Name: Shadwa Mohamed Elhady

# Docker Labs solution

## lab 0

### problem 1

- How do you run a Node.js application using Docker, using the official Node.js image?

```
sudo docker run -it node:18 node -v
```

##

- What command would you use to mount a local directory into a Node.js Docker container?

```
docker run -v ~/Desktop/Projects/nurserySystem:/app -it node:18
```

##

- What is the command to copy files from your local machine into a running Python Docker container and vice versa?

Copy file from local to container:

```
sudo docker cp ~/Desktop/Projects/test.txt f5adc3d17552:app/copied-file.txt
```

Copy file from container to local

```
sudo docker cp f5adc3d17552:app/copied-file.txt from-container.txt
```

##

- How can you execute a Python script inside a running Docker container?

```
sudo docker cp ~/Desktop/Projects/task.py f5adc3d17552:app/task.py
```

To execute a python script

```
sudo docker exec -it f5adc3d17552 python task.py
```

##

- How do you start a Nginx Docker container and expose it on port 80?

Dockerfile

```
# Use the official nginx base image
FROM nginx
# Copy the custom configuration file to the container
COPY nginx.conf /etc/nginx/nginx.conf
# Expose port 80
EXPOSE 80
```

sudo docker build -t first-ngin:v0.1 .
sudo docker run -d -p80:80 first-ngin:v0.1
###
- What command can you use to inspect the Nginx container's logs?
sudo docker logs f8c568a9a700
###
- How can you list all Docker images on your system?
sudo docker images
###
- What is the purpose of the docker ps command, and how can you see all containers, including stopped ones?
It is used to see the status of the process and it lists the running containers by default.
docker ps -a command to list all containers, including the stopped ones.
###
- How can you stop a running Docker container gracefully?
sudo docker container stop f5adc3d17552
###
- What command would you use to remove all stopped containers from your system?
docker container prune
###
- How can you view detailed information about a Docker image, including its layers?
sudo docker inspect getting-started
###
- What is the difference between a Docker image and a Docker container, and how do you create a container from an image?
Docker image is a template that defines how a container will be realized. A Docker container is a runtime instance of a Docker image. Docker images are immutable,In contrast, containers are mutable and allow modifications during runtime.
To create a container from an image
docker container create -i -t --name mycontainer alpine
docker container start --attach -i mycontainer
OR

```
docker run -it --name mycontainer2 alpine
```
##
##
## lab 1
### problem 1
- Run the container hello-world
```
docker run hello-world
```
##
- Check the container status
```
docker ps -a
```
##
- Start the stopped container
```
docker start hello-world
```
##
- Remove the container
```
docker container rm hello-world
```
##
- Remove the image
```
docker image rm hello-world
```
##
### problem 2
- Run container centos or ubuntu in an interactive mode
```
sudo docker pull ubuntu
sudo docker run -it ubuntu /bin/bash
```
##
- Run the following command in the container “echo docker ”
```
root@a3f74f970c91:/# echo “docker”
```
##
- Open a bash shell in the container and touch a file named hello-docker
```
touch hello-docker
```
- Stop the container and remove it. Write your comment about the file hello-docker
```
sudo docker stop a3f74f970c91
sudo docker rm a3f74f970c91
```
Comment about the file hello-docker: the file will be removed with the container because it exist inside it only.

##
### problem 3
- Run a container nginx with name nginx and attach a volume to the container

```
sudo docker run -d -v myvolume:/usr/share/nginx/html nginx
```

##

- Volume for containing static html file

```
sudo docker volume create html_volume
```

##

- Remove the container

```
sudo docker stop e4bcb38047b6
sudo docker rm e4bcb38047b6
```

##

- Run a new container with the following:
- Attach the volume that was attached to the previous container
- Map port 80 to port 9898 on you host machine

```
sudo docker run -d -v html_volume:/usr/share/nginx/html nginx
sudo docker exec -it 190cba76b7f6 bash
root@190cba76b7f6:/# cd /usr/share/nginx/html
root@190cba76b7f6:/usr/share/nginx/html# ls
50x.html  index.html
root@190cba76b7f6:/usr/share/nginx/html# touch contact.html
root@190cba76b7f6:/usr/share/nginx/html# ls
50x.html  contact.html   index.html
root@190cba76b7f6:/usr/share/nginx/html# echo "Hello from index" >
index.html
root@190cba76b7f6:/usr/share/nginx/html# cat index.html
Hello from index
root@190cba76b7f6:/usr/share/nginx/html# exit
exit
shadwa@shadwa-device:~/Desktop/Projects/Docker lec/Learn-Docker$
sudo docker stop 190cba76b7f6
190cba76b7f6
sudo docker run -d -v html_volume:/usr/share/nginx/html -p9898:80
nginx
```

