Name: De Omampo, Julius Mark A.	**Date Performed:** 11/24/24
Course/Section: CPE212 – CPE31S2	**Date Submitted:** 11/25/24
Instructor: Engr. Robin Valenzuela	**Semester and SY:** 1st Sem. (2024-2025)

Activity 12: Build a Sample Web App in a Docker Container

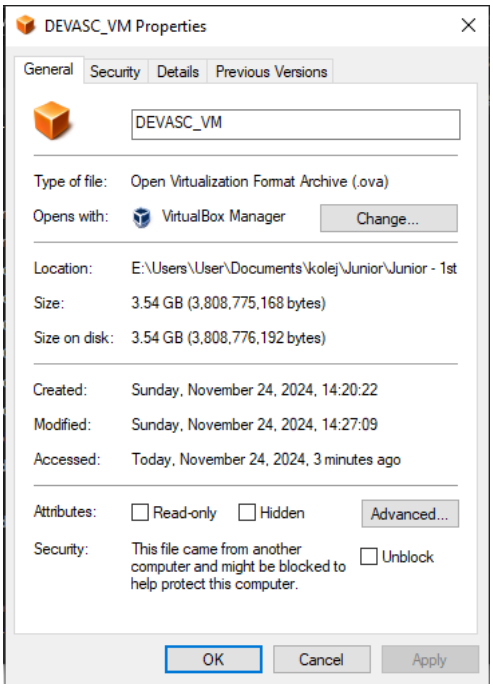
1. Tasks

Instructions before you begin this activity:

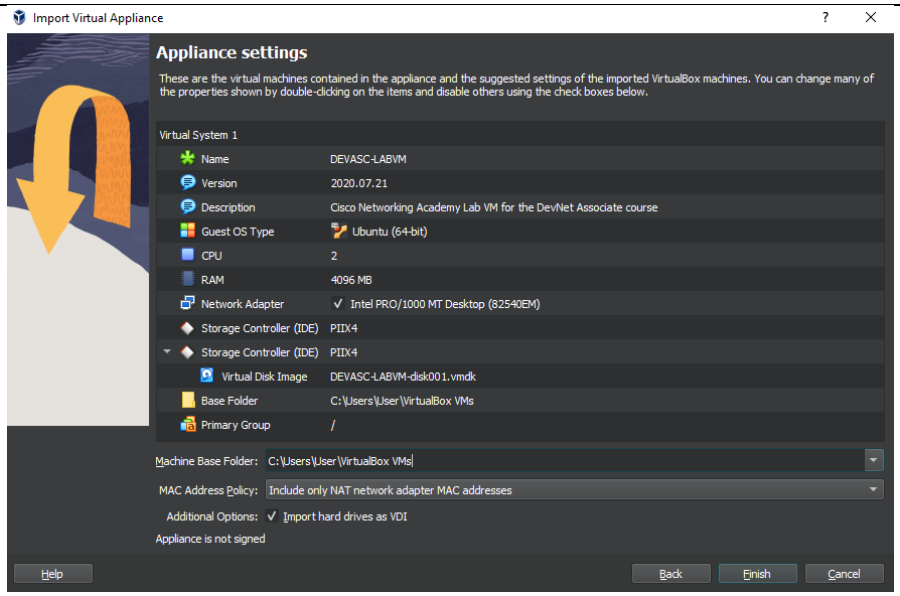
- Make sure you have the DEVASC Virtual Machine
- Refer to the following if you don't have the required VM
  - Download the application of the VM workstation here: https://www.netacad.com/portal/resources/file/a7801868-0dab-4d4c-83b1-c0a7372ab7e1
  - Instruction to import and install the ova: https://tip.instructure.com/courses/35734/files/17012039?wrap=1
- **NOTE: Provide screenshots of every output of the objectives.**

https://tip.instructure.com/courses/56188/files/25936771?wrap=1

2. Output



Installed DEVASC.ova



Importing DEVASC appliance to VM

```
devasc@labvm:~$ ~/labs/devnet-src/sample-app
bash: /home/devasc/labs/devnet-src/sample-app: Is a directory
devasc@labvm:~$
```

