



Name: Repani, Justin Jello J.	Date Performed: November 16, 2023
Course/Section: CPE31S6	Date Submitted: November 16, 2023
Instructor: Dr. Jonathan V. 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)	
First step is to create a new git repository for activity 11. And then clone this into the workstation	

← → ↻ https://github.com/JelzLow/HOA11_Repani 70% ☆

HOA11_Repani Public Pin Unwatch 1 Fork 0 Star 0

**Set up GitHub Copilot**
Use GitHub's AI pair programmer to autocomplete suggestions as you code.
[Get started with GitHub Copilot](#)

**Add collaborators to this repository**
Search for people using their GitHub username or email address.
[Invite collaborators](#)

Quick setup — if you've done this kind of thing before
or [HTTPS](#) [SSH](#) [Copy](#)
Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# HOA11_Repani" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin git@github.com:JelzLow/HOA11_Repani.git
git push -u origin main
```

[Copy](#)

```
jello@workstation: ~/HOA11_Repani
File Edit View Search Terminal Help
jello@workstation:~$ ls
CPE232_hoa6      Documents      HOA8_Repani    Repani_PrelimExam
CPE232_Repani    Downloads     HOA9_Repani    Templates
CPE_MIDEXAM_REPANI  examples.desktop  Music          token.txt
Desktop          HOA10_Repani  Pictures       Videos
docker           HOA7_Repani   Public
jello@workstation:~$ git clone git@github.com:JelzLow/HOA11_Repani.git
Cloning into 'HOA11_Repani'...
warning: You appear to have cloned an empty repository.
jello@workstation:~$ cd HOA11_Repani
jello@workstation:~/HOA11_Repani$
```

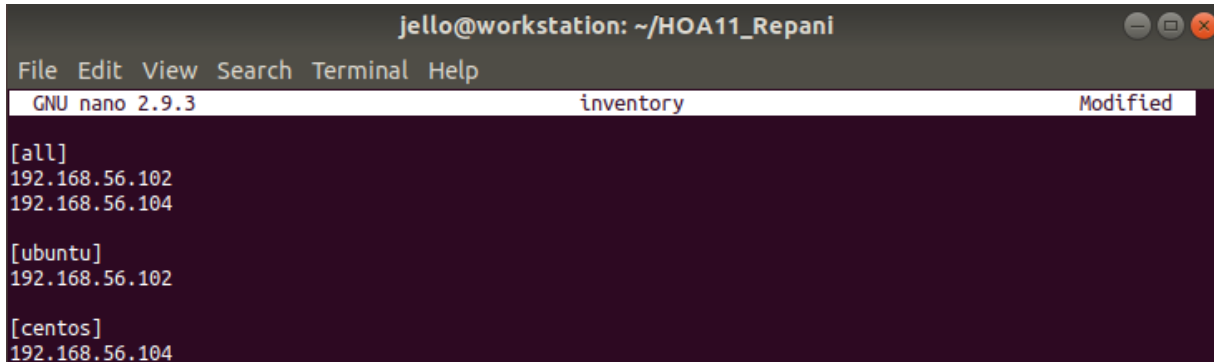
Next step is to copy the ansible.cfg and inventory files from the previous activity and create a roles directory that contains centos and ubuntu each with their own tasks and main.yml

```
jello@workstation:~/HOA11_Repani$ cd ~/HOA10_Repani
jello@workstation:~/HOA10_Repani$ ls
ansible.cfg  elasticstack.yml  inventory  roles
jello@workstation:~/HOA10_Repani$ cp ansible.cfg ~/HOA11_Repani
jello@workstation:~/HOA10_Repani$ cp inventory ~/HOA11_Repani
jello@workstation:~/HOA10_Repani$ cd ~/HOA11_Repani
jello@workstation:~/HOA11_Repani$ ls
ansible.cfg  inventory
```

The next step is to create a directory for the ubuntu docker, it has a similar structure to the roles directory where it has subdirectories containing the different yml files and dockerfile.

