

Karthik Pradeep Hegadi

2KE20CS032

Assignment 21

Understood. To follow the provided instructions and create the files/directory using the same name and case as provided in the task steps, please provide me with the specific names and case instructions for the files/directory you want to create.

DOCKER

My docker hub link: <https://hub.docker.com/u/kartik404>

Assignment -1 Run Nginx using Docker

S1 :Installing the docker and verifying

```
lusy@localhost.localdomain ~/Docker
sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: disabled)
   Active: active (running) since Sun 2023-09-17 12:05:05 IST; 25min ago
     TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
      Main PID: 105009 (dockerd)
        Tasks: 44
       Memory: 64.4M
          CPU: 1.441s
        CGroup: /system.slice/docker.service
                  ├─105009 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.
                  ├─107477 /usr/bin/docker-proxy -proto tcp -host-ip 0.0.0.0 -host-port 3307
                  ├─107483 /usr/bin/docker-proxy -proto tcp -host-ip :: -host-port 3307 -cont
                  ├─127422 /usr/bin/docker-proxy -proto tcp -host-ip 0.0.0.0 -host-port 82 -c
```

S2 :Creating the index file

```
lusy@localhost.localdomain ~/Docker (0.018s)
cat index.html
<h1> Docker Test <h1>
```

S3:Creating the Dockerfile

```
cat Dockerfile
FROM centos:7
RUN yum -y install epel-release
RUN yum -y update
RUN yum -y install nginx
ADD index.html /usr/share/nginx/html/index.html
CMD ["nginx", "-g daemon off;"]
```

S4 :Building the docker file

```
lusy@localhost.localdomain ~/Docker
sudo docker build . -t centos7

[+] Building 16.2s (2/3)
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 267B
[internal] load build definition from Dockerfile
[internal] load build definition from Dockerfile
```

S5: Verify the image

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
mysql	latest	e248da1c6d57	30 minutes ago	507MB
mattermost	latest	643152674348	About an hour ago	1.41GB
centos7	latest	1388cc76b12c	3 days ago	1.4GB
nginx	latest	91582cfffc2d	10 days ago	192MB
hello-world	latest	b038788ddb22	4 months ago	9.14kB
centos	latest	e6a0117ec169	2 years ago	272MB

S6 :Run the image

```
lusy@localhost.localdomain ~/Docker2 (0.291s)
sudo docker run -d -p 82:80 --name centos7 centos7
8fd304fae7dc0715535565dce1233f5eafce8104df2cb835884e999332a442e0
```

S7:Verify

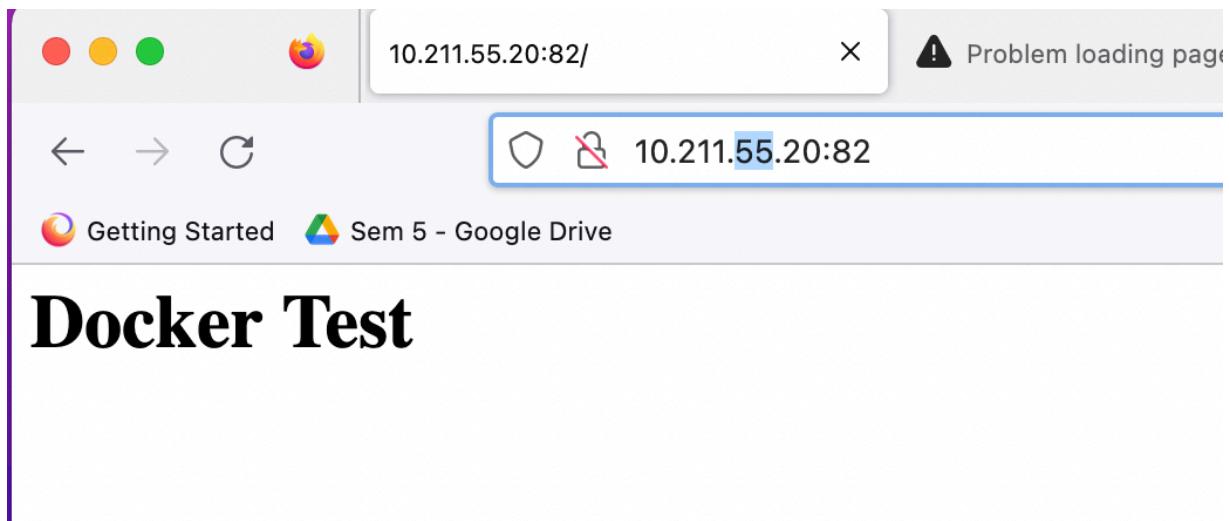
```
lusy@localhost.localdomain ~/Docker2 (0.08s)
sudo docker ps
CONTAINER ID   IMAGE      COMMAND           CREATED          STATUS          PORTS
NAMES
8fd304fae7dc   centos7   "nginx '-g daemon of..."   14 seconds ago   Up 14 seconds   0.0.0.0:82->80/tcp, :::82->80
centos7
b565ff3fd9a5   mysql     "/entrypoint.sh mysq..."   19 minutes ago   Up 19 minutes (healthy)  33060-33061/tcp, 0.0.0.0:3307
:3307->3306/tcp   mysqld
```

