

Name: Laxamana, Abigail	Date Performed: November 16, 2023
Course/Section: CPE 232 - CPE31S6	Date Submitted: November 16, 2023
Instructor: Dr. Jonathan Taylar	Semester and SY: 1st sem, SY: 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	
<p>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.</p> <p>Source: https://docs.docker.com/get-started/overview/</p> <p>You may also check the difference between containers and virtual machines. Click the link given below.</p> <p>Source: https://docs.microsoft.com/en-us/virtualization/windowscontainers/about/containers-vs-vm</p>	
3. Tasks	
<ol style="list-style-type: none"> 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)	

To start the activity, you need to clone your newly created repository to your workstation using the **git clone** command followed by the ssh link of your github repository.

```
laxamana_ubuntu@workstation:~$ git clone git@github.com:Abigaiiil/hoa11laxamana.git
Cloning into 'hoa11laxamana'...
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.
```

Then, you need to create **inventory** and **ansible.cfg** files. The **inventory** file contains the addresses of the servers you'll be needing for this activity while the **ansible.cfg** file contains the settings.

The inventory file should look like this

```
laxamana_ubuntu@workstation:~/hoa11laxamana$ cat inventory
[ubuntu]
192.168.56.103

[centos]
Laxamana@192.168.56.110
```

The ansible.cfg file should look like this

```
laxamana_ubuntu@workstation:~/hoa11laxamana$ cat ansible.cfg
[defaults]

inventory = inventory
host_key_checking = False

deprecation_warnings = False

remote_user = laxamana_ubuntu

private_key_file = ~/.ssh/
```

To install docker, enable docker socket, add docker group to current user, create dockerfile to install web and db server, and install and build the dockerfile, we could use a playbook as it is the most efficient way to do it.

The playbook should look like this

```
laxamana_ubuntu@workstation:~/hoa11laxamana$ sudo nano hoa11laxamana.yml
laxamana_ubuntu@workstation:~/hoa11laxamana$ cat hoa11laxamana.yml
```

```
---
- hosts: ubuntu
  become: true
  pre_tasks:
    - name: dpkg for Ubuntu
      shell:
        dpkg --configure -a
      when: ansible_distribution == "Ubuntu"

    - name: Install Docker (Ubuntu)
      apt:
        name: docker.io
        state: latest
      when: ansible_distribution == "Ubuntu"

    - name: Install SDK (Ubuntu)
      shell:
        pip3 install docker-py

    - name: Ensure Docker group exists (Ubuntu)
      group:
        name: docker
        state: present
      when: ansible_distribution == "Ubuntu"

    - name: Adding user to Docker group (Ubuntu)
      user:
        name: abbylaxamana
        groups: docker
        append: yes
      when: ansible_distribution == "Ubuntu"
```

```
laxamana_ubuntu@workstation: ~/hoa11laxamana
```

```
File Edit View Search Terminal Help
when: ansible_distribution == "Ubuntu"

- name: Enable/Restart Docker (Ubuntu)
  service:
    name: docker
    state: started
    enabled: yes
  when: ansible_distribution == "Ubuntu"

- name: Creating Directory for Dockerfile (Ubuntu)
  file:
    path: /root/demo-dockerfile
    state: directory
    owner: root
    group: root
    mode: '0755'
  when: ansible_distribution == "Ubuntu"

- name: Importing Dockerfile (Ubuntu)
  copy:
    src: dockerfile
    dest: /root/demo-dockerfile/dockerfile
    owner: root
    group: root
    mode: '0755'
  when: ansible_distribution == "Ubuntu"

- hosts: centos
  become: true
  pre_tasks:
    - name: Install required packages (CentOS)
      yum:
        name:
          - yum-utils
          - device-mapper-persistent-data
          - lvm2
        state: present
      when: ansible_distribution == "CentOS"
```

```

- name: Add Docker repository (CentOS)
  yum_repository:
    name: docker-ce
    description: Docker CE Stable - $basearch
    baseurl: https://download.docker.com/linux/centos/7/$basearch/stable
    gpgkey: https://download.docker.com/linux/centos/gpg
    enabled: yes
  when: ansible_distribution == "CentOS"

- name: Install Docker (CentOS)
  yum:
    name: docker-ce
    state: present
  when: ansible_distribution == "CentOS"

- name: Start and enable Docker service (CentOS)
  systemd:
    name: docker
    state: started
    enabled: yes
  when: ansible_distribution == "CentOS"
laxamana_ubuntu@workstation:~/hoa11laxamana$

```

The dockerfile should look like this

```

laxamana_ubuntu@workstation:~/hoa11laxamana$ sudo nano dockerfile
laxamana_ubuntu@workstation:~/hoa11laxamana$ cat dockerfile
FROM ubuntu
MAINTAINER laxamana <qamlaxamana01@tlp.edu.ph>

ARG DEBIAN_FRONTEND=noninteractive

RUN apt-get -y update

RUN apt packages; apt dist-upgrade -y

RUN apt install -y apache2 mariadb-server

ENTRYPOINT apache2ctl -D FOREGROUND

