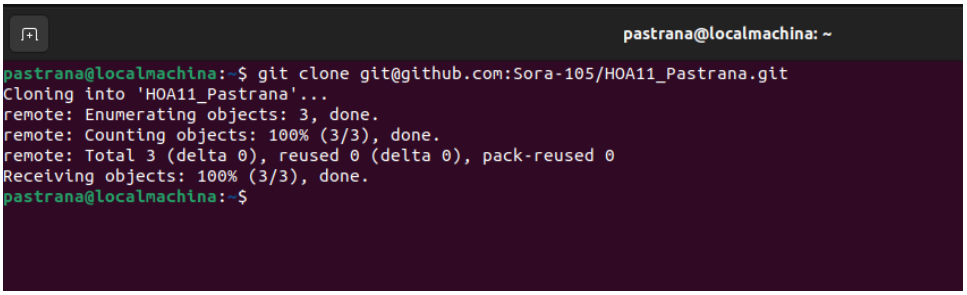
3. Tasks

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.

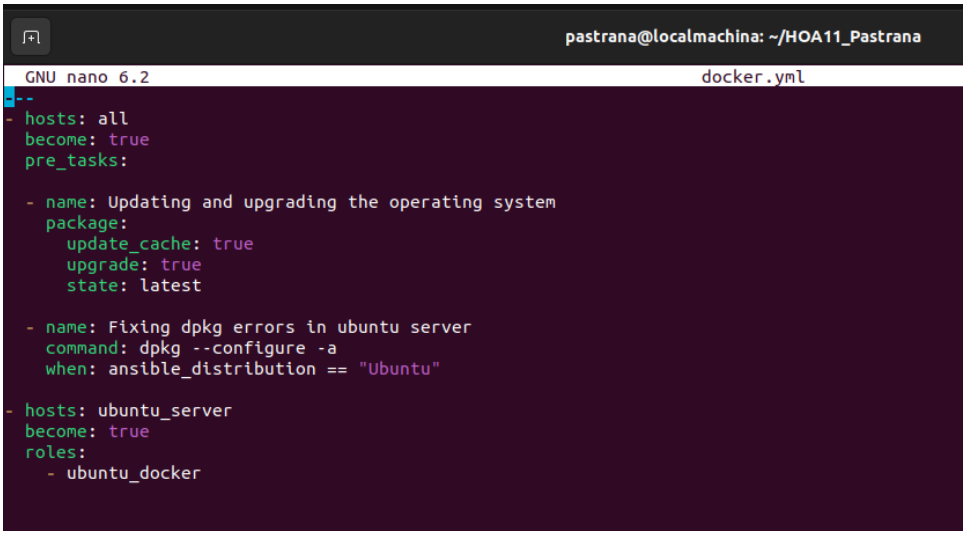
4. Output (screenshots and explanations)

1. Create a new repository for this activity.





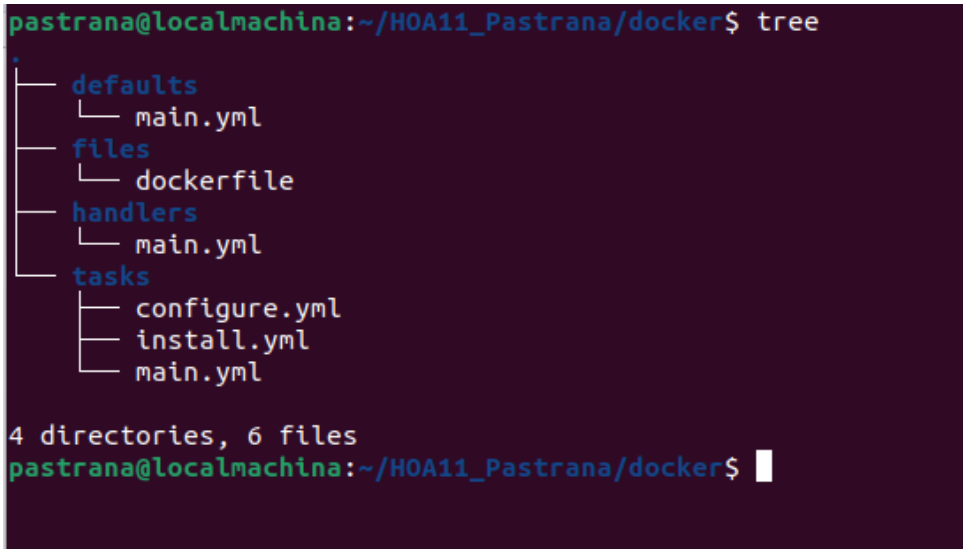
docker.yml



ansible.cfg

inventory

2. Create a Dockerfile to install web and DB server.



-dockerfile

```
  GNU nano 6.2                                                  dockerfile *
FROM ubuntu:latest
MAINTAINER pastrana <qmlspastrana@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
```