S8:Start the container and verify the weather it is running or not

```
lusy@localhost.localdomain ~/Docker2 (0.066s)
sudo docker start centos7
centos7
```

```
lusy@localhost.localdomain ~/Docker2 (0.043s)
curl 172.17.0.1:82
<h1> Docker Test <h1>
```

S9 :Verifying the image in website with port number (82 and centos IP)



Assignment - 2 - Run Mattermost Server in Docker

S1:Make your files and zip ready for building the image

```
lusy@localhost.localdomain:~/Docker1 (0.022s)
ls
config.json  Dockerfile  mattermost-v7.5.2-linux-arm64.tar.gz
```

S2:Creating the Dockerfile

```
lusy@localhost.localdomain:~/Docker1 (0.015s)
cat Dockerfile
# Use the CentOS 7 base image
FROM centos:7

# Download and extract Mattermost
COPY mattermost-v7.5.2-linux-arm64.tar.gz /
RUN tar -xvzf *.gz
RUN mv /mattermost /opt/

# Copy the Mattermost configuration file
COPY config.json /opt/mattermost/config/config.json

# Create a system user and group for Mattermost
RUN useradd --system --user-group mattermost

# Change ownership of the Mattermost directory
RUN chown -R mattermost:mattermost /opt/mattermost

# Give group write permissions to certain directories
RUN chmod -R g+w /opt/mattermost/

# Switch to the Mattermost user
USER mattermost:mattermost

# Define the volume for Mattermost data
VOLUME /opt/mattermost/data

# Start Mattermost
CMD ["/opt/mattermost/bin/mattermost"]
```

S3: Building the image

```
lusy@localhost.localdomain ~/Docker1 (3.164s)
sudo docker build . -t mattermost

[+] Building 3.2s (13/13) FINISHED
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 831B
=> [internal] load metadata for docker.io/library/centos:7
=> [1/8] FROM docker.io/library/centos:7@sha256:be65f488b7764a
=> [internal] load build context
      1 file copied, 216B
```

S4: Verify the image

```
lusy@localhost.localdomain ~/Docker1 (0.067s)
sudo docker images

REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
mysql           latest   e248da1c6d57  About an hour ago  507M
mattermost       latest   643152674348  2 hours ago   1.4M
centos7          latest   1388cc76b12c  3 days ago   1.4M
nginx            latest   91582cfffc2d  10 days ago  192M
hello-world      latest   b038788ddb22  4 months ago  9.1M
centos           latest   e6a0117ec169  2 years ago  272M
```

S5 :Run the container(On port 8067)

```
lusy@localhost.localdomain ~/Docker1 (0.272s)
sudo docker run -d -p 8067:8065 --name mattermost mattermost
8aa4efee5e1ab2d95ea2b8345b626c182916181d55d4ed50e85b3940b49da9b
```

S6:Verify the container process

```
lusy@localhost.localdomain ~/Docker1 (0.104s)
sudo docker ps

CONTAINER ID  IMAGE      COMMAND      CREATED      STATUS      PORTS
8aa4efee5e1  mattermost  "/opt/mattermost/bin..."  15 minutes ago  Up 15 minutes  0.0.0.0:8067->8065/tcp, :::8067->8065/tcp
8fd304fae7dc  centos7    "nginx '-g daemon of..."  52 minutes ago  Up 52 minutes  0.0.0.0:82->80/tcp, :::82->80/tcp
```

S7: Start the container

```
lusy@localhost.localdomain ~/Docker1 (0.07s)
sudo docker start mattermost
mattermost
```

