Name: Kevin Roi A. Sumaya	Date Performed: November 16 2023
Course/Section: CPE 31S3	Date Submitted: November 16 2023
Instructor: Dr. Jonathan Taylar	Semester and 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

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.

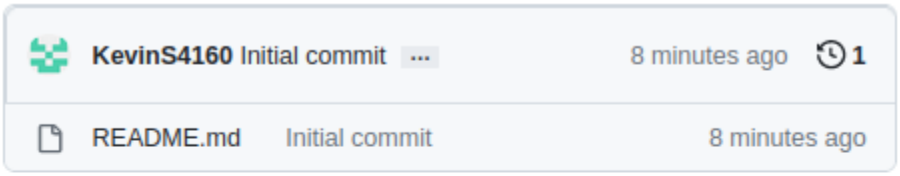
Source: https://docs.docker.com/get-started/overview/

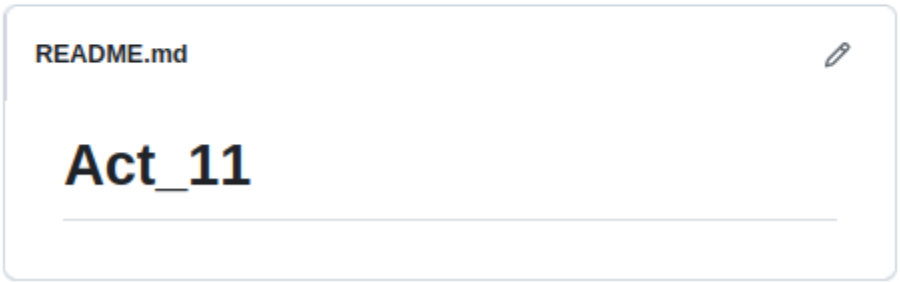
You may also check the difference between containers and virtual machines. Click the link given below.

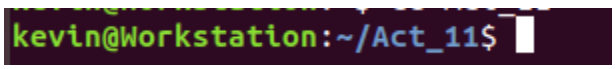
Source: https://docs.microsoft.com/en-us/virtualization/windowscontainers/about/containers-vs-vm