```
jello@workstation:~/HOA11_Repani$ tree
.
├── ansible.cfg
├── dockerfile
├── ins_dock.yml
├── install_docker.yml
├── inventory
├── roles
│   ├── centos
│   │   └── tasks
│   │       └── main.yml
│   └── ubuntu
│       └── tasks
│           └── main.yml
└── 5 directories, 7 files
```

The inventory file

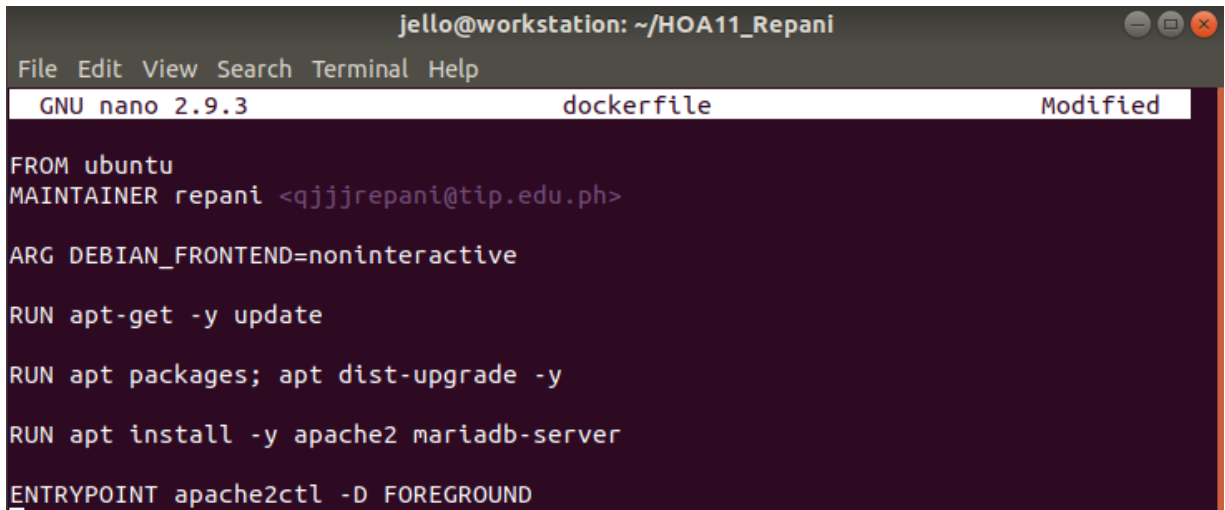


```
jello@workstation: ~/HOA11_Repani
File Edit View Search Terminal Help
GNU nano 2.9.3 inventory Modified
[all]
192.168.56.102
192.168.56.104

[ubuntu]
192.168.56.102

[centos]
192.168.56.104
```

The dockerfile

A screenshot of a terminal window showing a nano text editor. The window title is 'jello@workstation: ~/HOA11_Repani'. The editor's status bar shows 'GNU nano 2.9.3', the filename 'dockerfile', and the state 'Modified'. The content of the file is a Dockerfile with the following instructions:

```
FROM ubuntu
MAINTAINER repani <qjjjrepani@tip.edu.ph>

ARG DEBIAN_FRONTEND=noninteractive

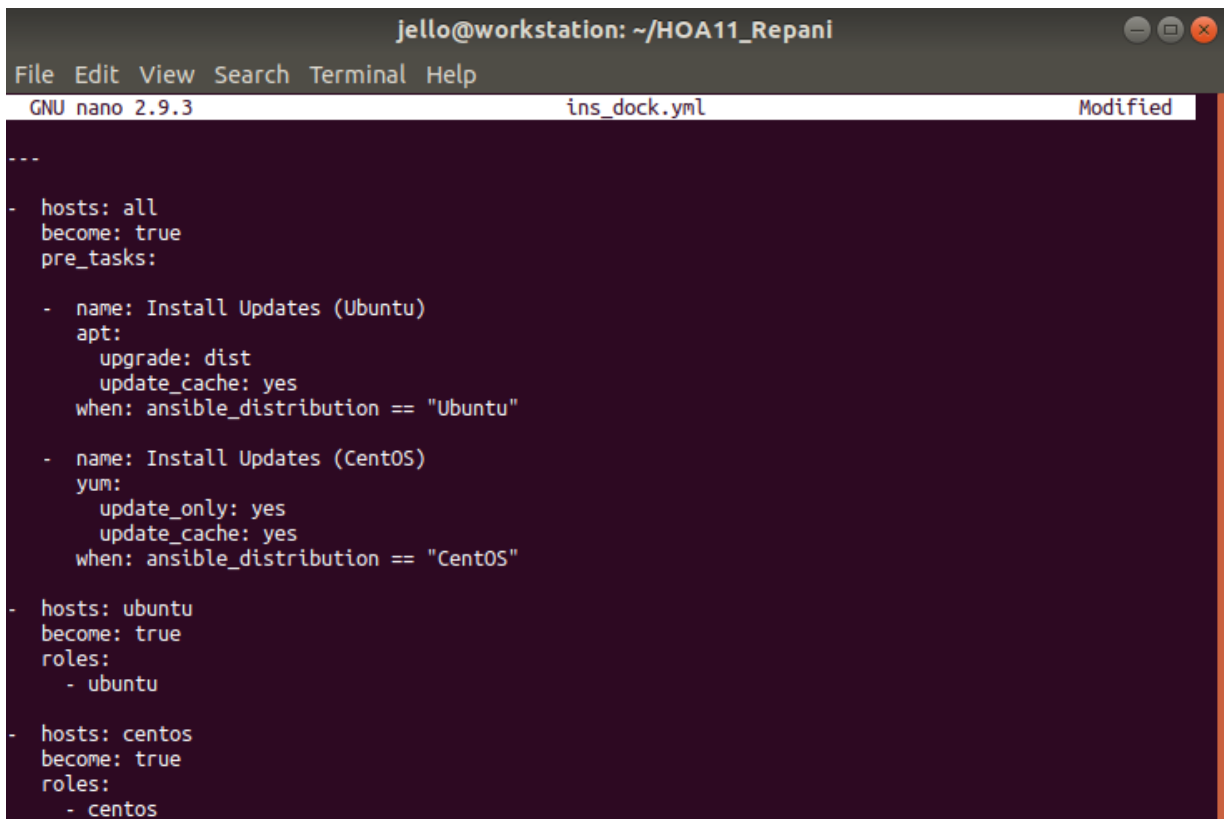
RUN apt-get -y update

RUN apt packages; apt dist-upgrade -y

RUN apt install -y apache2 mariadb-server

ENTRYPOINT apache2ctl -D FOREGROUND
```

ins_dock.yml

A screenshot of a terminal window showing a nano text editor. The window title is 'jello@workstation: ~/HOA11_Repani'. The editor's status bar shows 'GNU nano 2.9.3', the filename 'ins_dock.yml', and the state 'Modified'. The content of the file is an Ansible playbook with the following tasks:

```
---
- hosts: all
  become: true
  pre_tasks:
    - name: Install Updates (Ubuntu)
      apt:
        upgrade: dist
        update_cache: yes
        when: ansible_distribution == "Ubuntu"
    - name: Install Updates (CentOS)
      yum:
        update_only: yes
        update_cache: yes
        when: ansible_distribution == "CentOS"
- hosts: ubuntu
  become: true
  roles:
    - ubuntu
- hosts: centos
  become: true
  roles:
    - centos
```

The install_docker.yml contains all the necessary codes needed in order to make

```
jello@workstation: ~/HOA11_Repani
File Edit View Search Terminal Help
GNU nano 2.9.3 install_docker.yml Modified
---
- 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:
        pip 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: repani
        groups: docker

    - 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"

```

Running the install_docker.yml

```

jello@workstation:~/HOA11_Repani$ ansible-playbook --ask-become-pass install_docker.yml
BECOME password:

PLAY [ubuntu] *****

TASK [Gathering Facts] *****
ok: [192.168.56.102]

TASK [dpkg for Ubuntu] *****
changed: [192.168.56.102]

TASK [Install Docker (Ubuntu)] *****
ok: [192.168.56.102]

TASK [Install SDK (Ubuntu)] *****
changed: [192.168.56.102]

TASK [Ensure Docker group exists (Ubuntu)] *****
ok: [192.168.56.102]

TASK [Adding user to Docker group (Ubuntu)] *****
changed: [192.168.56.102]

TASK [Enable/Restart Docker (Ubuntu)] *****
ok: [192.168.56.102]

TASK [Creating Directory for Dockerfile (Ubuntu)] *****
changed: [192.168.56.102]

