

Name: Gabriel Soriano	Date Performed: November 17, 2022
Course/Section:	Date Submitted: November 23, 2022
Instructor: Engr. Taylar	Semester and SY: 1st sem - SY 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	
<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. DONE 2. Install Docker and enable the docker socket. DONE 3. Add to Docker group to your current user. DONE 4. Create a Dockerfile to install web and DB server. DONE 5. Install and build the Dockerfile using Ansible. DONE 6. Add, commit and push it to your repository. DONE 	
4. Output (screenshots and explanations)	

main

Go to file

Add file

Code

About



qgsoriano1 Initial commit

5 minutes ago 1



README.md

Initial commit

5 minutes ago

README.md



HOA11_Soriano

No description, website, or topics provided.

Readme

0 stars

1 watching

0 forks

Releases

No releases published

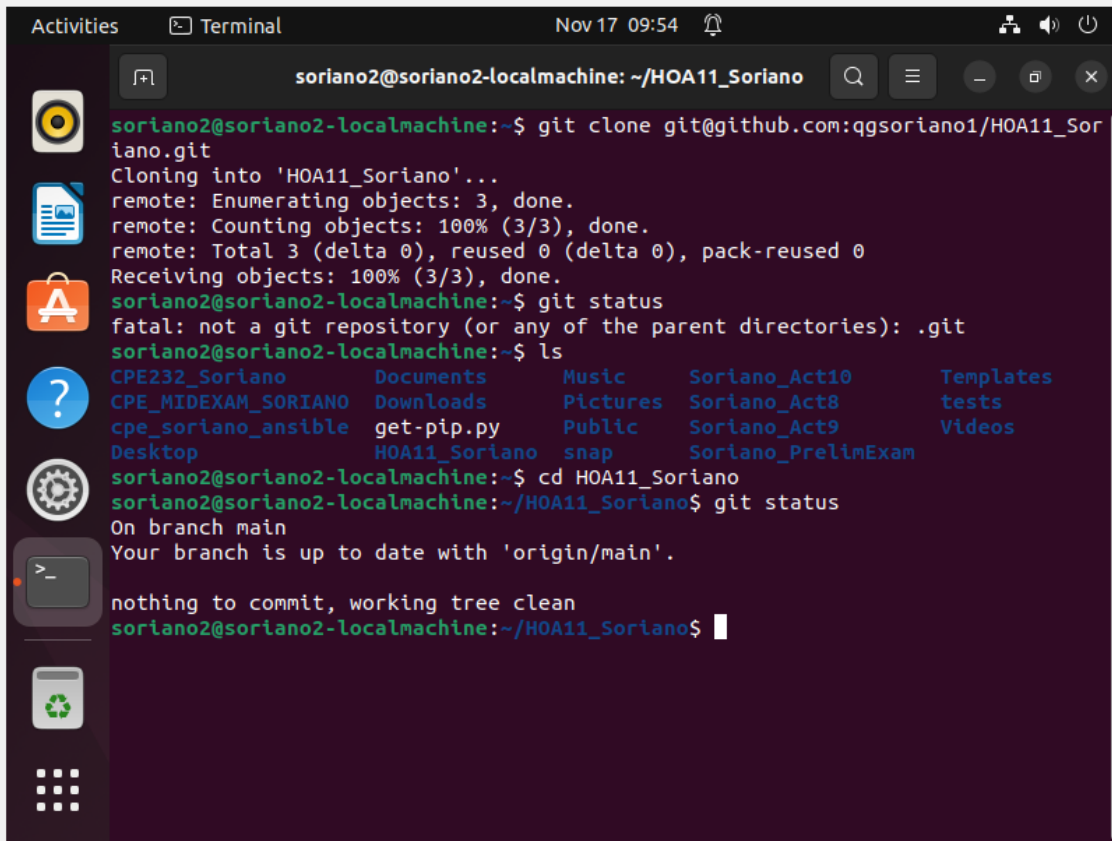
[Create a new release](#)

Packages

No packages published

[Publish your first package](#)