3. Tasks

1. Create a new repository for this activity.







2. Install Docker and enable the docker socket.

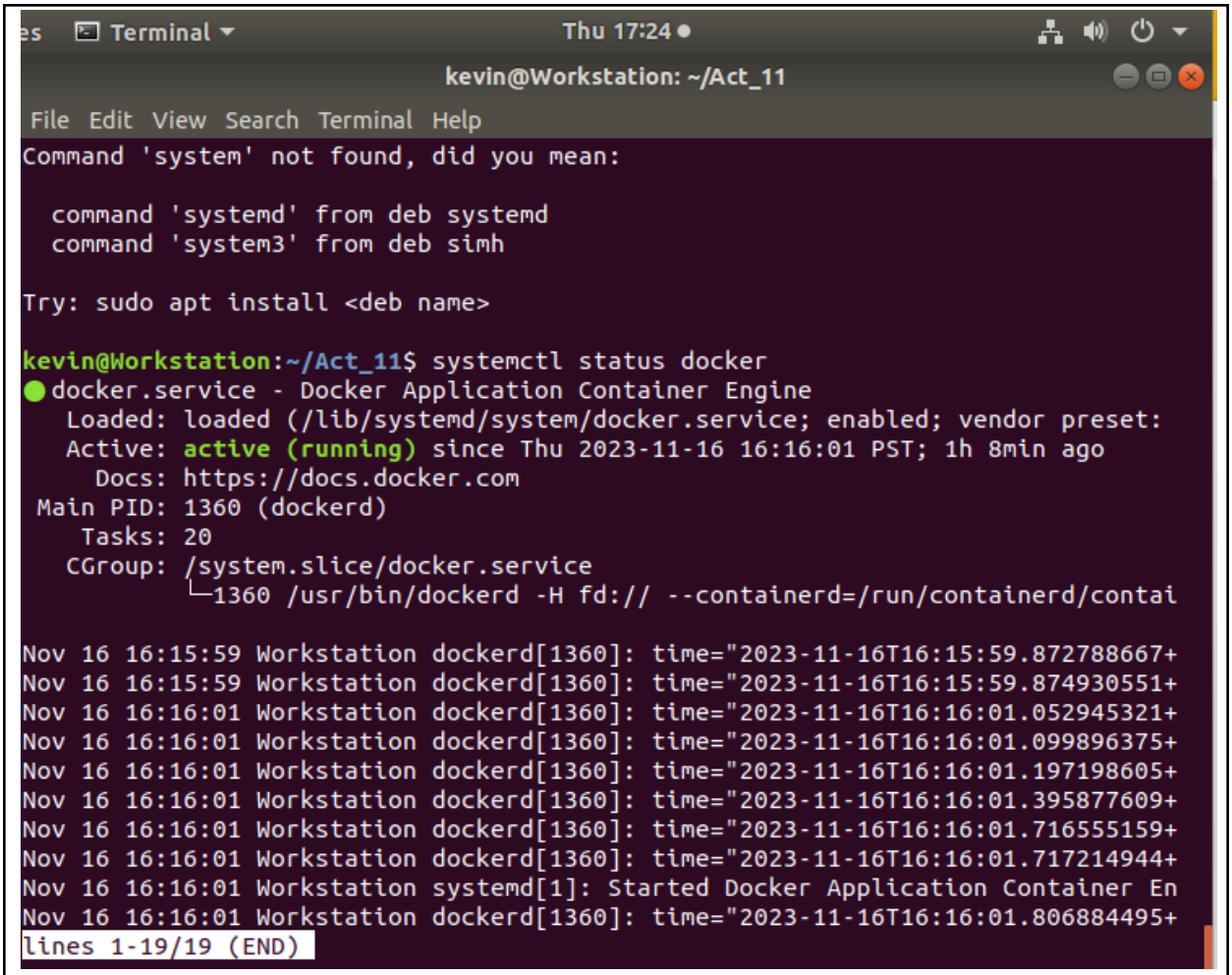
INSTALLING DOCKER

```
kevin@Workstation:~/Act_11$ sudo apt install docker.io
Reading package lists... Done
Building dependency tree
Reading state information... Done
docker.io is already the newest version (20.10.21-0ubuntu1~18.04.3).
The following package was automatically installed and is no longer required:
  libllvm7
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
kevin@Workstation:~/Act_11$
```

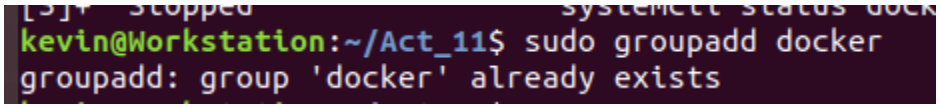
```
kevin@Workstation:~/Act_11$ sudo apt upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following package was automatically installed and is no longer required:
  libllvm7
Use 'sudo apt autoremove' to remove it.
The following security updates require Ubuntu Pro with 'esm-infra' enabled:
  libwebp6 libkrb5-3 libgssapi-krb5-2 libpython3.6-minimal poppler-utils
  libnghttp2-14 libisccfg160 libcups2 intel-microcode linux-libc-dev
  xserver-common vim-common libldap-2.4-2 openssl libdw1 libpython3.6-dev
  imagemagick libsystemd0 python-requests libavahi-glib1 libgs9
  python2.7-minimal libpam-cap libpython3.6-stdlib libelf1 python3-urllib3
  binutils libmagickwand-6.q16-3 libirs160 bind9-host
  linux-headers-generic-hwe-18.04 libavahi-common-data dnsutils
  libavahi-common3 libpython2.7 libncurses5 python2.7
  gir1.2-accountsservice-1.0 libpython3.6 python3.6 libwinpr2-2 libyajl2
  libk5crypto3 libisc169 udev cups-server-common procps amd64-microcode
  cups-common libncursesw5 libprocps6 libx11-6 python3-requests libudev1
  libvpx5 libapparmor1 libwebpdemux2 krb5-locales libavahi-ui-gtk3-0
  python3.6-minimal binutils-x86-64-linux-gnu xserver-xephyr imagemagick-6.q16
  libtiff5 libfreerdp2-2 libisc-export169 busybox-static cups-ppdc
  libcupsmime1 libtinfo5 libkrb5support0 avahi-daemon libnss-myhostname
  libcap2 systemd-sysv libcap2-bin libldap-common libavahi-core7 liblwres160
  linux-image-generic-hwe-18.04 libglib2.0-bin libpam-systemd xwayland
```