S8: Verification

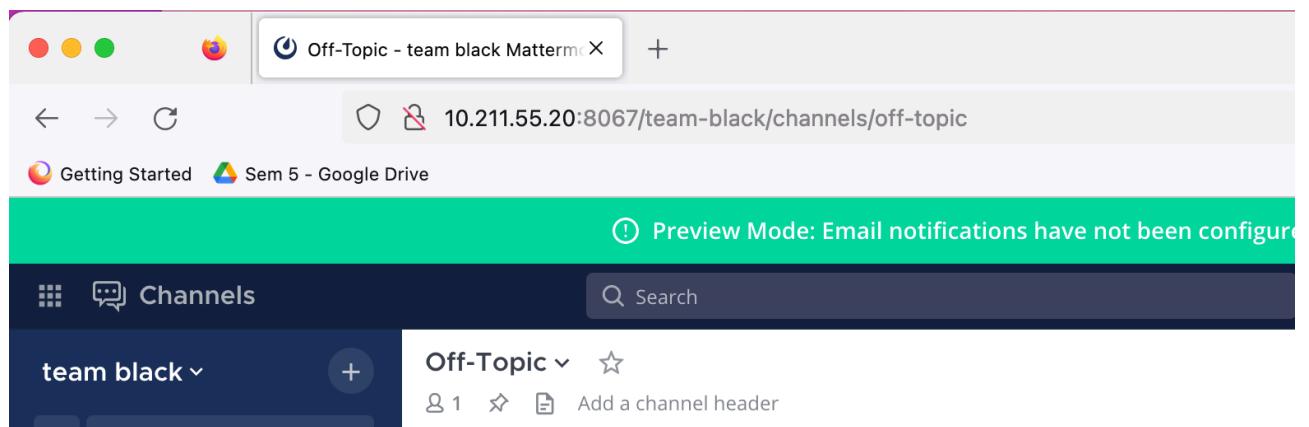
S8-1 :Using Docker ip with port

```
lusy@localhost.localdomain ~/Docker1 (0.039s)
curl 172.17.0.1:8067
using Docker IP
<!doctype html><html lang="en"><head><meta charset="utf-8"><meta name="ts" content="noindex, nofollow"><meta name="referrer" content="no-re-name" content="Mattermost"><meta name="format-detection" content="t
```

S8-2:Using centos ip with port

```
lusy@localhost.localdomain ~/Docker1 (0.03s)
curl 10.211.55.20:8067
using my iP
<!doctype html><html lang="en"><head><meta charset="utf-8"><meta name="ts" content="noindex, nofollow"><meta name="referrer" content="no-re-name" content="Mattermost"><meta name="format-detection" content="tel
```

S8-3: Verify whether it is running in the web site or not (with port no 8067)



S9 :Optional Verification

```
sudo docker exec -it --user root mattermost /bin/bash
[root@8aa4efeeef5e1 /]# ls
anaconda-post.log  bin  dev  etc  home  lib  lib64  mattermost-v7.5.2-linux-arm64.tar.gz  media
[root@8aa4efeeef5e1 /]#
```

Assignment 3 : Run sql container and configure mattermost db

S1 :Read the required files for Dockerfile to build

```
lusy@localhost.localdomain ~/Docker2 (0.02s)
tree

.
└── Dockerfile
    └── init.sql

0 directories, 2 files
```

S2 :init.sql

```
lusy@localhost.localdomain ~/Docker2 (0.023s)
cat init.sql

CREATE USER 'mmuser1'@'%' IDENTIFIED BY 'mmuser1';
CREATE DATABASE mattermost;
GRANT ALL PRIVILEGES ON mattermost.* TO 'mmuser1'@'%';
```

S3 : Dockerfile

```
lusy@localhost.localdomain:~/Docker2 (0.016s)
cat Dockerfile

FROM mysql/mysql-server:latest
COPY init.sql /docker-entrypoint-initdb.d/
```

S4:Build the image

```
lusy@localhost.localdomain ~/Docker2 (3.042s)
sudo docker build -t mysqld .

[+] Building 3.1s (7/7) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 171B
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load metadata for docker.io/mysql/mysql-s
=> [internal] load build context
=> => transferring context: 89B
=> [1/2] FROM docker.io/mysql/mysql-server:latest@sha25
=> CACHED [2/2] COPY init.sql /docker-entrypoint-initdb.d/
=> exporting to image
=> => exporting layers
```

S5 :Run the container(on port 3307)

```
lusy@localhost.localdomain ~/Docker2 (0.305s)
sudo docker run "--name=mysqld" -p 3307:3306 --restart on-failure -d mysql
432aa0b33a8d3b4d3f6e05cc06e201623afeceb29ca516d9a197652141fd849
```

S6 :Verify the docker process

```
sudo docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS
432aa0b33a8d mysqld "/entrypoint.sh mysq..." About an hour ago Up About an hour (healthy) 33060-33061/tcp, 0.0.0.0:3307->3306/tcp, :::3307->3306/tcp
8aa4feef5e1 mattermost "/opt/mattermost/bin..." 2 hours ago Up 2 hours 0.0.0.0:8067->8065/tcp, :::8067->8065/tcp
8fd304fae7dc centos7 "nginx '-g daemon of..." 2 hours ago Up 2 hours 0.0.0.0:82->80/tcp, :::82->80/tcp
```

S7 :Know your temp password of mysqld

```
lusy@localhost.localdomain ~/Docker2 (0.083s)
sudo docker logs mysqld 2>&1 | grep GENERATED
[Entrypoint] GENERATED ROOT PASSWORD: uSq_yPn4sPy_u^22?*7+47u7FBMR&5,K
```

S8: Get in to mysqld using the root user with commands

```
lusy@localhost.localdomain ~/Docker2 (40m 59.49s)
sudo docker exec -it mysqld mysql -u root -p
Enter password:
```