changing the working directory

```
devasc@labvm:~$ cd labs/devnet-src/sample-app/
devasc@labvm:~/labs/devnet-src/sample-app$ touch user-input.sh
devasc@labvm:~/labs/devnet-src/sample-app$
```

creating a simple bash file

```
  GNU nano 4.8                                  user-input.sh
#! /bin/bash

echo -n "Enter Your Name: "
read userName
echo "Your name is $userName."
```

simple bash file contents

```
devasc@labvm:~/labs/devnet-src/sample-app$ bash user-input.sh
Enter Your Name: Julius
Your name is Julius.
devasc@labvm:~/labs/devnet-src/sample-app$
```

testing the bash file

```
devasc@labvm:~/labs/devnet-src/sample-app$ ls -l user-input.sh
-rw-rw-r-- 1 devasc devasc 87 Nov 24 06:48 user-input.sh
devasc@labvm:~/labs/devnet-src/sample-app$ chmod a+x user-input.sh
devasc@labvm:~/labs/devnet-src/sample-app$ ls -l user-input.sh
-rwxrwxr-x 1 devasc devasc 87 Nov 24 06:48 user-input.sh
devasc@labvm:~/labs/devnet-src/sample-app$
```

changing the permissions to be executable by all users

```
devasc@labvm:~/labs/devnet-src/sample-app$ mv user-input.sh user-input
devasc@labvm:~/labs/devnet-src/sample-app$
```

renaming the user-input.sh to just user-input

```
devasc@labvm:~/labs/devnet-src/sample-app$ ./user-input
Enter Your Name: Julius
Your name is Julius.
devasc@labvm:~/labs/devnet-src/sample-app$
```

running the bash script without the source command

```
devasc@labvm:~/labs/devnet-src/sample-app$ pip3 install flask
Requirement already satisfied: flask in /home/devasc/.local/lib/python3.8/site-packages (1.1
.2)
Requirement already satisfied: click>=5.1 in /home/devasc/.local/lib/python3.8/site-packages
 (from flask) (7.1.2)
Requirement already satisfied: itsdangerous>=0.24 in /home/devasc/.local/lib/python3.8/site-
packages (from flask) (1.1.0)
Requirement already satisfied: Jinja2>=2.10.1 in /home/devasc/.local/lib/python3.8/site-pack
ages (from flask) (2.11.2)
Requirement already satisfied: Werkzeug>=0.15 in /home/devasc/.local/lib/python3.8/site-pack
ages (from flask) (1.0.1)
Requirement already satisfied: MarkupSafe>=0.23 in /home/devasc/.local/lib/python3.8/site-pa
ckages (from Jinja2>=2.10.1->flask) (1.1.1)
devasc@labvm:~/labs/devnet-src/sample-app$
```

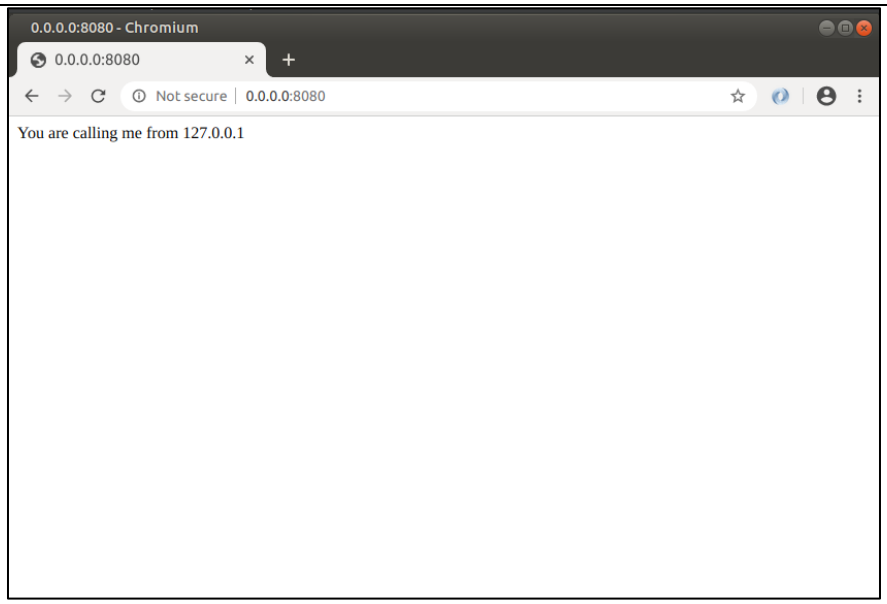
Flask framework installation

```
  GNU nano 4.8                              sample_app.py
# Add to this file for the sample app lab

from flask import Flask
from flask import request

sample = Flask(__name__)
@sample.route("/")
def main():
    return "You are calling me from " + request.remote_addr + "\n"


if __name__ == "__main__":
    sample.run(host="0.0.0.0", port=8080)
```