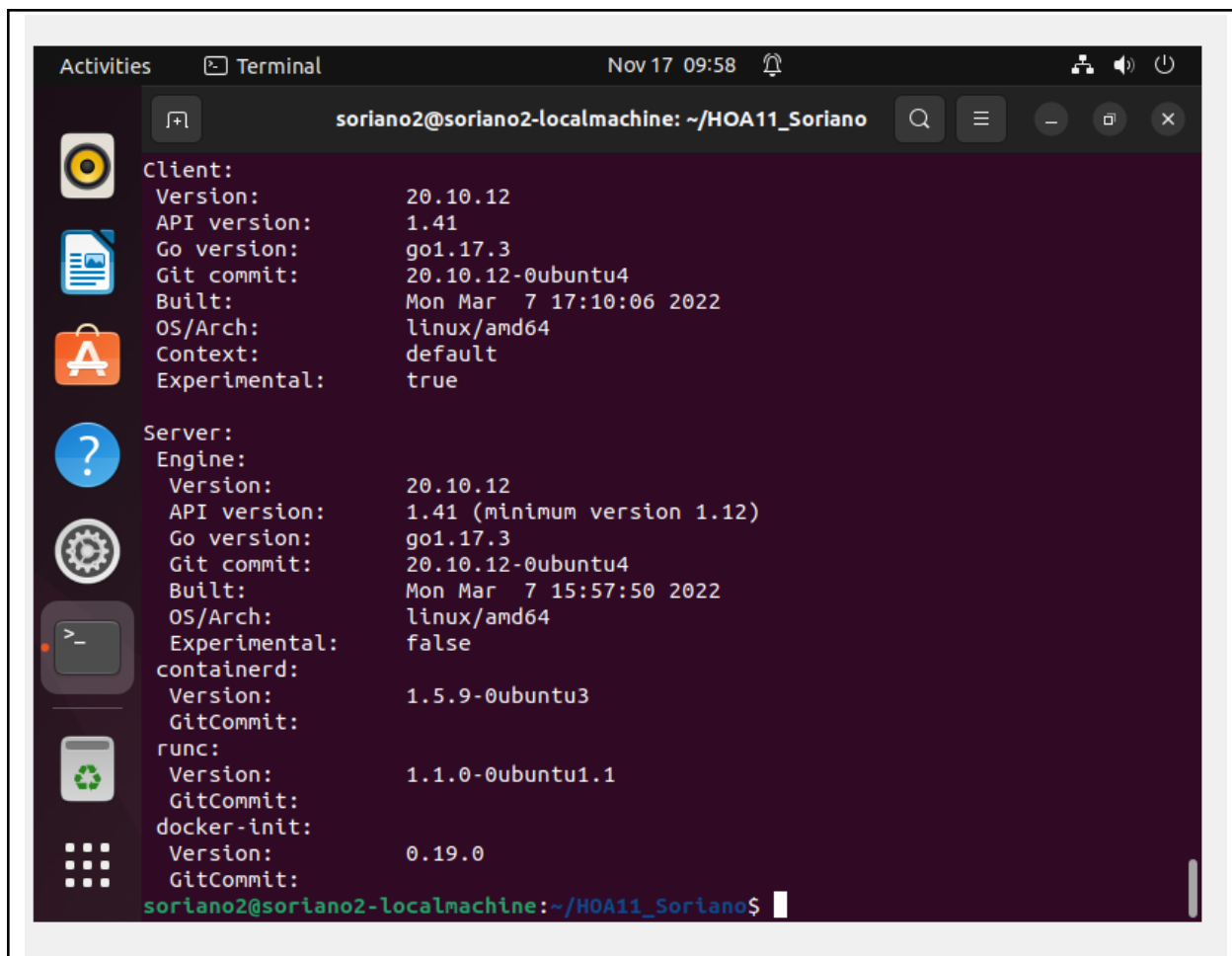


A terminal window titled 'Terminal' with a date and time of 'Nov 17 09:54'. The window shows a user 'soriano2' at a machine 'soriano2-localmachine' in the directory '~/HOA11_Soriano'. The user runs 'git clone git@github.com:qgsoriano1/HOA11_Soriano.git', which clones a repository. Then, they run 'git status', which shows a fatal error because it's not a git repository. They run 'ls' to list files, showing a directory structure with various folders and files. Finally, they run 'cd HOA11_Soriano' and 'git status' again, which shows they are on the 'main' branch and the working tree is clean.

```
soriano2@soriano2-localmachine: ~$ git clone git@github.com:qgsoriano1/HOA11_Soriano.git
Cloning into 'HOA11_Soriano'...
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.
soriano2@soriano2-localmachine:~$ git status
fatal: not a git repository (or any of the parent directories): .git
soriano2@soriano2-localmachine:~$ ls
CPE232_Soriano      Documents      Music          Soriano_Act10     Templates
CPE_MIDEXAM_SORIANO Downloads      Pictures       Soriano_Act8       tests
cpe_soriano_ansible get-pip.py     Public         Soriano_Act9       Videos
Desktop             HOA11_Soriano snap           Soriano_PrelimExam
soriano2@soriano2-localmachine:~$ cd HOA11_Soriano
soriano2@soriano2-localmachine:~/HOA11_Soriano$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
soriano2@soriano2-localmachine:~/HOA11_Soriano$
```