-defaults/main.yml

```
pastrana@localmachina: ~/HOA11_Pastrana/docker/defaults
  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
```

-handlers/main.yml

```
pastrana@localmachina: ~/HOA11_Pastrana/docker/handlers
  GNU nano 6.2                                                  main.yml *
---
- name: Start docker
  service:
    name: "{{ item }}"
    state: restarted
    enabled: true
  with_items:
    - docker
    - containerd
```

-tasks/configure.yml

pastrana@localmachina: ~/HOA11_Pastrana/docker/tasks

GNU nano 6.2 configure.yml

```
---
- 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 }}"
```

-tasks/install.yml

pastrana@localmachina: ~/HOA11_Pastrana/docker/tasks

GNU nano 6.2 install.yml

```
- 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:
    - "{{ docker_apps.containerd }}.deb"
    - "{{ docker_apps.docker_ce_cli }}.deb"
    - "{{ docker_apps.docker_ce }}.deb"
    - "{{ docker_apps.docker_compose }}.deb"

- name: Installing docker components
  shell: |
    cd /home/userver/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: 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

- 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 }}"
```

-tasks/main.yml

```
pastrana@localmachina: ~/HOA11_Pastrana/docker/tasks
 GNU nano 6.2                                  main.yml
---
- import_tasks: install.yml
- import_tasks: configure.yml
```

3. Install and build the Dockerfile using Ansible.

```
pastrana@localmachina:~/MOA11_Pastrana$ ansible-playbook --ask-become-pass docker.yml
BECOME password:

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

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

TASK [Updating and upgrading the operating system] *****************************************************************************************************
changed: [192.168.56.106]

TASK [Fixing dpkg errors in ubuntu server] *************************************************************************************************************
changed: [192.168.56.106]

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

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

TASK [docker : Uninstall old Docker versions] **********************************************************************************************************
ok: [192.168.56.106]

TASK [docker : Creating a directory for packages] ******************************************************************************************************
changed: [192.168.56.106]

TASK [docker : Downloading docker components] **********************************************************************************************************
changed: [192.168.56.106] => (item=containerd.io_1.6.9-1_amd64.deb)
changed: [192.168.56.106] => (item=docker-ce-cli_20.10.21~3-0~ubuntu-jammy_amd64.deb)
changed: [192.168.56.106] => (item=docker-ce_20.10.21~3-0~ubuntu-jammy_amd64.deb)
changed: [192.168.56.106] => (item=docker-compose-plugin_2.6.0~ubuntu-jammy_amd64.deb)
```

```
TASK [docker : Installing docker components] ***********************************************************************************************************
changed: [192.168.56.106] => (item=containerd.io_1.6.9-1_amd64.deb)
changed: [192.168.56.106] => (item=docker-ce-cli_20.10.21~3-0~ubuntu-jammy_amd64.deb)
changed: [192.168.56.106] => (item=docker-ce_20.10.21~3-0~ubuntu-jammy_amd64.deb)
changed: [192.168.56.106] => (item=docker-compose-plugin_2.6.0~ubuntu-jammy_amd64.deb)

TASK [docker : Fixing /var/run/docker.sock error] ******************************************************************************************************
changed: [192.168.56.106]

TASK [docker : Ensure group docker exists] *************************************************************************************************************
ok: [192.168.56.106]

TASK [docker : Adding docker to the group of the current user] *****************************************************************************************
changed: [192.168.56.106]

TASK [docker : Start docker services] ******************************************************************************************************************
ok: [192.168.56.106] => (item=docker)
ok: [192.168.56.106] => (item=containerd)

TASK [docker : Install python] *************************************************************************************************************************
ok: [192.168.56.106]

TASK [docker : Install python sdk] *********************************************************************************************************************
changed: [192.168.56.106]

TASK [docker : Verifying docker service] ***************************************************************************************************************
changed: [192.168.56.106]
```

```
TASK [docker : debug] **********************************************************************************************************************************
ok: [192.168.56.106] => {
    "msg": {
        "changed": true,
        "cmd": "systemctl list-unit-files | grep docker",
        "delta": "0:00:01.425921",
        "end": "2023-11-22 13:20:56.653107",
        "failed": false,
        "msg": "",
        "rc": 0,
        "start": "2023-11-22 13:20:55.227186",
        "stderr": "",
        "stderr_lines": [],
        "stdout": "docker.service                                enabled         enabled\ndocker.socket                                en
abled         enabled",
        "stdout_lines": [
            "docker.service                               enabled         enabled",
            "docker.socket                                enabled         enabled"
        ]
    }
}

TASK [docker : Verifying user groups] ******************************************************************************************************************
changed: [192.168.56.106]
```

```
TASK [docker : debug] **********************************************************************************************************************************
ok: [192.168.56.100] => {
    "msg": {
        "changed": true,
        "cmd": "groups userver",
        "delta": "0:00:00.005657",
        "end": "2023-11-22 13:20:57.039120",
        "failed": false,
        "msg": "",
        "rc": 0,
        "start": "2023-11-22 13:20:57.033463",
        "stderr": "",
        "stderr_lines": [],
        "stdout": "userver : userver docker",
        "stdout_lines": [
            "userver : userver docker"
        ]
    }
}

TASK [docker : Verifying docker installation] **********************************************************************************************************
changed: [192.168.56.106]
```

```
TASK [docker : debug] ***********************************************************************************************************************
ok: [192.168.56.106] => {
    "msg": {
        "changed": true,
        "cmd": "docker --version",
        "delta": "0:00:00.028976",
        "end": "2023-11-22 13:20:57.459193",
        "failed": false,
        "msg": "",
        "rc": 0,
        "start": "2023-11-22 13:20:57.430217",
        "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 Dockerfile] *****************************************************************************************
changed: [192.168.56.106]

TASK [docker : Copying the dockerfile] ******************************************************************************************************
changed: [192.168.56.106]

TASK [docker : Creating volume] *************************************************************************************************************
changed: [192.168.56.106]

TASK [docker : Building image] **************************************************************************************************************
changed: [192.168.56.106]

TASK [docker : Deploying container] *********************************************************************************************************
changed: [192.168.56.106]
```

```
TASK [docker : Verify if lamp-userver container is running] *********************************************************************************
changed: [192.168.56.106]

TASK [docker : debug] ***********************************************************************************************************************
ok: [192.168.56.106] => {
    "msg": {
        "changed": true,
        "cmd": "docker ps",
        "delta": "0:00:00.151101",
        "end": "2023-11-22 13:22:40.614950",
        "failed": false,
        "msg": "",
        "rc": 0,
        "start": "2023-11-22 13:22:40.463849",
        "stderr": "",
        "stderr_lines": [],
        "stdout": "CONTAINER ID   IMAGE              COMMAND                  CREATED         STATUS                  PORTS                    NAMES\nc224abd30f17   lamp-userver:1.0   \"/bin/sh -c 'apache2…\"   1 second ago   Up Less than a second   0.0.0.0:8080->80/tcp   lamp-userver",
        "stdout_lines": [
            "CONTAINER ID   IMAGE              COMMAND                  CREATED         STATUS                  PORTS                    NAMES",
            "c224abd30f17   lamp-userver:1.0   \"/bin/sh -c 'apache2…\"   1 second ago   Up Less than a second   0.0.0.0:8080->80/tcp   lamp-userver"
        ]
    }
}

PLAY RECAP **********************************************************************************************************************************
192.168.56.106             : ok=27   changed=17   unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

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