sample_app.py contents

```
devasc@labvm:~/labs/devnet-src/sample-app$ python3 sample_app.py
 * Serving Flask app "sample_app" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://0.0.0.0:8080/ (Press CTRL+C to quit)
```

checking if sample_app.py is running



verified successfully that server is running

```
devasc@labvm:~$ curl http://0.0.0.0:8080
You are calling me from 127.0.0.1
devasc@labvm:~$
```

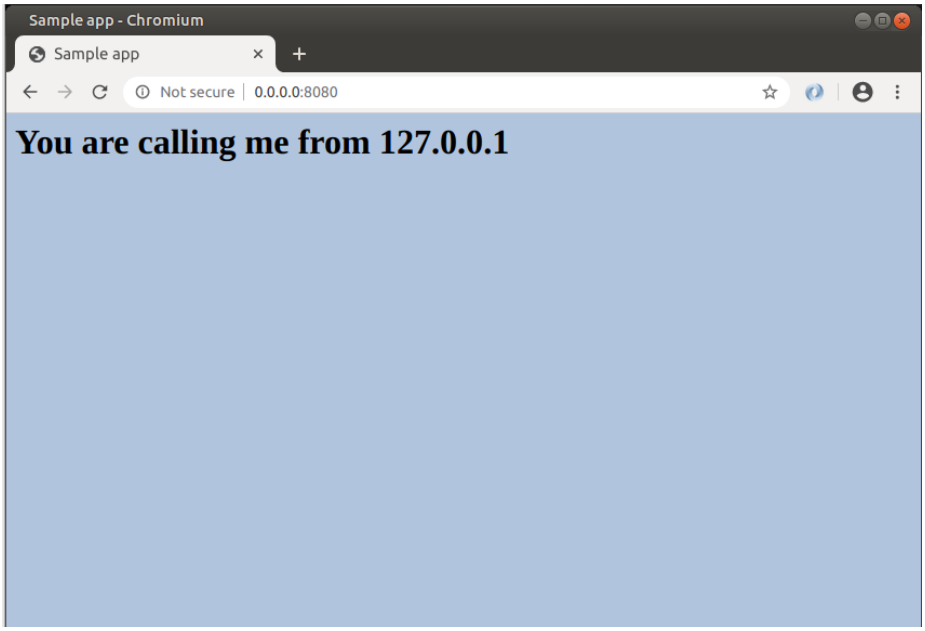
verified via command-line URL tool

```
^Cdevasc@labvm:~/labs/devnet-src/sample-app$ cat templates/index.html
<html>
<head>
    <title>Sample app</title>
    <link rel="stylesheet" href="/static/style.css" />
</head>
<body>
    <h1>You are calling me from {{request.remote_addr}}</h1>
</body>
</html>
devasc@labvm:~/labs/devnet-src/sample-app$
```

concatenating index.html contents

```
 GNU nano 4.8                                    sample_app.py
# Add to this file for the sample app lab

from flask import Flask
from flask import request
from flask import render_template

sample = Flask(__name__)
@sample.route("/")
def main():
    return render_template("index.html")

if __name__ == "__main__":
    sample.run(host="0.0.0.0", port=8080)
```