ENABLING DOCKER

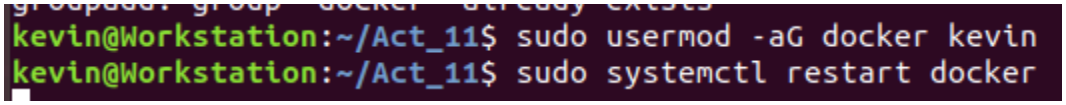
```
kevin@Workstation:~/Act_11$ systemctl enable docker
```



3. Add to Docker group to your current user.



ADDING USER TO THE GROUP AND RESTARTING DOCKER



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

```
  GNU nano 2.9.3                          dockerfile

FROM ubuntu
MAINTAINER kevin <qkrasumaya@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
```

5. Install and build the Dockerfile using Ansible.

```
kevin@Workstation: ~/Act_11
File  Edit  View  Search  Terminal  Help
  GNU nano 2.9.3                    dockerfile.yml

- hosts: web_servers
  become: true
  pre_tasks:

    - name: dpkg for Ubuntu
      shell:
        dpkg --configure -a
      when: ansible_distribution == "Ubuntu"

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

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

    - name: Adding group to Docker
      shell:
        usermod -aG docker kevin

                              [ Read 77 lines ]
```

```
 GNU nano 2.9.3                    dockerfile.yml

      service:
        name: docker
        state: started
        enabled: true

    - name: Creating Directory for Dockerfile
      file:
        path: ./root/demo-dockerfile
        state: directory
        owner: root
        group: root
        mode: '0755'

    - name: Importing of Dockerfile
      copy:
        src: ./dockerfile
        dest: ./root/demo-dockerfile/dockerfile
        owner: root
        group: root
        mode: '0755'

- hosts: db_servers
  become: true
```

```
 GNU nano 2.9.3                    dockerfile.yml


    - name: Install required packages
      yum:
        name:
          - yum-utils
          - device-mapper-persistent-data
          - lvm2
        state: present

    - name: Add Docker repository
      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

    - name: Install Docker
      yum:
        name: docker-ce
        state: present

    - name: Start and enable Docker service
```

```
    - name: Install Docker
      yum:
        name: docker-ce
        state: present

    - name: Start and enable Docker service
      systemd:
        name: docker
        state: started
        enabled: yes
```

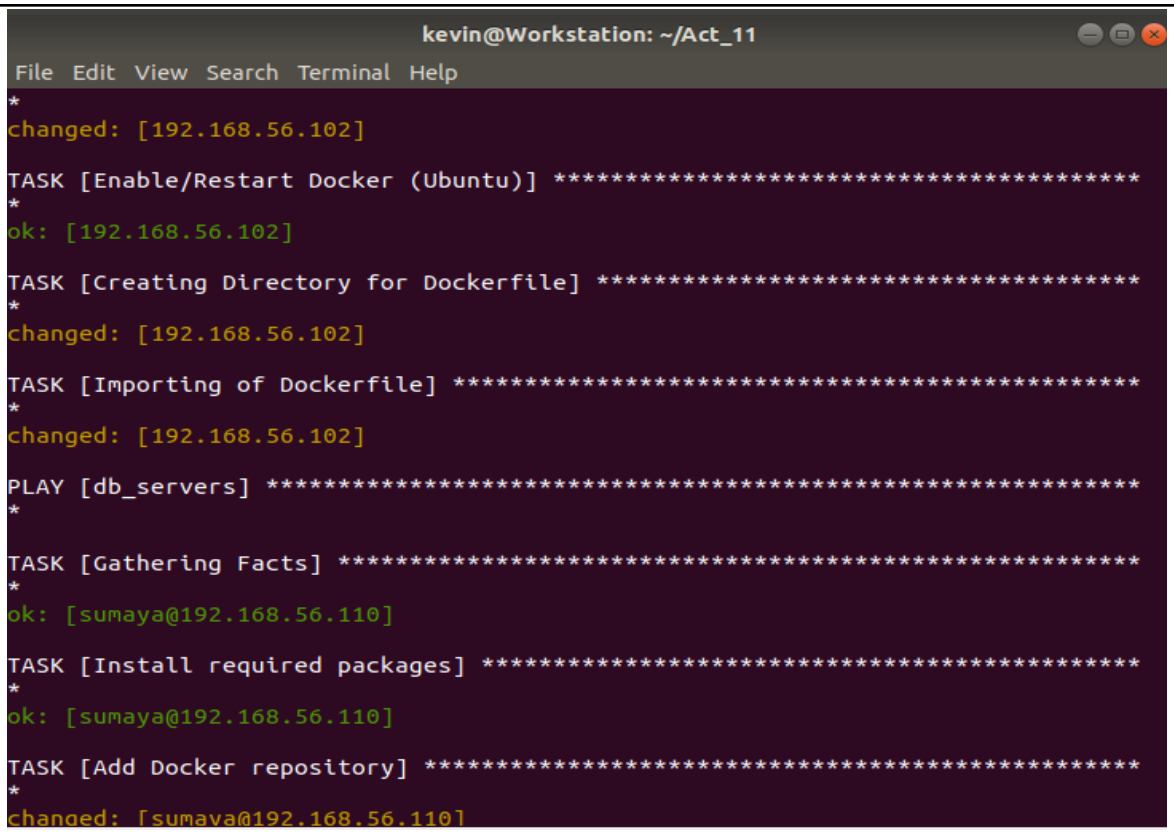
6. Add, commit and push it to your repository.