S9:Set your password according to you

```
'> ^C
mysql> ALTER USER 'root'@'localhost' IDENTIFIED BY 'mysql!123';
Query OK, 0 rows affected (0.02 sec)
```

S10 :Inspect your IP of mysqld

```
lusy@localhost.localdomain ~/Docker (0.067s)
sudo docker inspect mysqld | grep IPAddress
    "SecondaryIPAddresses": null,
    "IPAddress": "172.17.0.4",
    "IPAddress": "172.17.0.4",
```

S11:Get in to your container change the directory to config file

```
lusy@localhost.localdomain ~/Docker (36m 25.24s)
sudo docker exec -it mattermost /bin/bash
bash-4.2$ cd /opt/mattermost/config/
bash-4.2$ ls
README.md  config.json
```

S12:Change the config file set your ip address of container (**the user name changed**)

```
    "IosMinVersion": "",
},
"SqlSettings": {
    "DriverName": "mysql",
    "DataSource": "mmuser1:mmuser1@tcp(172.17.0.4:3306)/mattermost?charset=u
    "DataSourceReplicas": [],
    "DataSourceSearchReplicas": []
```

S13:Verify the mattermost is running or not in container

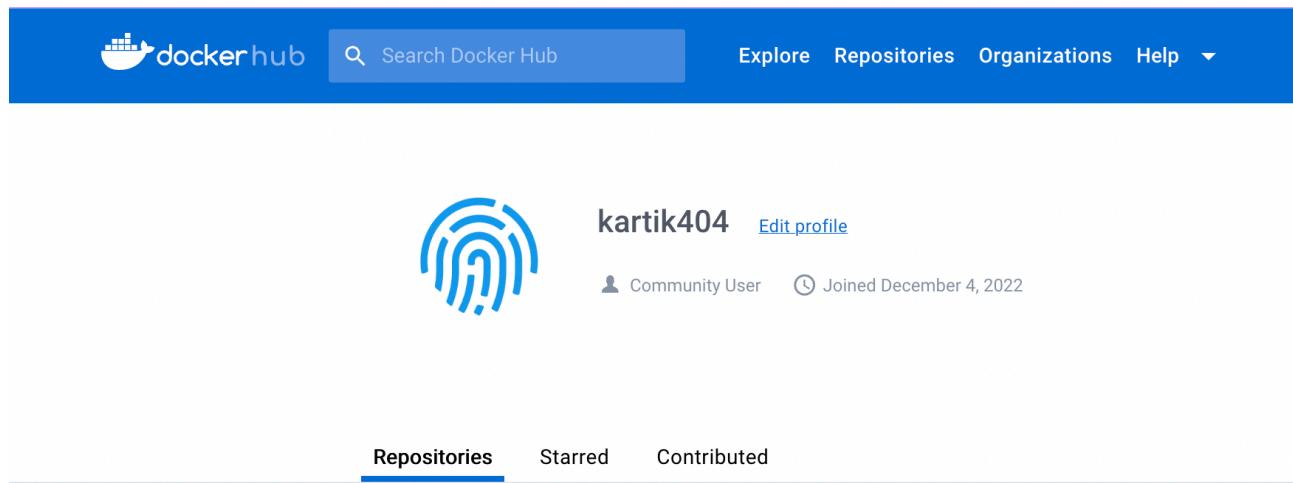
```
bash-4.2$ cd /opt/mattermost/bin/
bash-4.2$ ./mattermost
{"timestamp": "2023-09-17 08:26:44.933 Z", "level": "info", "msg": "Server is in
{"timestamp": "2023-09-17 08:26:44.933 Z", "level": "info", "msg": "Pinging SQL"
{"timestamp": "2023-09-17 08:26:44.966 Z", "level": "info", "msg": "Pinging SQL"
 {"timestamp": "2023-09-17 08:26:47.535 Z", "level": "info", "msg": "Starting web
 {"timestamp": "2023-09-17 08:26:47.535 Z", "level": "warn", "msg": "Sentry repor
```

S14: Verify the process running or not

```
lusi@localhost.localdomain ~ (0.066s)
sudo docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORT
432aa0b33a8d mysqld "/entrypoint.sh mysql" About an hour ago Up About an hour (healthy) 3306
8aa4eefef5e1 mattermost "/opt/mattermost/bin..." 2 hours ago Up 2 hours 0.0.
8fd304fae7dc centos7 "nginx '-g daemon of..." 2 hours ago Up 2 hours 0.0.
```

Assignment - 4 - Push image to your docker hub repository

S1: Create your docker hub account



The screenshot shows a Docker Hub profile for the user 'kartik404'. The profile includes a blue fingerprint icon, the username 'kartik404', a link to 'Edit profile', and status information: 'Community User' joined on December 4, 2022. Below the profile, there are three tabs: 'Repositories' (which is underlined in blue), 'Starred', and 'Contributed'.