```

Process

```

laxamana_ubuntu@workstation:~/hoa11laxamana$ sudo nano hoa11laxamana.yml
laxamana_ubuntu@workstation:~/hoa11laxamana$ ansible-playbook --ask-become-pass hoa11laxamana.yml
BECOME password:

PLAY [ubuntu] *****

TASK [Gathering Facts] *****
ok: [192.168.56.103]

TASK [dpkg for Ubuntu] *****
changed: [192.168.56.103]

TASK [Install Docker (Ubuntu)] *****
ok: [192.168.56.103]

TASK [Install SDK (Ubuntu)] *****
changed: [192.168.56.103]

TASK [Ensure Docker group exists (Ubuntu)] *****
ok: [192.168.56.103]

TASK [Adding user to Docker group (Ubuntu)] *****
changed: [192.168.56.103]

TASK [Enable/Restart Docker (Ubuntu)] *****
ok: [192.168.56.103]

TASK [Creating Directory for Dockerfile (Ubuntu)] *****
changed: [192.168.56.103]

TASK [Importing Dockerfile (Ubuntu)] *****
changed: [192.168.56.103]

PLAY [centos] *****

TASK [Gathering Facts] *****
ok: [Laxamana@192.168.56.110]

```

```

laxamana_ubuntu@workstation: ~/hoa11laxamana
File Edit View Search Terminal Help

PLAY [centos] *****

TASK [Gathering Facts] *****
ok: [Laxamana@192.168.56.110]

TASK [Install required packages (CentOS)] *****
ok: [Laxamana@192.168.56.110]

TASK [Add Docker repository (CentOS)] *****
changed: [Laxamana@192.168.56.110]

TASK [Install Docker (CentOS)] *****
changed: [Laxamana@192.168.56.110]

TASK [Start and enable Docker service (CentOS)] *****
changed: [Laxamana@192.168.56.110]

PLAY RECAP *****
192.168.56.103      : ok=9    changed=5    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0
Laxamana@192.168.56.110 : ok=5    changed=3    unreachable=0    failed=0    skipped=0    rescued=0
ignored=0

```

Proof of github commit

```

laxamana_ubuntu@workstation:~/hoa11laxamana$ git add .
laxamana_ubuntu@workstation:~/hoa11laxamana$ git commit -m "Success!!!"
git commit -m "Successgit add .!"
[main e45f585] Successgit add .!
4 files changed, 117 insertions(+)
create mode 100644 ansible.cfg
create mode 100644 dockerfile
create mode 100644 hoa11laxamana.yml
create mode 100644 inventory
laxamana_ubuntu@workstation:~/hoa11laxamana$ git push origin
Counting objects: 6, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 1.34 KiB | 1.34 MiB/s, done.
Total 6 (delta 0), reused 0 (delta 0)
To github.com:Abigaiiil/hoa11laxamana.git
b674b7a..e45f585  main -> main

```

```

[Laxamana@localhost ~]$ systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; vendor preset: disabled)
   Active: active (running) since Thu 2023-11-16 04:33:37 EST; 27min ago
     Docs: https://docs.docker.com
   Main PID: 11928 (dockerd)
      Tasks: 8
     Memory: 31.2M
    CGroup: /system.slice/docker.service
            └─11928 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd...

Nov 16 04:33:36 localhost.localdomain systemd[1]: Starting Docker Application Conta....
Nov 16 04:33:36 localhost.localdomain dockerd[11928]: time="2023-11-16T04:33:36.459..."
Nov 16 04:33:36 localhost.localdomain dockerd[11928]: time="2023-11-16T04:33:36.510..."
Nov 16 04:33:37 localhost.localdomain dockerd[11928]: time="2023-11-16T04:33:37.301..."
Nov 16 04:33:37 localhost.localdomain dockerd[11928]: time="2023-11-16T04:33:37.375..."
Nov 16 04:33:37 localhost.localdomain dockerd[11928]: time="2023-11-16T04:33:37.395...7
Nov 16 04:33:37 localhost.localdomain dockerd[11928]: time="2023-11-16T04:33:37.395..."
Nov 16 04:33:37 localhost.localdomain dockerd[11928]: time="2023-11-16T04:33:37.428..."
Nov 16 04:33:37 localhost.localdomain systemd[1]: Started Docker Application Contai....
Hint: Some lines were ellipsized, use -l to show in full.

```

github repository link:

<https://github.com/Abigaiiiiil/hoa11laxamana.git>

Reflections:

Answer the following:

1. What are the benefits of implementing containerizations?

By enclosing programs and dependencies, reducing compatibility problems, and improving portability from development to production, containerization provides consistency across different environments. Additionally, containerization improves resource economy since it allows several containers to run on the same hardware without sacrificing speed and allows them to share the host system's kernel.

Conclusions:

To sum up, containerization has become a game-changing technology that is redefining application development, deployment, and management. Its unmatched portability and deployment simplicity come from its ability to encapsulate dependencies and programs, guaranteeing consistency across a variety of settings. Because of containers' resource efficiency, infrastructure may be used as efficiently as possible, and their flexibility and scalability allow for dynamic scaling to meet changing demands. Additionally, by isolating applications and conducting vulnerability scans, containerization improves security and speeds up development cycles in line with contemporary DevOps methodologies. Containerization is a key option for enterprises seeking more security, scalability, and agility in their software infrastructure. In today's rapidly evolving technological world, containerization offers a more efficient method of managing and deploying applications.