TASK [Importing Dockerfile (Ubuntu)] *****
changed: [192.168.56.102]

```

```
PLAY [centos] *****

TASK [Gathering Facts] *****
ok: [192.168.56.104]

TASK [Install required packages (CentOS)] *****
ok: [192.168.56.104]

TASK [Add Docker repository (CentOS)] *****
changed: [192.168.56.104]

TASK [Install Docker (CentOS)] *****
changed: [192.168.56.104]

TASK [Start and enable Docker service (CentOS)] *****
changed: [192.168.56.104]

PLAY RECAP *****
192.168.56.102      : ok=9    changed=5    unreachable=0    failed=0    skipped=0    rescued=0
  ignored=0
192.168.56.104     : ok=5    changed=3    unreachable=0    failed=0    skipped=0    rescued=0
  ignored=0
```

```
jello@workstation:~/HOA11_Repani$ ansible-playbook --ask-become-pass ins_dock.yml
BECOME password:

PLAY [ubuntu] *****

TASK [Gathering Facts] *****
ok: [192.168.56.102]

TASK [ubuntu : Start and Enable Docker in Ubuntu] *****
changed: [192.168.56.102]

TASK [ubuntu : Add User to Docker Group] *****
ok: [192.168.56.102]

TASK [ubuntu : Create directory for Docker in Ubuntu] *****
ok: [192.168.56.102]

TASK [ubuntu : Create dockerfile in Docker Directory in Ubuntu] *****
changed: [192.168.56.102]

TASK [ubuntu : Modify dockerfile with Content in Ubuntu] *****
ok: [192.168.56.102]

TASK [ubuntu : Create container for apache2-mariadb in Ubuntu] *****
changed: [192.168.56.102]

TASK [ubuntu : Run container, apache2-mariadb in Ubuntu] *****
changed: [192.168.56.102]
```

```
PLAY [centos] *****
TASK [Gathering Facts] *****
ok: [192.168.56.104]

TASK [centos : Start and Enable Docker in CentOS] *****
changed: [192.168.56.104]

TASK [centos : Add User to Docker Group] *****
changed: [192.168.56.104]

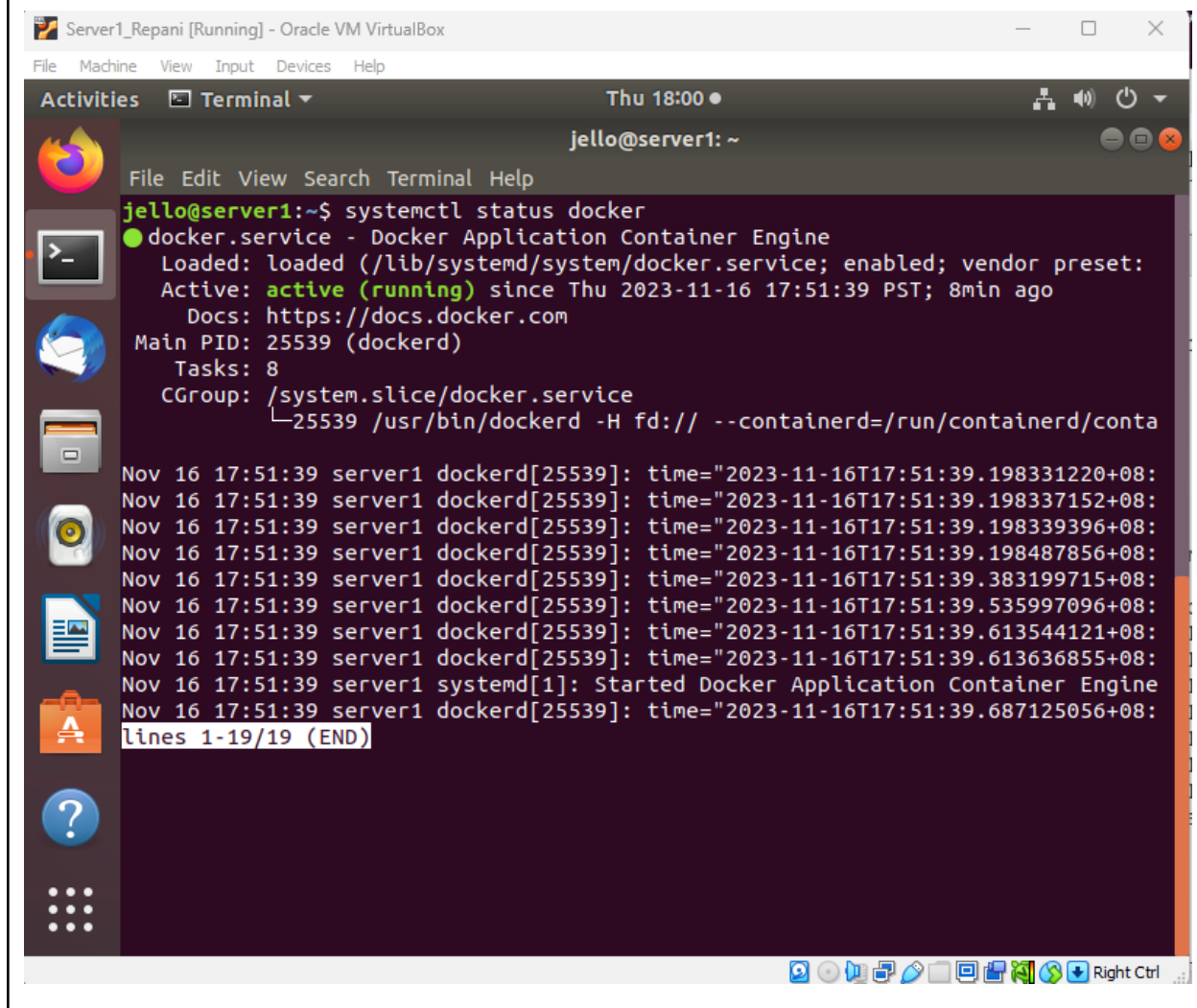
TASK [centos : Create directory for Docker] *****
changed: [192.168.56.104]