S2: Create your tag name

```
lusi@localhost.localdomain ~ (0.128s)
sudo docker tag mysqld kartik404/mysqld:first_dhub

lusi@localhost.localdomain ~ (0.058s)
sudo docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
kartik404/mysqld first_dhub e248da1c6d57 4 hours ago 507MB
mysqld latest e248da1c6d57 4 hours ago 507MB
mysqld/kartik404 first_dhub e248da1c6d57 4 hours ago 507MB
mattermost latest 643152674348 4 hours ago 1.41GB
centos7 latest 1388cc76b12c 4 days ago 1.4GB
nginx latest 91582cfffc2d 10 days ago 192MB
hello-world latest b038788ddb22 4 months ago 9.14kB
centos latest e6a0117ec169 2 years ago 272MB
```

S3: Login your docker in terminal to push the image use your **credentials** to login

```
lusy@localhost.localdomain ~ (3.634s)
sudo docker login
Authenticating with existing credentials...
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
```

S4: Push the image

```
lusy@localhost.localdomain ~ (3m 34.42s)
sudo docker push kartik404/mysqld:first_dhub
The push refers to repository [docker.io/kartik404/mysqld]
fc8614a31e47: Pushed
3535bf4059e0: Pushed
351d1f676c03: Pushed
b1fc9c09140e: Pushed
6cc85e8967db: Pushed
9724055dd422: Pushed
43d5f4639a98: Pushed
3abd5e1fb1b7: Pushed
first_dhub: digest: sha256:3de26902f90c56f7b79fa496c36b5534954fa67d81349b342106
```

S5: Verify the image is pushed or not

The screenshot shows a web browser displaying the Docker Hub repository page for the user 'kartik404'. The URL in the address bar is 'hub.docker.com/repositories/kartik404'. The page header includes the Docker logo and navigation links for 'Explore' and 'Repositories'. A search bar at the top right contains the text 'Search Docker Hub'. Below the search bar, there is a dropdown menu set to 'kartik404' and a search input field with the placeholder 'Search by repository name'. The main content area displays a single repository entry: 'kartik404 / mysqld'. Below the repository name, it says 'Contains: Image | Last pushed: a few seconds ago'.

S6:I have repeated the Same process for other images

The screenshot shows the Docker Hub interface. At the top, there are dropdown menus for 'kartik404' (selected), 'Search by repository name' (with a magnifying glass icon), and 'All Content'. To the right is a blue 'Create repository' button. Below these are two repository cards. The first card is for 'kartik404 / mysqlid', which contains an image last pushed 2 hours ago. It has an 'Inactive' status, 0 stars, 3 downloads, and is public. The second card is for 'kartik404 / mattermost', also containing an image last pushed 2 hours ago, with an 'Inactive' status, 0 stars, 2 downloads, and is public.

Assignment - 5 - Write a docker file to host a simple html content using apache web server

S1 :Make sure your required files are ready

```
lusy@localhost.localdomain ~/Docker3 (0.025s)
tree

.

└── Dockerfile
    └── index.html

0 directories, 2 files
```

S2 :Index file

```
lusy@localhost.localdomain ~/Docker3 (13.23s)
sudo vi index.html
[sudo] password for lusy:

lusy@localhost.localdomain:~/Docker3 (0.02s)
cat index.html
<!DOCTYPE html>
<html>
<head>
    <title>My Dockerized Web App</title>
</head>
<body>
    <h1>Hello, Docker!</h1>
    <p>This is a simple HTML page served by Apache</p>
</body>
</html>
```

S3:Dockerfile

```
lusy@localhost.localdomain:~/Docker3 (0.023s)
cat Dockerfile
# Use an official Apache web server image:tag_name
FROM httpd:2.4

# Copy your HTML file into the default Apache web server
COPY ./index.html /usr/local/apache2/htdocs/
```

S4:Build the image for httpd (apachi)

```
lusy@localhost.localdomain ~/Docker3 (47.53s)
sudo docker build -t httpd_test .

[+] Building 51.3s (8/8) FINISHED
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 279B
=> [internal] load metadata for docker.io/library/httpd:2.4
=> [auth] library/httpd:pull token for registry-1.docker.io
=> [internal] load build context
=> => transferring context: 311B
```