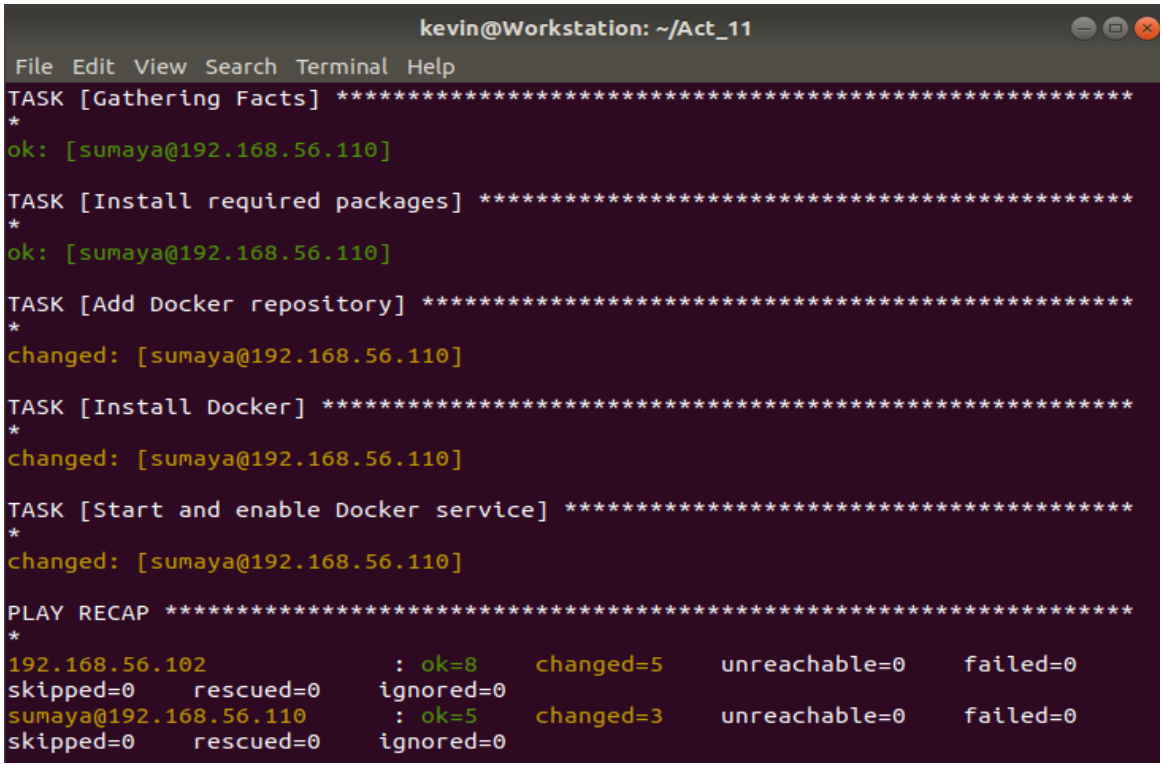
```
kevin@Workstation:~/Act_11$ git push origin
Username for 'https://github.com': KevinS4160
Password for 'https://KevinS4160@github.com':
Counting objects: 6, done.
Delta compression using up to 6 threads.
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 1.26 KiB | 1.26 MiB/s, done.
Total 6 (delta 0), reused 0 (delta 0)
To https://github.com/KevinS4160/Act_11.git
   282cf30..7a7f23e  main -> main
```