TASK [centos : Create dockerfile in Docker Directory] *****
changed: [192.168.56.104]

TASK [centos : Modify dockerfile with Content] *****
changed: [192.168.56.104]

PLAY RECAP *****
192.168.56.102      : ok=8    changed=4    unreachable=0    failed=0    skipped=0    rescued=0
  ignored=0
192.168.56.104      : ok=6    changed=5    unreachable=0    failed=0    skipped=0    rescued=0
  ignored=0
```


Proof:



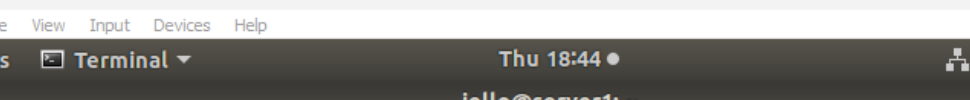
```
Server1_Repani [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Thu 18:00
jello@server1: ~
File Edit View Search Terminal Help
jello@server1:~$ systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset:
   Active: active (running) since Thu 2023-11-16 17:51:39 PST; 8min ago
     Docs: https://docs.docker.com
    Main PID: 25539 (dockerd)
       Tasks: 8
      CGroup: /system.slice/docker.service
              └─25539 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/conta

Nov 16 17:51:39 server1 dockerd[25539]: time="2023-11-16T17:51:39.198331220+08:
Nov 16 17:51:39 server1 dockerd[25539]: time="2023-11-16T17:51:39.198337152+08:
Nov 16 17:51:39 server1 dockerd[25539]: time="2023-11-16T17:51:39.198339396+08:
Nov 16 17:51:39 server1 dockerd[25539]: time="2023-11-16T17:51:39.198487856+08:
Nov 16 17:51:39 server1 dockerd[25539]: time="2023-11-16T17:51:39.383199715+08:
Nov 16 17:51:39 server1 dockerd[25539]: time="2023-11-16T17:51:39.535997096+08:
Nov 16 17:51:39 server1 dockerd[25539]: time="2023-11-16T17:51:39.613544121+08:
Nov 16 17:51:39 server1 dockerd[25539]: time="2023-11-16T17:51:39.613636855+08:
Nov 16 17:51:39 server1 systemd[1]: Started Docker Application Container Engine
Nov 16 17:51:39 server1 dockerd[25539]: time="2023-11-16T17:51:39.687125056+08:
lines 1-19/19 (END)
```

```
CentOS_Repani [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places Terminal Thu 18:00
jello@localhost:~

File Edit View Search Terminal Help
[jello@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 17:57:51 PST; 2min 33s ago
     Docs: https://docs.docker.com
   Main PID: 24467 (dockerd)
      Tasks: 8
     Memory: 25.0M
    CGroup: /system.slice/docker.service
            └─24467 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd...

