Create an docker volume named sampledata

docker volume create sampledata
docker volume inspect sampledata

Spin up an ubuntu docker container, mount the sampledata docker volume. Pass the env variable os=ubuntu to the docker container when bringing it up

docker run -dit -v sampledata:/sampleVolume --env OS=ubuntu --name ubuntu ubuntu

Create an dir names /sampledir mount it to an nginx:1.19.0 docker container

mkdir sampledir
docker run -dit -v sampledir:/var/www/html/ --name nginx nginx:1.19.0

```
raji@raji-VivoBook-ASUSLaptop-X409MA:~/practice$ docker volume create sampledata
sampledata
raji@raji-VivoBook-ASUSLaptop-X409MA:~/practice$ docker volume inspect sampledata
[
    {
        "CreatedAt": "2022-06-20T19:18:35+05:30",
        "Driver": "local",
        "Labels": {},
        "Mountpoint": "/var/lib/docker/volumes/sampledata/_data",
        "Name": "sampledata",
        "Options": {},
        "Scope": "local"
    }
]
raji@raji-VivoBook-ASUSLaptop-X409MA:~/practice$ docker run -dit -v sampledata:/sampleVolume --env OS=ubuntu --name ubuntu ubuntu
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
405f018f9d1d: Pull complete
Digest: sha256:b6b83d3c331794420340093eb706a6f152d9c1fa51b262d9bf34594887c2c7ac
Status: Downloaded newer image for ubuntu:latest
d1e0dbc2d51437f5d2d2282d2345e997ff2264c521e3fe5842f61b4b672f6c3e
raji@raji-VivoBook-ASUSLaptop-X409MA:~/practice$ mkdir sampledir
raji@raji-VivoBook-ASUSLaptop-X409MA:~/practice$ docker run -dit -v sampledir:/var/www/html/ --name nginx nginx:1.19.0
Unable to find image 'nginx:1.19.0' locally
1.19.0: Pulling from library/nginx
8559a31e96f4: Pull complete
8d69e59170f7: Pull complete
3f9f1ec1d262: Pull complete
d1f5ff4f210d: Pull complete
1e22bfa8652e: Pull complete
Digest: sha256:21f32f6c08406306d822a0e6e8b7dc81f53f336570e852e25fbe1e3e3d0d0133
Status: Downloaded newer image for nginx:1.19.0
7b00e802d40300339a07175d0470989b39f256e36025ded0ae6ba1446d14ba7a
raji@raji-VivoBook-ASUSLaptop-X409MA:~/practice$
```

Create an docker file to create an image with httpd2 server in it, on running the image as an container the server should start up

git clone https://github.com/s-rajarajeshwari/Website.git
vim Dockerfile

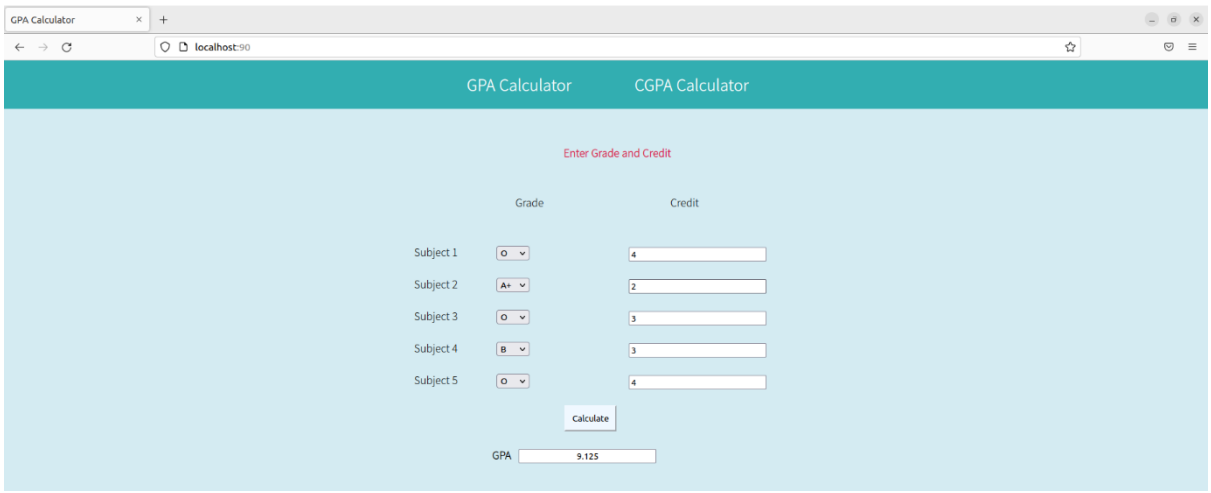
```
FROM httpd:2.4
COPY ./Website/ /usr/local/apache2/htdocs/

~
~
```