##

- Access the html files from your browser:

open the browser and open :http://localhost:9898/
##
### problem 4

- Run the image nginx again without attaching any volumes

```
sudo docker run -d --name nginx nginx
```
##
- Add html static files to the container and make sure they are accessible

```
sudo docker cp ~/Desktop/Projects/contact.html
b209146d18c6:usr/share/nginx/html/contact.html

sudo docker cp ~/Desktop/Projects/pricing.html
b209146d18c6:usr/share/nginx/html/pricing.html

sudo docker exec -it b209146d18c6 bash
root@b209146d18c6:/# cd usr/share/nginx/html/
root@b209146d18c6:/usr/share/nginx/html# ls
50x.html  contact.html   index.html  pricing.html
```
##
- Commit the container with image name my nginx

```
sudo docker commit b209146d18c6 my_nginx
```
##
- Create a dockerfile for ngnix and build the image from this dockerfile

Dockerfile

```
# Use the official nginx base image
FROM nginx
COPY /Desktop/Projects/index2.html /usr/share/nginx/html
# Expose port 80
EXPOSE 80
sudo docker build -t my_nginx .
```
##
### problem 5

- Create a volume called mysql_data, then:

```
sudo docker volume create mysql_data
```
##
- deploy a MySQL database called app-database.

- use the mysql latest image
- use the -e flag to set MYSQL_ROOT_PASSWORD to P4sSw0rd0!.
- mount the mysql_data volume to /var/lib/mysql.
- the container should run in the background.

```
sudo docker run -d --name app-database -e
MYSQL_ROOT_PASSWORD=P4sSw0rd0! -v mysql_data:/var/lib/mysql
mysql:latest
```

##
##
## lab 2
### problem 1
- Create your own nginx docker image based on ubuntu “NEVER USE FROM nginx
- Install nginx
- index.html one as file
- Expose
- Start
- Port mapping

Dockerfile

```
# Use the official nginx base image
FROM ubuntu
RUN apt-get -y update && apt-get -y install nginx
COPY index.html /var/www/html/
# Expose port 80
EXPOSE 80
CMD ["nginx", "-g", "daemon off;"]
```

```
sudo docker build -t my-nginx-docker .
sudo docker run -d -p3100:80 my-nginx-docker
```

##
### problem 2
- Create react app docker container "using Dockerfile"

```
sudo npx create-react-app react-project
cd react-project
```

Dockerfile

```
# Fetching the latest node image on alpine linux
```

```
FROM node:alpine AS development
# Declaring env
ENV NODE_ENV development

# Setting up the work directory
WORKDIR /react-app
# Installing dependencies
COPY ./package.json /react-app
RUN npm install
# Copying all the files in our project
COPY . .
# Starting our application
CMD npm start
```

```
sudo docker build -t react-project .
sudo docker run -p3400:3000 react-project
```

##

### problem 3

- Create flask app to count number of visits to browser:

- Create new directory called flask then add app.py and requirements.txt files

app.py

```
from flask import Flask
from redis import Redis
app = Flask(__name__)
redis = Redis(host='redis', port=6379)
@app.route('/')
def hello():
redis.incr('visits')
visits = redis.get('visits').decode('utf-8')
return f'Number of visits: {visits}'
if __name__ == '__main__':
app.run(host='0.0.0.0')
```

requirements.txt

```
Flask
redis
```

##

- Create Dockerfile for the python app

Dockerfile

```
FROM python:3.9
WORKDIR /app
COPY requirements.txt .
RUN pip install --no-cache-dir -r requirements.txt
COPY app.py .
CMD ["python", "app.py"]
```

##

- Create docker-compose for the app and use Redis as temp DB.

docker-compose.yml

```
version: "3"
services:
  web:
    build:
      context: .
      dockerfile: Dockerfile
    ports:
      - 5000:5000
    depends_on:
      - redis
  redis:
    image: redis
```

```
sudo curl -L https://github.com/docker/compose/releases/download/1.21.0/docker-compose-$(uname -s)-$(uname -m) -o /usr/local/bin/docker-compose

sudo chmod +x /usr/local/bin/docker-compose

sudo docker-compose up
```