modified sample_app.py



rendering index.html web file

```
 GNU nano 4.8                          sample-app.sh                          Modified

mkdir tempdir
mkdir tempdir/templates
mkdir tempdir/static

cp sample_app.py tempdir/.
cp -r templates/* tempdir/templates/.
cp -r static/* tempdir/static/.

echo "FROM python" >> tempdir/Dockerfile
echo "RUN pip install flask" >> tempdir/Dockerfile

echo "COPY  ./static /home/myapp/static/" >> tempdir/Dockerfile
echo "COPY  ./templates/home/myapp/templates/" >> tempdir/Dockerfile
echo "COPY  sample_app.py /home/myapp/" >> tempdir/Dockerfile

echo "EXPOSE 8080" >> tempdir/Dockerfile
echo "CMD python3 /home/myapp/sample_app.py" >> tempdir/Dockerfile

cd tempdir
docker build -t sampleapp .
docker run -t -d -p 8080:8080 --name samplerunning sampleapp

docker ps -a
```

sample-app.sh bash script

```
devasc@labvm:~/labs/devnet-src/sample-app$ bash ./sample-app.sh
Sending build context to Docker daemon  6.144kB
Step 1/7 : FROM python
latest: Pulling from library/python
b2b31b28ee3c: Pull complete
c3cc7b6f0473: Pull complete
2112e5e7c3ff: Pull complete
af247aac0764: Pull complete
ef45f15f570b: Pull complete
4d87d670c3ff: Pull complete
3ed3f00b0d2c: Pull complete
Digest: sha256:bc78d3c007f86dbb87d711b8b082d9d564b8025487e780d24ccb8581d83ef8b0
Status: Downloaded newer image for python:latest
 ---> c41ea8273365
Step 2/7 : RUN pip install flask
 ---> Running in 9ddcbfc1fbc9
Collecting flask
  Downloading flask-3.1.0-py3-none-any.whl.metadata (2.7 kB)
Collecting Werkzeug>=3.1 (from flask)
  Downloading werkzeug-3.1.3-py3-none-any.whl.metadata (3.7 kB)
Collecting Jinja2>=3.1.2 (from flask)
  Downloading jinja2-3.1.4-py3-none-any.whl.metadata (2.6 kB)
Collecting itsdangerous>=2.2 (from flask)
  Downloading itsdangerous-2.2.0-py3-none-any.whl.metadata (1.9 kB)
Collecting click>=8.1.3 (from flask)
  Downloading click-8.1.7-py3-none-any.whl.metadata (3.0 kB)
Collecting blinker>=1.9 (from flask)
  Downloading blinker-1.9.0-py3-none-any.whl.metadata (1.6 kB)
```

executing sample-app.sh (1)

```
Collecting MarkupSafe>=2.0 (from Jinja2>=3.1.2->flask)
  Downloading MarkupSafe-3.0.2-cp313-cp313-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.me
tadata (4.0 kB)
Downloading flask-3.1.0-py3-none-any.whl (102 kB)
Downloading blinker-1.9.0-py3-none-any.whl (8.5 kB)
Downloading click-8.1.7-py3-none-any.whl (97 kB)
Downloading itsdangerous-2.2.0-py3-none-any.whl (16 kB)
Downloading jinja2-3.1.4-py3-none-any.whl (133 kB)
Downloading werkzeug-3.1.3-py3-none-any.whl (224 kB)
Downloading MarkupSafe-3.0.2-cp313-cp313-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (23
kB)
Installing collected packages: MarkupSafe, itsdangerous, click, blinker, Werkzeug, Jinja2, f
lask
Successfully installed Jinja2-3.1.4 MarkupSafe-3.0.2 Werkzeug-3.1.3 blinker-1.9.0 click-8.1.
7 flask-3.1.0 itsdangerous-2.2.0
```