S5:Verify your images

```
lusy@localhost.localdomain ~/Docker3 (0.094s)
sudo docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
httpd_test	latest	834e577d6152	18 seconds ago	195MB
kartik404/mysql	first_dhub	e248da1c6d57	4 hours ago	507MB
mysql	latest	e248da1c6d57	4 hours ago	507MB
mysql/kartik404	first_dhub	e248da1c6d57	4 hours ago	507MB
mattermost	latest	643152674348	4 hours ago	1.41GB
centos7	latest	1388cc76b12c	4 days ago	1.4GB
nginx	latest	91582cfffc2d	10 days ago	192MB
hello-world	latest	b038788ddb22	4 months ago	9.14kB
centos	latest	e6a0117ec169	2 years ago	272MB

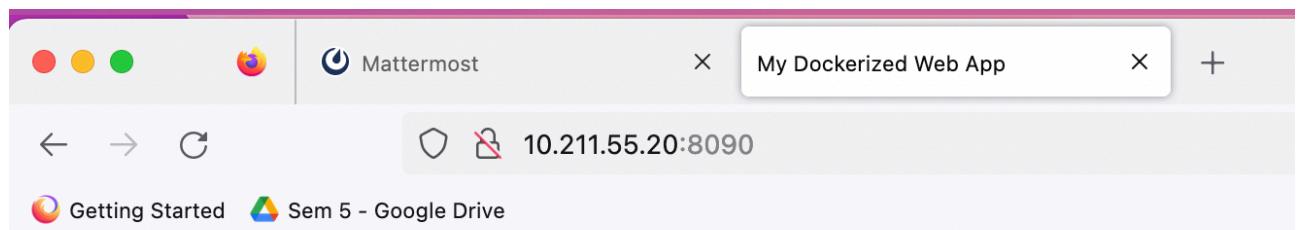
S6: Run your images on your required port which should be free

```
lusy@localhost.localdomain ~/Docker3 (0.356s)
sudo docker run -d -p 8090:80 httpd_test
f034dbc02dc8bdc5f2c9beec168302b4dc8d4679eebceefbce26f811dea1b5
```

S7-1 : Verify it using your ip

```
lusy@localhost.localdomain ~/Docker3 (0.037s)
curl 10.211.55.20:8090
<!DOCTYPE html>
<html>
<head>
    <title>My Dockerized Web App</title>
</head>
<body>
    <h1>Hello, Docker!</h1>
    <p>This is a simple HTML page served by Apache inside a Docker container.</p>
</body>
</html>
```

S7-2: Verify in your web-browser



Hello, Docker!

This is a simple HTML page served by Apache inside a Docker container.

Assignment - 6 - Mount a host directory as a docker volume to mysqld server

container path /var/lib/mysql

S1 :Make a directory

```
lusu@localhost.localdomain:~/Docker4_sql_data (0.019s)
mkdir ~/mysql_data
```

S2 :Run it usign this below mentioned long command 😊 (put your mysqld password and port)

```
sudo docker run -d -p 3308:3306 \
--name mysql_new \
-e MYSQL_ROOT_PASSWORD=mysql!123 \
-v ~/mysql_data:/var/lib/mysql \
mysql:latest
-e MYSQL_ROOT_PASSWORD=mysql123 \

Unable to find image 'mysql:latest' locally
latest: Pulling from library/mysql
d095360c6402: Already exists
a0858301c0b2: Already exists
f442c1551840: Already exists
29d70ba84e12: Already exists
344316b4b482: Already exists
646e59d2b591: Already exists
9d710d330bbe: Already exists
f1fa9584c724: Already exists
59c19adc05d5: Already exists
de6b1dad8c4a: Already exists
Digest: sha256:85ab57eb4a48ada2a341dcf7d96733ce2f370fffb8e8e216991b106e50fa6434
Status: Downloaded newer image for mysql:latest
e2659e1c9ca4a49d0c2cbd673f23ac70884b992bfe7f423555b2ce17e9b46bb8
```

S3 :Verify the the running process

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
e2659e1c9ca4	mysql:latest	"docker-entrypoint.s..."	About a minute ago	Up About a minute	3306/tcp, 0.0.0.0:3308->3306/tcp, :::3308->3306/tcp	mysql_new
f034dbc02dcb	httpd_test	"httpd-foreground"	34 minutes ago	Up 34 minutes	0.0.0.0:8090->80/tcp, :::8090->80/tcp	stupefied_goodall
8aa4efee5sei	mattermost	"/opt/mattermost/bin..."	3 hours ago	Up 3 hours	0.0.0.0:8067->8065/tcp, :::8067->8065/tcp	mattermost
8fd304fae7dc	centos7	"nginx '-g daemon of..."	4 hours ago	Up 4 hours	0.0.0.0:82->80/tcp, :::82->80/tcp	centos7

S4 :Get it in to you directory

```
lusu@localhost.localdomain:~ (0.02s)
ls
Docker Docker1 Docker2 Docker3 Docker4_sql_data mattermost mysql_data sql -v.hwm -v.pwd -v.pwi
```

S5: Check the files