- This shows the creation of a new repository in github site, this is named HOA11_Soriano, and is successfully connected to the local machine.

A terminal window titled 'Terminal' with a dark background and light text. The window shows the output of the 'docker version' command. The output is divided into two sections: 'Client:' and 'Server:'. The 'Client:' section lists version 20.10.12, API version 1.41, Go version go1.17.3, Git commit 20.10.12-0ubuntu4, build date Mon Mar 7 17:10:06 2022, OS/Arch linux/amd64, context default, and experimental true. The 'Server:' section lists engine version 20.10.12, API version 1.41 (minimum version 1.12), Go version go1.17.3, Git commit 20.10.12-0ubuntu4, build date Mon Mar 7 15:57:50 2022, OS/Arch linux/amd64, experimental false, containerd version 1.5.9-0ubuntu3, runc version 1.1.0-0ubuntu1.1, and docker-init version 0.19.0. The prompt at the bottom is 'soriano2@soriano2-localmachine: ~/HOA11_Soriano\$'.

```
Client:
Version:      20.10.12
API version:  1.41
Go version:   go1.17.3
Git commit:   20.10.12-0ubuntu4
Built:        Mon Mar  7 17:10:06 2022
OS/Arch:      linux/amd64
Context:      default
Experimental: true

Server:
Engine:
Version:      20.10.12
API version:  1.41 (minimum version 1.12)
Go version:   go1.17.3
Git commit:   20.10.12-0ubuntu4
Built:        Mon Mar  7 15:57:50 2022
OS/Arch:      linux/amd64
Experimental: false
containerd:
Version:      1.5.9-0ubuntu3
GitCommit:
runc:
Version:      1.1.0-0ubuntu1.1
GitCommit:
docker-init:
Version:      0.19.0
GitCommit:
soriano2@soriano2-localmachine: ~/HOA11_Soriano$
```

- This shows the successful installation of docker in the local machine

```
Activities Terminal Nov 17 09:59
soriano2@soriano2-localmachine: ~/HOA11_Soriano

Version: 0.19.0
GitCommit:
soriano2@soriano2-localmachine:~/HOA11_Soriano$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor prese
   Active: active (running) since Thu 2022-11-17 08:54:38 PST; 1h 4min ago
   TriggeredBy: ● docker.socket
     Docs: https://docs.docker.com
    Main PID: 37611 (dockerd)
       Tasks: 10
      Memory: 57.6M
         CPU: 13.111s
    CGroup: /system.slice/docker.service
            └─37611 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/co>

Nov 17 09:18:31 soriano2-localmachine dockerd[37611]: time="2022-11-17T09:18:3
Nov 17 09:18:35 soriano2-localmachine dockerd[37611]: time="2022-11-17T09:18:3
Nov 17 09:20:57 soriano2-localmachine dockerd[37611]: time="2022-11-17T09:20:5
Nov 17 09:28:53 soriano2-localmachine dockerd[37611]: time="2022-11-17T09:28:5
Nov 17 09:29:06 soriano2-localmachine dockerd[37611]: time="2022-11-17T09:29:0
Nov 17 09:29:13 soriano2-localmachine dockerd[37611]: time="2022-11-17T09:29:1
Nov 17 09:29:20 soriano2-localmachine dockerd[37611]: time="2022-11-17T09:29:2
Nov 17 09:29:28 soriano2-localmachine dockerd[37611]: time="2022-11-17T09:29:2
Nov 17 09:30:17 soriano2-localmachine dockerd[37611]: time="2022-11-17T09:30:1
Nov 17 09:30:17 soriano2-localmachine dockerd[37611]: time="2022-11-17T09:30:1

soriano2@soriano2-localmachine:~/HOA11_Soriano$ sudo systemctl enable docker
soriano2@soriano2-localmachine:~/HOA11_Soriano$ sudo systemctl start docker
soriano2@soriano2-localmachine:~/HOA11_Soriano$
```