executing sample-app.sh (2)

```
Removing intermediate container 9ddcbfc1fbc9
 ---> f31b53cb67a8
Step 3/7 : COPY  ./static /home/myapp/static/
 ---> c4beb13a6806
Step 4/7 : COPY  ./templates /home/myapp/templates/
 ---> 977494bfd72d
Step 5/7 : COPY  sample_app.py /home/myapp/
 ---> 3daf29d56483
Step 6/7 : EXPOSE 8080
 ---> Running in 721a50264490
Removing intermediate container 721a50264490
 ---> 34c24c1effd6
Step 7/7 : CMD python3 /home/myapp/sample_app.py
 ---> Running in 63b36b0f4306
Removing intermediate container 63b36b0f4306
 ---> 73b1e01a11e3
Successfully built 73b1e01a11e3
Successfully tagged sampleapp:latest
7749057f11bad0386d2dda61d8c166ceaff5379d50a98596f28a6cf908362ed5
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS
                 PORTS                    NAMES
7749057f11ba        sampleapp           "/bin/sh -c 'python3…"   1 second ago        Up Less
 than a second   0.0.0.0:8080->8080/tcp   samplerunning
devasc@labvm:~/labs/devnet-src/sample-app$
```

executing sample-app.sh (3)

```
devasc@labvm:~/labs/devnet-src/sample-app$ ls tempdir/
Dockerfile  sample_app.py  static  templates
devasc@labvm:~/labs/devnet-src/sample-app$
```

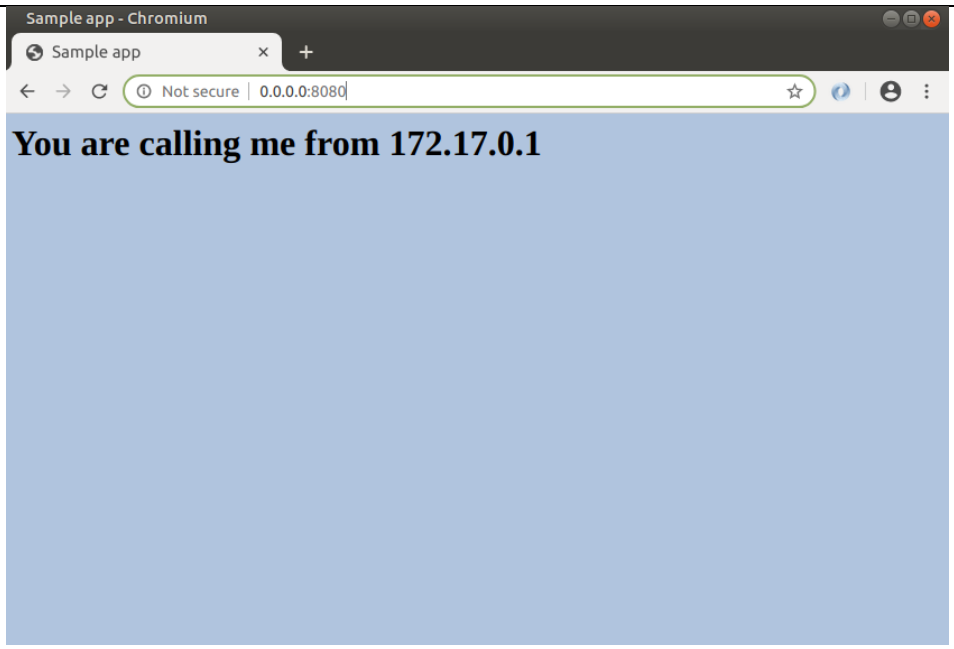
verifying tempdir contents

```
devasc@labvm:~/labs/devnet-src/sample-app$ cat tempdir/Dockerfile
FROM python
RUN pip install flask
COPY  ./static /home/myapp/static/
COPY  ./templates /home/myapp/templates/
COPY  sample_app.py /home/myapp/
EXPOSE 8080
CMD python3 /home/myapp/sample_app.py
devasc@labvm:~/labs/devnet-src/sample-app$
```

verifying Dockerfile contents

```
  GNU nano 4.8                                                     running.txt
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS                    NAMES
f95784c5962a        sampleapp           "/bin/sh -c 'python3…"   45 seconds ago      Up 44 seconds       0.0.0.0:8080->8080/tcp   samplerunning
```