```
lusy@localhost.localdomain ~ (0.019s)
cd mysql_data

lusy@localhost.localdomain ~/mysql_data (0.024s)
tree

.
├── auto.cnf
├── binlog.000001
├── binlog.000002
├── binlog.index
├── ca-key.pem
├── ca.pem
├── client-cert.pem
├── client-key.pem
├── #ib_16384_0 dblwr
├── #ib_16384_1 dblwr
├── ib_buffer_pool
├── ibdata1
├── ibtmp1
├── #innodb_redo [error opening dir]
├── #innodb_temp [error opening dir]
├── mysql [error opening dir]
├── mysql.ibd
└── mysql.sock -> /var/run/mysqld/mysqld.sock
    performance_schema [error opening dir]
    private_key.pem
    public_key.pem
```

Assignment - 7 Hosting Mattermost server using Docker-compose

S1 :List your images and verify ports and other required things and your tag names

```
~ (0.24s)
ssh lusy@10.211.55.20

lusy@localhost.localdomain ~ (0.099s)
docker images

REPOSITORY          TAG      IMAGE ID      CREATED        SIZE
httpd_test          latest   834e577d6152  27 hours ago  195MB
kartik404/mysqlld  arm     e248da1c6d57  31 hours ago  507MB
kartik404/mysqlld  first_dhub  e248da1c6d57  31 hours ago  507MB
mysqlld             latest   e248da1c6d57  31 hours ago  507MB
mysqlld/kartik404  first_dhub  e248da1c6d57  31 hours ago  507MB
kartik404/mattermost  arm    643152674348  31 hours ago  1.41GB
mattermost           latest   643152674348  31 hours ago  1.41GB
mysql               latest   abcbfd216487  2 days ago   599MB
centos7_t            latest   1388cc76b12c  5 days ago   1.4GB
centos7              latest   1388cc76b12c  5 days ago   1.4GB
nginx               latest   91582cfffc2d  11 days ago  192MB
postgres             13      8510658ac7ae  11 days ago  427MB
redis                latest   9160e8a40f34  11 days ago  157MB
hello-world          latest   b038788ddb22  4 months ago  9.14kB
centos               latest   e6a0117ec169  2 years ago  272MB
```

S2: Your docker-compose file

```
lusy@localhost.localdomain ~ (0.022s)
tree

.
└── docker-compose.yml

0 directories, 1 file
```

S3:Content of your compose file

```
version: '1'
services:
  web:
    image: kartik404/mattermost:arm
    ports:
      - "8068:8065"
    networks:
      network:
        ipv4_address: 10.5.0.5
    container_name: mattermost_cont

  database:
    image: kartik404/mysql:arm
    ports:
      - "3308:3306"
    networks:
      network:
        ipv4_address: 10.5.0.6
    container_name: mysql_cont
    volumes:
      - mount_path:/var/lib/mysql

networks:
  network:
    driver: bridge
    ipam:
      config:
        - subnet: 10.5.0.0/16
          gateway: 10.5.0.1

volumes:
  mount_path: New mount point
```

S4:Compose your file

```
lusy@localhost.localdomain ~/Docker5
docker-compose up

[+] Running 3/1
 ✓ Network docker5_network    Created
 ✓ Container mysql_cont        Created
 ✓ Container mattermost_cont   Created
Attaching to mattermost_cont, mysql_cont
mysql_cont    | [Entrypoint] MySQL Docker Image 8.0.32-1.2.1
mysql_cont    | [Entrypoint] Starting MySQL 8.0.32-1.2.11-se
mattermost_cont | {"timestamp":"2023-09-18 13:13:51.014 Z","le
mattermost_cont | {"timestamp":"2023-09-18 13:13:51.014 Z","le
mattermost_cont | {"timestamp":"2023-09-18 13:13:51.027 Z","le
mattermost_cont | {"timestamp":"2023-09-18 13:13:51.041 Z","le
mattermost_cont | {"timestamp":"2023-09-18 13:13:51.041 Z","le
server.go:306"}
mattermost_cont | {"timestamp":"2023-09-18 13:13:51.043 Z","le
```

S5: Verify your process is running or not