```
pastrana@localmachina:~/HOA11_Pastrana$ git add *
pastrana@localmachina:~/HOA11_Pastrana$ git commit -m "HOA11_Pastrana"
[main ca284a8] HOA11_Pastrana
 10 files changed, 200 insertions(+)
 create mode 100644 ansible.cfg
 create mode 100644 docker.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/configure.yml
 create mode 100644 docker/tasks/install.yml
 create mode 100644 docker/tasks/main.yml
 create mode 100644 hosts
 create mode 100644 inventory
pastrana@localmachina:~/HOA11_Pastrana$ git push origin
Enumerating objects: 18, done.
Counting objects: 100% (18/18), done.
Delta compression using up to 4 threads
Compressing objects: 100% (12/12), done.
Writing objects: 100% (17/17), 2.66 KiB | 1.33 MiB/s, done.
Total 17 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:Sora-105/HOA11_Pastrana.git
   796c81e..ca284a8  main -> main
pastrana@localmachina:~/HOA11_Pastrana$
```

https://github.com/Sora-105/HOA11_Pastrana

Reflections:

Answer the following:

1. **What are the benefits of implementing containerizations?**
    - Containerization offers a range of benefits for software development and deployment, including portability, resource efficiency, isolation, scalability, agility, version control, improved security, faster app startup, and easier management. By packaging applications and their dependencies into standardized units called containers, containerization enables applications to run reliably and consistently across different computing environments, leading to greater efficiency, agility, and security.

Conclusions:

- **Doing this Hands-On Activity taught us on how to install the docker manually and using the ansible-playbook. In the school lab we install the docker manually on our workstation and then at home we configure using the ansible-playbook and then when in the lab we can just copy our git repository on the workstation. We also utilized in the playbook the containerization. Overall, containerization is a powerful tool that can help to improve the efficiency, agility, and security of software development and deployment. If you are developing or deploying applications, I recommend considering using containerization.**