running.txt contents



verifying through website

```
File  Edit  View  Search  Terminal  Help
devasc@labvm:~$ curl http://0.0.0.0:8080
<html>
<head>
    <title>Sample app</title>
    <link rel="stylesheet" href="/static/style.css" />
</head>
<body>
    <h1>You are calling me from 172.17.0.1</h1>
</body>
</html>devasc@labvm:~$
```

verifying through command-line URL tool

```
devasc@labvm:~/labs/devnet-src/sample-app$ ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
```

local ip address

```
4: docker0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP gr
oup default
    link/ether 02:42:57:aa:a1:84 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
       valid_lft forever preferred_lft forever
    inet6 fe80::42:57ff:feaa:a184/64 scope link
       valid_lft forever preferred_lft forever
```

docker ip address

```
devasc@labvm:~/labs/devnet-src/sample-app$ docker exec -it samplerunning /bin/bash
root@f95784c5962a:/# ls
bin   dev  home  lib64  mnt  proc  run   srv  tmp  var
boot  etc  lib   media  opt  root  sbin  sys  usr
root@f95784c5962a:/# ls home/myapp/
sample_app.py  static  templates
root@f95784c5962a:/# exit
exit
devasc@labvm:~/labs/devnet-src/sample-app$
```

accessing and exploring running docker container

```
devasc@labvm:~/labs/devnet-src/sample-app$ docker stop samplerunning
samplerunning
devasc@labvm:~/labs/devnet-src/sample-app$ docker ps -a
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS
                    PORTS               NAMES
f95784c5962a        sampleapp           "/bin/sh -c 'python3…"   24 minutes ago      Exited
(137) 38 seconds ago                    samplerunning
```

stopping samplerunning docker container

```
devasc@labvm:~/labs/devnet-src/sample-app$ docker start samplerunning
samplerunning
devasc@labvm:~/labs/devnet-src/sample-app$ docker ps -a
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS
            PORTS                     NAMES
f95784c5962a        sampleapp           "/bin/sh -c 'python3…"   25 minutes ago      Up 9 se
conds       0.0.0.0:8080->8080/tcp    samplerunning
devasc@labvm:~/labs/devnet-src/sample-app$
```

starting samplerunning docker container

```
devasc@labvm:~/labs/devnet-src/sample-app$ docker stop samplerunning
samplerunning
devasc@labvm:~/labs/devnet-src/sample-app$ docker rm samplerunning
samplerunning
devasc@labvm:~/labs/devnet-src/sample-app$ docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS
       PORTS               NAMES
devasc@labvm:~/labs/devnet-src/sample-app$
```

stopping and removing samplerunning docker container

Github Link: https://github.com/jmado-biscoff/Activity-12.git

3. Conclusion

In this activity, I successfully built a sample web app within a Docker container as part of the task. The process involved setting up the necessary environment, including the installation of the Flask framework and configuring the Docker container to host the web app. However, I faced a challenge in accessing the website from the Docker container, as it was initially unreachable, causing some frustration.

After troubleshooting, I explored several possible causes, such as verifying that the correct ports were exposed and mapped between the container and the host machine. I also ensured the Flask application was configured to listen on all available interfaces (0.0.0.0), rather than just 127.0.0.1. Additionally, I checked the Docker container's network settings and verified that there were no firewall or resource limitations blocking the connection.

Through these troubleshooting steps, I was able to resolve the issue and successfully access the website hosted in the Docker container. Ultimately, the web application functioned as intended, providing a valuable learning experience in both Docker container management and web application deployment.

The use of Docker in developing websites is of significant importance. It allows developers to create isolated environments, ensuring that the application runs consistently across different platforms. Docker containers package the application and its dependencies, making it easier to deploy, scale, and manage web apps in a reproducible and efficient manner. This is particularly useful in modern development workflows, where consistency, speed, and scalability are key factors. Docker simplifies testing, continuous integration, and deployment, reducing the "it works on my machine" problem and enhancing collaboration among development teams. As a result, Docker plays a vital role in the development, deployment, and maintenance of web applications.