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

Final Outputs:

```
kevin@Workstation:~/Act_11$ ansible-playbook --ask-become-pass dockerfile.yml
BECOME password:

PLAY [web_servers] *************************************************************
*

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]
```


kevin@Workstation: ~/Act_11
File Edit View Search Terminal Help
*
changed: [192.168.56.102]
TASK [Enable/Restart Docker (Ubuntu)] **
*
ok: [192.168.56.102]
TASK [Creating Directory for Dockerfile] *************************************
*
changed: [192.168.56.102]
TASK [Importing of Dockerfile] ***
*
changed: [192.168.56.102]
PLAY [db_servers] **
*
TASK [Gathering Facts] ***
*
ok: [sumaya@192.168.56.110]
TASK [Install required packages] ***
*
ok: [sumaya@192.168.56.110]
TASK [Add Docker repository] ***
*
changed: [sumaya@192.168.56.110]


kevin@Workstation: ~/Act_11
File Edit View Search Terminal Help
TASK [Gathering Facts] ***
*
ok: [sumaya@192.168.56.110]
TASK [Install required packages] ***
*
ok: [sumaya@192.168.56.110]
TASK [Add Docker repository] ***
*
changed: [sumaya@192.168.56.110]
TASK [Install Docker] **
*
changed: [sumaya@192.168.56.110]
TASK [Start and enable Docker service] ***************************************
*
changed: [sumaya@192.168.56.110]
PLAY RECAP ***
*
192.168.56.102 : ok=8 changed=5 unreachable=0 failed=0
skipped=0 rescued=0 ignored=0
sumaya@192.168.56.110 : ok=5 changed=3 unreachable=0 failed=0
skipped=0 rescued=0 ignored=0

```
kevin@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 18:10:43 PST; 12min ago
     Docs: https://docs.docker.com
 Main PID: 19630 (dockerd)
    Tasks: 11
   CGroup: /system.slice/docker.service
           └─19630 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/conta

Nov 16 18:10:34 server1 dockerd[19630]: time="2023-11-16T18:10:34.438083813+08:
Nov 16 18:10:34 server1 dockerd[19630]: time="2023-11-16T18:10:34.438111572+08:
Nov 16 18:10:34 server1 dockerd[19630]: time="2023-11-16T18:10:34.438626710+08:
Nov 16 18:10:35 server1 dockerd[19630]: time="2023-11-16T18:10:35.573181435+08:
Nov 16 18:10:38 server1 dockerd[19630]: time="2023-11-16T18:10:38.706456367+08:
Nov 16 18:10:42 server1 dockerd[19630]: time="2023-11-16T18:10:42.381983048+08:
Nov 16 18:10:43 server1 dockerd[19630]: time="2023-11-16T18:10:43.046114003+08:
Nov 16 18:10:43 server1 dockerd[19630]: time="2023-11-16T18:10:43.046233619+08:
Nov 16 18:10:43 server1 dockerd[19630]: time="2023-11-16T18:10:43.096881098+08:
Nov 16 18:10:43 server1 systemd[1]: Started Docker Application Container Engine
lines 1-19/19 (END)
```

```
[sumaya@localhost ~]$ systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; vendor preset: disa
bled)
   Active: active (running) since Thu 2023-11-16 05:16:53 EST; 6min ago
     Docs: https://docs.docker.com
 Main PID: 3348 (dockerd)
    Tasks: 10
   Memory: 30.4M
   CGroup: /system.slice/docker.service
           └─3348 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd....

Nov 16 05:16:38 localhost.localdomain systemd[1]: Starting Docker Application Conta....
Nov 16 05:16:38 localhost.localdomain dockerd[3348]: time="2023-11-16T05:16:38.6094..."
Nov 16 05:16:40 localhost.localdomain dockerd[3348]: time="2023-11-16T05:16:40.6259..."
Nov 16 05:16:52 localhost.localdomain dockerd[3348]: time="2023-11-16T05:16:52.3660..."
Nov 16 05:16:52 localhost.localdomain dockerd[3348]: time="2023-11-16T05:16:52.7703..."
Nov 16 05:16:52 localhost.localdomain dockerd[3348]: time="2023-11-16T05:16:52.9724...7
Nov 16 05:16:52 localhost.localdomain dockerd[3348]: time="2023-11-16T05:16:52.9727..."
Nov 16 05:16:53 localhost.localdomain dockerd[3348]: time="2023-11-16T05:16:53.1078..."
Nov 16 05:16:53 localhost.localdomain systemd[1]: Started Docker Application Contai....
Hint: Some lines were ellipsized, use -l to show in full.
[sumaya@localhost ~]$
```