docker build -t apache-image .

Create an container out of the image you created in earlier step expose the 80 port to the parent host

//Exposing to port 90, as 80 is already in use
docker run -d -p 90:80 --name apache-container apache-image

```
raji@raji-VivoBook-ASUSLaptop-X409MA:~/practice/docker$ git clone https://github.com/s-rajarajeshwari/Website.git
Cloning into 'Website'...
remote: Enumerating objects: 10, done.
remote: Counting objects: 100% (10/10), done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 10 (delta 1), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (10/10), done.
Resolving deltas: 100% (1/1), done.
raji@raji-VivoBook-ASUSLaptop-X409MA:~/practice/docker$ vim Dockerfile
raji@raji-VivoBook-ASUSLaptop-X409MA:~/practice/docker$ docker build -t apache-image .
Sending build context to Docker daemon  73.22kB
Step 1/2 : FROM httpd:2.4
 ---> b260a49eebf9
Step 2/2 : COPY ./Website/ /usr/local/apache2/htdocs/
 ---> e0cfc40cf56b
Successfully built e0cfc40cf56b
Successfully tagged apache-image:latest
raji@raji-VivoBook-ASUSLaptop-X409MA:~/practice/docker$ docker run -d -p 90:80 --name apache-container apache-image
a42761ffad6c31d9a6458a4e9565db8ab3fd23e7cc06e121b9188bdeccb786c2
raji@raji-VivoBook-ASUSLaptop-X409MA:~/practice/docker$
```



Create an bridge network and spin up two containers in the network one with the image you created

docker network create bridge-network
docker network connect bridge-network apache-container
docker network connect bridge-network ubuntu
docker network inspect bridge-network

```
raji@raji-VivoBook-ASUSLaptop-X409MA:~/practice/docker$ docker network create bridge-network
a3c871841bb6a81143aad2a10467b4ec06a106925b010a1192d24d5446484f9f
raji@raji-VivoBook-ASUSLaptop-X409MA:~/practice/docker$ docker network connect bridge-network apache-container
raji@raji-VivoBook-ASUSLaptop-X409MA:~/practice/docker$ docker network connect bridge-network ubuntu
raji@raji-VivoBook-ASUSLaptop-X409MA:~/practice/docker$ docker network inspect bridge-network
[
    {
        "Name": "bridge-network",
        "Id": "a3c871841bb6a81143aad2a10467b4ec06a106925b010a1192d24d5446484f9f",
        "Created": "2022-06-20T21:46:16.263579088+05:30",
        "Scope": "local",
        "Driver": "bridge",
        "EnableIPv6": false,
        "IPAM": {
            "Driver": "default",
            "Options": {},
            "Config": [
                {
                    "Subnet": "172.19.0.0/16",
                    "Gateway": "172.19.0.1"
                }
            ]
        },
        "Internal": false,
        "Attachable": false,
        "Ingress": false,
        "ConfigFrom": {
            "Network": ""
        },
        "ConfigOnly": false,
        "Containers": {
            "36534e08c5f62b2c7103c63fea643a3da78292e67055d33ff8366f102464a412": {
                "Name": "apache-container",
                "EndpointID": "e8b955076ad37a3f7b3d8d4ca3afa6e34c34daf3b7ea65e0f96e6e1e2f4093cb",
                "MacAddress": "02:42:ac:13:00:02",
                "IPv4Address": "172.19.0.2/16",
                "IPv6Address": ""
            },
            "d1e0dbc2d51437f5d2d2282d2345e997ff2264c521e3fe5842f61b4b672f6c3e": {
                "Name": "ubuntu",
                "EndpointID": "b86544e08af90a2aaba533895664ff0ccf84faf016a0a9e87b8290e81dea24da",
                "MacAddress": "02:42:ac:13:00:03",
                "IPv4Address": "172.19.0.3/16",
                "IPv6Address": ""
            }
        },
        "Options": {},
        "Labels": {}
    }
]
raji@raji-VivoBook-ASUSLaptop-X409MA:~/practice/docker$
```