- This shows the enabling of the docker in the local machine.

```
Activities  Terminal  Nov 17 10:28  [icons]

soriano2@soriano2-localmachine: ~/HOA11_Soriano

- name: Create default containers
  community.docker.docker_container:
    name: "{{ default_container_name }}"
    image: "{{ default_container_image }}"
    command: "{{ default_container_command }}"
    state: present
  with_sequence: count={{ container_count }}

soriano2@soriano2-localmachine:~/test1$ cd
soriano2@soriano2-localmachine:~$ ls
CPE232_Soriano  Documents  Music  Soriano_Act10  Templates
CPE_MIDEXAM_SORIANO  Downloads  Pictures  Soriano_Act8  test1
cpe_soriano_ansible  get-pip.py  Public  Soriano_Act9  tests
Desktop  HOA11_Soriano  snap  Soriano_PrelimExam  Videos

soriano2@soriano2-localmachine:~$ cd HOA11_Soriano
soriano2@soriano2-localmachine:~/HOA11_Soriano$ ls
ansible.cfg  Dockerfile  docker.yml  inventory  README.md
soriano2@soriano2-localmachine:~/HOA11_Soriano$ cd docker.yml
bash: cd: docker.yml: Not a directory
soriano2@soriano2-localmachine:~/HOA11_Soriano$ nano docker.yml
soriano2@soriano2-localmachine:~/HOA11_Soriano$ ansible -m ping all
192.168.56.105 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
soriano2@soriano2-localmachine:~/HOA11_Soriano$
```

- This shows the successful connection between the workstation and the remote server

```
soriano2@soriano2-localmachine: ~/HOA11_Soriano
*** System restart required ***
Last login: Thu Nov 17 10:26:24 2022 from 192.168.56.1
ls
soriano2@soriano2-localmachine:~$ ls
CPE232_Soriano    Documents    Music       Soriano_Act10  Templates
CPE_MIDEXAM_SORIANO Downloads    Pictures    Soriano_Act8   test1
cpe_soriano_ansible get-pip.py   Public      Soriano_Act9   tests
Desktop           HOA11_Soriano snap        Soriano_PrelimExam Videos
soriano2@soriano2-localmachine:~$ cd HOA11_Soriano
soriano2@soriano2-localmachine:~/HOA11_Soriano$ ls
ansible.cfg  apache_folder Dockerfile  docker.yml  inventory  README.md
soriano2@soriano2-localmachine:~/HOA11_Soriano$ cat inventory
[workstation]
localhost ansible_connection=local
soriano2@soriano2-localmachine:~/HOA11_Soriano$ cat docker.yml
---
- hosts: all
  become: true
  tasks:
    - name: Install and build dockerfile
      command: docker build -t apache2/mariadb .
      when: ansible_distribution == "Ubuntu"
soriano2@soriano2-localmachine:~/HOA11_Soriano$
```

```
soriano2@soriano2-localmachine: ~/HOA11_Soriano
- hosts: all
  become: true
  tasks:
    - name: Install and build dockerfile
      command: docker build -t apache2/mariadb .
      when: ansible_distribution == "Ubuntu"
soriano2@soriano2-localmachine:~/HOA11_Soriano$ ls
ansible.cfg  apache_folder Dockerfile  docker.yml  inventory  README.md
soriano2@soriano2-localmachine:~/HOA11_Soriano$ cat Dockerfile
FROM ubuntu
MAINTAINER soriano2 <qgsoriano1@tip.edu.ph>

# skip prompt
ARG DEBIAN_FRONTEND=noninteractive

# update packages
RUN apt update; apt dist-upgrade -y

# install apache
RUN apt install -y apache2

# set entrypoint
ENTRYPOINT apache2ctl -D FOREGROUND
soriano2@soriano2-localmachine:~/HOA11_Soriano$
```