- This is the proof that both server 1 and CentOS maridb and docker are working.

Inventory

Code Blame 5 lines (4 loc) · 65 Bytes Code 55% faster with GitHub Copilot

```
1  [web_servers]
2  192.168.56.102
3
4  [db_servers]
5  sumaya@192.168.56.110
```

- This is the inventory where I placed the ipaddress of the Server 1 and CentOS for the dockerfile installation path to download and install the code in it.

Ansible.cfg

KevinS4160 Latest

Code Blame 9 lines (6 loc) · 137 Bytes Code 55% faster with GitHub Copilot

```
1  [defaults]
2
3  inventory = inventory
4  host_key_checking = False
5
6  deprecation_warnings = False
7
8  remote_user = kevin
9  private_key_file = ~/.ssh
```

- This is the ansible.cfg to know where and what workstation to be able to access when downloading the specific code.

dockerfile

KevinS4160 Latest

Code | Blame | 12 lines (7 loc) · 231 Bytes | Code 55% faster with GitHub Copilot

```
1   FROM ubuntu
2   MAINTAINER kevin <qkrasumaya@tip.edu.ph>
3
4   ARG DEBIAN_FRONTEND=noninteractive
5
6   RUN apt-get -y update
7
8   RUN apt packages; apt dist-upgrade -y
9
10  RUN apt install -y apache2 mariadb-server
11
12  ENTRYPOINT apache2ctl -D FOREGROUND
```

- This is the code to be able to run the specific code of the mariadb-server, apt packages, etc.

dockerfile.yml

Code Blame 77 lines (63 loc) · 1.64 KB Code 55% faster with Git

```
- hosts: web_servers
  become: true
  pre_tasks:

    - name: dpkg for Ubuntu
      shell:
        dpkg --configure -a
      when: ansible_distribution == "Ubuntu"

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

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

    - name: Adding group to Docker
      shell:
        usermod -aG docker kevin

    - name: Enable/Restart Docker (Ubuntu)
      service:
        name: docker
        state: started
        enabled: true

    - name: Creating Directory for Dockerfile
      file:
        path: ./root/demo-dockerfile
        state: directory
        owner: root
        group: root
        mode: '0755'

    - name: Importing of Dockerfile
      copy:
        src: ./dockerfile
        dest: ./root/demo-dockerfile/dockerfile
        owner: root
        group: root
```

```
44            mode: '0755'
45
46    - hosts: db_servers
47      become: true
48      pre_tasks:
49
50        - name: Install required packages
51          yum:
52            name:
53              - yum-utils
54              - device-mapper-persistent-data
55              - lvm2
56            state: present
57
58        - name: Add Docker repository
59          yum_repository:
60            name: docker-ce
61            description: Docker CE Stable - $basearch
62            baseurl: https://download.docker.com/linux/centos/7/$basearch/stable
63            gpgkey: https://download.docker.com/linux/centos/gpg
64            enabled: yes
65
66        - name: Install Docker
67          yum:
68            name: docker-ce
69            state: present
70
71        - name: Start and enable Docker service
72          systemd:
73            name: docker
74            state: started
75            enabled: yes
```

- This is the specific code to be able to install and download the mariadb-server and dockerfile for both Server 1 and CENTOS.

GIT REPOSITORY LINK

https://github.com/KevinS4160/Act_11.git

Reflections:

Answer the following:

1. What are the benefits of implementing containerizations?
    - Containerization is a technique that allows applications to run in isolated environments, called containers, on a shared operating system. It benefits through Portability, Efficiency, Agility, Security and many more that help you optimize your application development and delivery process.

Conclusions:

- In doing this activity I learned how to install docker through ansible-playbook. When doing the activity I learned many debugging techniques to fix my commands on installing the playbook.