Create an docker compose file with an nginx and an redis server both connected via an bridge network

docker-compose --version
vim docker-compose.yml

```
version: "3"
services:
  nginx:
    image: nginx
    ports:
      - "8080:80"
    networks:
      - bridge-network
  redis:
    image: redis
    ports:
      - "8085:6379"
    networks:
      - bridge-network
networks:
  bridge-network:
~
~
```

docker-compose up

Scenario Based exercise

Write Docker file to create an image with the apache httpd2 server make the start of the httpd2 server as the container entry point.

vi index.html

```
<h1>Hello World</h1>
~
~
```

vi Dockerfile

```
FROM ubuntu:latest

RUN apt-get -y update
RUN apt-get install -y apache2 curl

EXPOSE 80
WORKDIR /var/www/html
COPY index.html /var/www/html/index.html

ENTRYPOINT ["/usr/sbin/apache2ctl"]
CMD ["-D", "FOREGROUND"]
~
~
~
```

sudo docker build -t apache-server .
sudo docker images
sudo docker run -d --name apache apache-server

sudo docker exec apache curl localhost

```
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100    21  100 <h1>Hello World</h1>  0      0 --:--:-- --:--:-- --:--:--     0
   21    0     0   1330      0 --:--:-- --:--:-- --:--:--  1400
ubuntu@ip-172-31-33-124:~/docker$
```

Create an git repository with some html files for our webserver, clone the repo to an local linux machine.

git clone https://github.com/s-rajarajeshwari/Website.git

```
Cloning into 'Website'...
remote: Enumerating objects: 10, done.
remote: Counting objects: 100% (10/10), done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 10 (delta 1), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (10/10), done.
Resolving deltas: 100% (1/1), done.
ubuntu@ip-172-31-33-124:~/docker$
```

Create an bash script which will pull the contents from the GitHub Repo every 5 hours (use Linux Cron) to the local cloned repo, The script should log all these activities to an log file with timestamp (file which keep track of all script runs)

vi pull.sh

```
#!/bin/bash
# Author: Rajarajeshwari Sridharan
# Date: 21/06/2022
# Description: Pull contents from git and log activities in log.txt
# Date Modified: 21/06/2022

git pull
date  >> /home/ubuntu/docker/log.txt
~
~
```

crontab -e

```
  GNU nano 6.2                                    /tmp/crontab.Gh03h7/crontab
0 */5 * * * /bin/sh /home/ubuntu/docker/pull.sh
```

Spin up the container out of the image you create from the docker file, expose the port 80, mount the dir where you have cloned the git repo to the dir inside the container from where httpd2 server reads the HTML files (/var/www/html)

sudo docker run -dit -p 80:80 -v /home/ubuntu/docker/Website:/var/www/html --name apache apache-server

Do all the above steps inside an AWS linux EC2 machine and share the publicip/DNS of the VM, by allowing the traffic to reach your webpage

Public IPv4 address: 13.234.213.84