- This shows the content of the inventory, docker.yml, and the dockerfile

```
Activities Terminal Nov 17 11:19 soriano2@soriano2-localmachine: ~/HOA11_Soriano

PLAY RECAP *****
*
localhost : ok=1 changed=0 unreachable=0 failed=1
skipped=0 rescued=0 ignored=0

soriano2@soriano2-localmachine:~/HOA11_Soriano$ ls
ansible.cfg apache_folder Dockerfile docker.yml inventory README.md
soriano2@soriano2-localmachine:~/HOA11_Soriano$ nano Dockerfile
soriano2@soriano2-localmachine:~/HOA11_Soriano$ ansible-playbook --ask-become-p
ass docker.yml
BECOME password:

PLAY [all] *****
*

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

TASK [Install and build dockerfile] *****
*
changed: [localhost]

PLAY RECAP *****
*
localhost : ok=2 changed=1 unreachable=0 failed=0
skipped=0 rescued=0 ignored=0

soriano2@soriano2-localmachine:~/HOA11_Soriano$
```

- This shows the successful run of the ansible file named docker.yml

qgsoriano1 / HOA11_SorianoPublic

Pin

Unwatch1

Fork0

Star0

<> Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

...

main

Go to file

Add file

<> Code

About

qgsoriano1 finished at 11:30am nov 17, 202233 seconds ago2

apache_folder	finished at 11:30am nov 17, 2022	33 seconds ago
Dockerfile	finished at 11:30am nov 17, 2022	33 seconds ago
README.md	Initial commit	2 hours ago
ansible.cfg	finished at 11:30am nov 17, 2022	33 seconds ago
docker.yml	finished at 11:30am nov 17, 2022	33 seconds ago
inventory	finished at 11:30am nov 17, 2022	33 seconds ago

README.md

HOA11_Soriano

No description, website, or topics provided.

Readme

0 stars

1 watching

0 forks

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

TermsPrivacySecurityStatusDocsContact GitHubPricingAPITrainingBlogAbout

-

- This shows the successful committing of all of the files that has been performed in this activity, including the ansible file and the dockerfile



Apache2 Debian Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Debian's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Debian tools. The configuration system is **fully documented in [/usr/share/doc/apache2/README.Debian.gz](#)**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Debian systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|   |-- ports.conf
|-- mods-enabled
|   |-- *.load
|   |-- *.conf
|-- conf-enabled
|   |-- *.conf
|-- sites-enabled
|   |-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective `*-available/` counterparts. These should be managed by using our helpers `a2enmod`, `a2dismod`, `a2ensite`, `a2dissite`, and `a2enconf`, `a2disconf`. See their respective man pages for detailed information.
- The binary is called `apache2`. Due to the use of environment variables, in the default configuration, `apache2` needs to be started/stopped with `/etc/init.d/apache2` or `apache2ctl`. **Calling `/usr/bin/apache2` directly will not work** with the default configuration.

- This shows the proof of installation of the apache on the remote server

Reflections:

Answer the following:

1. What are the benefits of implementing containerizations?

- It is portable, you can take your application almost anywhere without having to rebuild it to take into account a different environment because a container bundles all dependencies. Additionally, containerization's abstraction guarantees that your container will function the same wherever you deploy it. This means that you can run your app on bare metal, in a virtual machine, or in the cloud. You can deploy quickly as long as the host operating system is compatible with your containerization tools. It is also efficient, One of the most effective virtualization techniques for developers is containerization. Containers increase productivity in two ways: by utilizing all resources and reducing overhead. Containers enable a host to utilize nearly all of the resources available when set up properly. A single host can carry out numerous tasks thanks to isolated containers, which can operate independently of other containers. There are other more advantages in using containers and all of the statements above are just one of them.

Conclusions:

- **While performing this activity, there are times that I forget some of the codes and commands in the ansible part. Honestly, this activity is new to us because docker was just introduced to some of us, and I'm not familiar at all with this platform. Docker is a platform that offers service products and softwares in packages that are known as containers. The service offers both free and paid tiers. Docker is basically a collection of platforms that delivers software in packages that are called containers that are used in multiple platforms, and we have used this in Linux, Ubuntu Operating system. By this, we were able to run multiple installation commands in just one go. It is very similar to ansible, but the process is way shorter when compared to ansible. By the end of this activity, the creation of the requirement, which is the dockerfile, is completed and successful. By this, we were able to install apache, web and db servers on a remote server, which is also a linux Ubuntu Operating system. Proofs are pasted above. Thank you.**