Nov 16 17:57:47 localhost.localdomain systemd[1]: Starting Docker Application Conta...
Nov 16 17:57:47 localhost.localdomain dockerd[24467]: time="2023-11-16T17:57:47.096..."
Nov 16 17:57:47 localhost.localdomain dockerd[24467]: time="2023-11-16T17:57:47.216..."
Nov 16 17:57:50 localhost.localdomain dockerd[24467]: time="2023-11-16T17:57:50.386..."
Nov 16 17:57:51 localhost.localdomain dockerd[24467]: time="2023-11-16T17:57:51.113..."
Nov 16 17:57:51 localhost.localdomain dockerd[24467]: time="2023-11-16T17:57:51.148..."
Nov 16 17:57:51 localhost.localdomain dockerd[24467]: time="2023-11-16T17:57:51.148..."
Nov 16 17:57:51 localhost.localdomain dockerd[24467]: time="2023-11-16T17:57:51.677..."
Nov 16 17:57:51 localhost.localdomain systemd[1]: Started Docker Application Contai...
Hint: Some lines were ellipsized, use -l to show in full.
[jello@localhost ~]$
```



The screenshot shows a terminal window titled "Server1_Repani [Running] - Oracle VM VirtualBox". The terminal is running on a system with the username "jello". The user has executed the command "sudo visudo" twice, and then "sudo docker images". The output of "docker images" is as follows:

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
apache2-mariadb	latest	b99dfb8a1381	17 minutes ago	260MB
ubuntu	latest	e4c58958181a	6 weeks ago	77.8MB

Sync with Git repository

```
jello@workstation:~/HOA11_Repani$ git add *
jello@workstation:~/HOA11_Repani$ git commit -m "finished"
[master (root-commit) e09f66f] finished
8 files changed, 243 insertions(+)
create mode 100644 ansible.cfg
create mode 100644 dockerfile
create mode 100644 ins_dock.retry
create mode 100644 ins_dock.yml
create mode 100644 install_docker.yml
create mode 100644 inventory
create mode 100644 roles/centos/tasks/main.yml
create mode 100644 roles/ubuntu/tasks/main.yml
jello@workstation:~/HOA11_Repani$ git push origin
Counting objects: 15, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (10/10), done.
Writing objects: 100% (15/15), 2.54 KiB | 2.54 MiB/s, done.
Total 15 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), done.
To github.com:JelzLow/HOA11_Repani.git
* [new branch]      master -> master
```

HOA11_Repani Public

1 branch 0 tags

Go to file Add file <> Code

JelzLow finished e09f66f 2 minutes ago 1 commit

File	Status	Time
roles	finished	2 minutes ago
ansible.cfg	finished	2 minutes ago
dockerfile	finished	2 minutes ago
ins_dock.retry	finished	2 minutes ago
ins_dock.yml	finished	2 minutes ago
install_docker.yml	finished	2 minutes ago
inventory	finished	2 minutes ago

Help people interested in this repository understand your project by adding a README. [Add a README](#)

About Hands on Activity 11 - CPE 232

Activity

0 stars

1 watching

0 forks

Releases

No releases published

[Create a new release](#)

Packages

No packages published

[Publish your first package](#)

https://github.com/JelzLow/HOA11_Repani

Reflections:

Answer the following:

1. What are the benefits of implementing containerizations?

- The benefits of using containerization is that it is more efficient than performing the different tasks traditionally; this is because making use of containerization does not require the user to have an entirely different system.

Conclusions:

This hands-on-activity is quite confusing because of the introduction of docker. With the help of lots of online resources the activity was accomplished with the roles application from the previous lessons as well as the installation of docker and the groups with the different images. Combining these concepts enabled me to accomplish the given task in the activity and achieve the output required.