```
lusy@localhost.localdomain ~ (0.053s)
docker ps

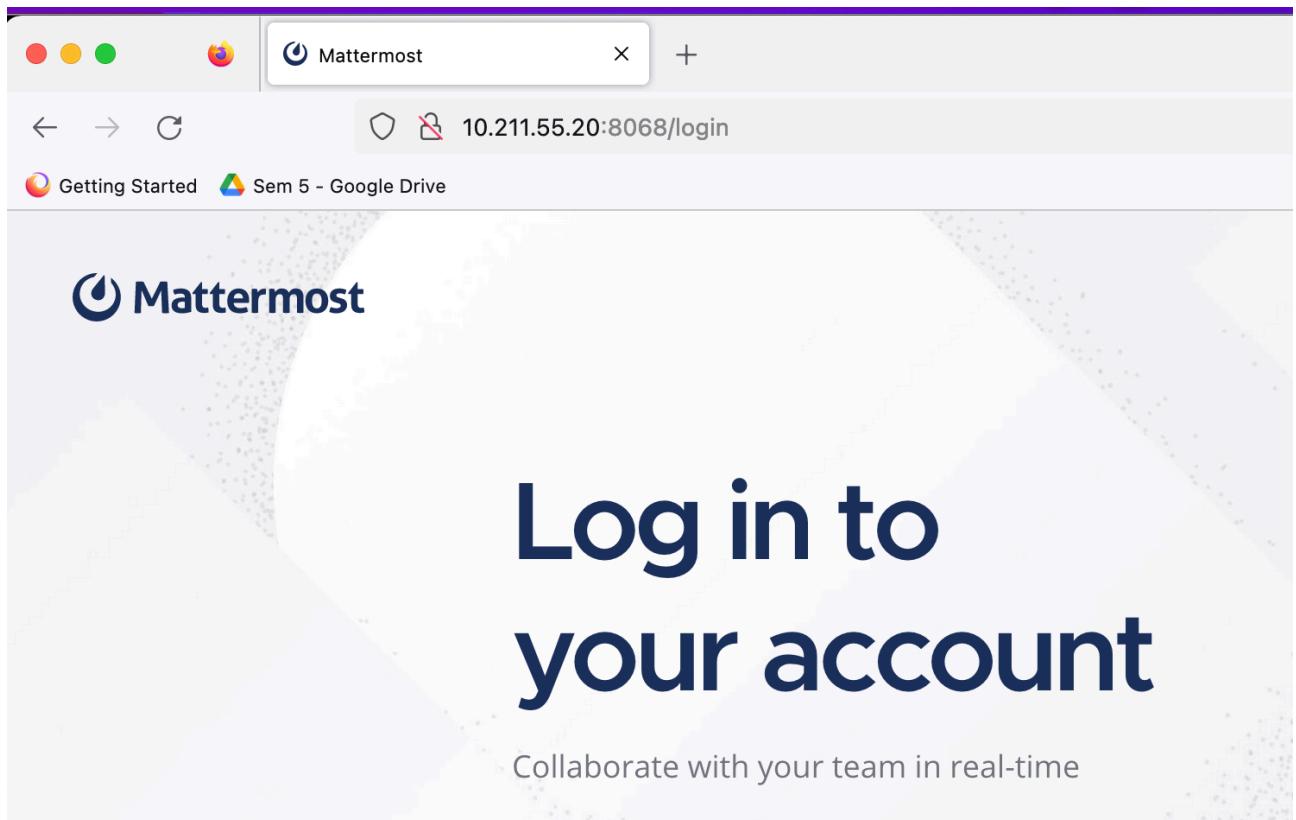
CONTAINER ID IMAGE COMMAND
NAMES
03b998af7840 kartik404/mattermost:arm "/opt/mattermost/bin...
mattermost_cont
1f27e2168082 kartik404/mysql:arm "/entrypoint.sh mysq...
tcp mysql_cont
ffb32d11e1de centos7_t "nginx '-g daemon off; ...
centos7_t
7a3bfc3d5b84 mysqld "/entrypoint.sh mysq...
tcp centos7
ede3a5b1e417 mattermost "/opt/mattermost/bin...
mattermost
7c44d8a7924b nginx:latest "/docker-entrypoint....
```

S6: Using your IP verify it (with Given port)

```
lusy@localhost.localdomain ~ (0.037s)
curl 172.17.0.1:8068

<!doctype html><html lang="en"><head><meta charset="utf-8"><meta ...
ts" content="noindex, nofollow"><meta name="referrer" content="...
-name" content="Mattermost"><meta name="format-detection" content="...
es="16x16"><link rel="icon" type="image/png" href="/static/images/fav...
con/favicon-default-32x32.png" sizes="32x32"><link rel="icon" type="...
age/png" href="/static/images/favicon/favicon-default-96x96.png" ...
vetica, Arial, sans-serif; padding-top: 50px; max-width: 750px; font-size: ...
ne-height: 1.2}.error-screen ul{padding-left: 15px; line-height: 1.1}. ...
r-top: 1px solid #eee}.error-screen-visible{display: block}</style>
<script defer="defer" src="/static/main.2eeb625aa194df7a2985.js">
le" content="yes" /><meta name="apple-mobile-web-app-status-bar-style" ...
l="apple-touch-icon" sizes="72x72" href="/static/icon_72x72.png" ...
es="57x57" href="/static/icon_57x57.png" /><link rel="apple-touch-icon" ...
static/icon_144x144.png" /><link rel="apple-touch-icon" sizes="144x144" ...
/>
```

S7: Verify in browser



S8 :Verify it

```
easy@localhost:~/dockers$ docker-compose up
[+] Running 2/0
 ✓ Container mysql_cont    Running
 ✓ Container mattermost_cont  Running
Attaching to mattermost_cont mysql_cont
```