NAME: RIMA P

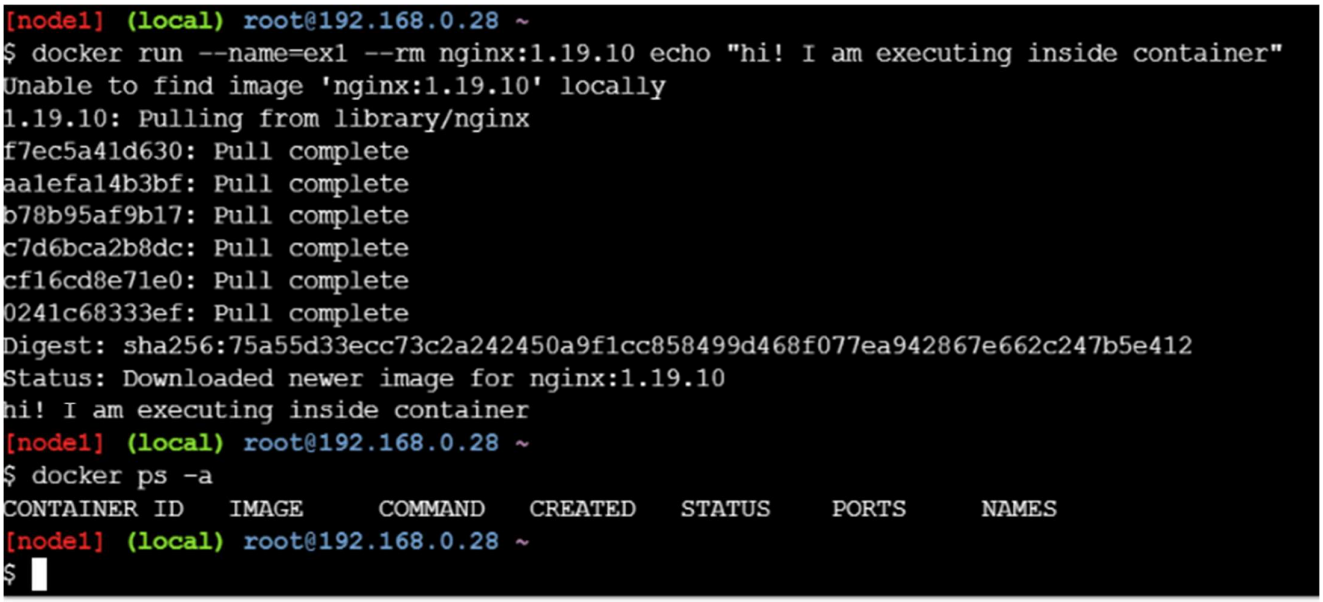
DOCKER EXERCISES:

1. Spin up a temporary container with image nginx:1.19.10 and execute inside it, such that the container should be destroyed, once you exit from the container

ANS:

docker run --name=ex1 --rm nginx:1.19.10 echo "hi! I am executing inside container"

SCREENSHOT:

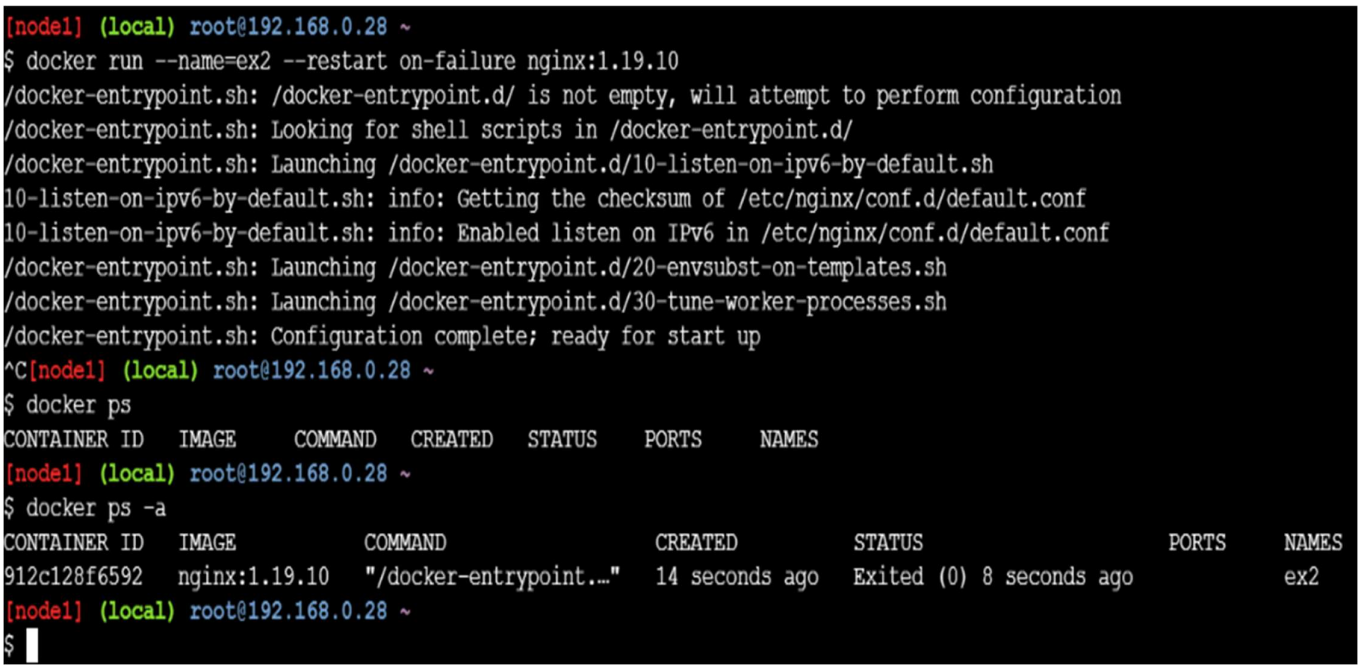


2. Spin up a container with image nginx:1.19.10 such that it should restart automatically if any fatal errors are encountered

ANS:

docker run --name=ex2 --restart on-failure nginx:1.19.10

SCREENSHOT:



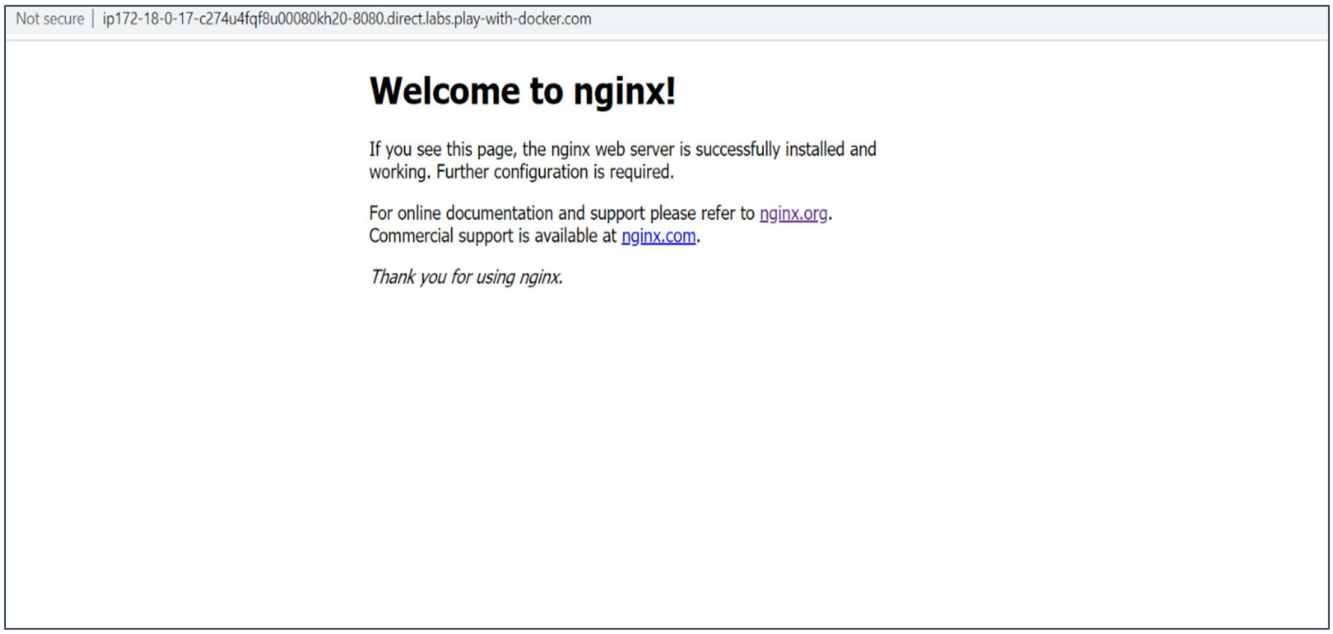
3. Spin up a container with image nginx:1.19.10 such that port 80 of the container can be connected from port 8080 of the host

ANS:

docker run --name=ex3 -p 8080:80 nginx:1.19.10

SCREENSHOTS:

```
[node1] (local) root@192.168.0.28 ~
$ docker run --name=ex3 -p 8080:80 nginx:1.19.10
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
172.18.0.1 - - [02/May/2021:07:10:12 +0000] "GET / HTTP/1.1" 200 612 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/90.0.4430.93 Safari/537.36" "-"
2021/05/02 07:10:14 [error] 32#32: *2 open() "/usr/share/nginx/html/favicon.ico" failed (2: No such file or directory), client: 172.18.
0.1, server: localhost, request: "GET /favicon.ico HTTP/1.1", host: "ip172-18-0-17-c274u4fqf8u00080kh20-8080.direct.labs.play-with-dock
er.com", referrer: "http://ip172-18-0-17-c274u4fqf8u00080kh20-8080.direct.labs.play-with-docker.com/"
172.18.0.1 - - [02/May/2021:07:10:14 +0000] "GET /favicon.ico HTTP/1.1" 404 556 "http://ip172-18-0-17-c274u4fqf8u00080kh20-8080.direct.
labs.play-with-docker.com/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/90.0.4430.93 Safar
i/537.36" "-"
172.18.0.1 - - [02/May/2021:07:10:14 +0000] "GET / HTTP/1.1" 200 612 "http://ip172-18-0-17-c274u4fqf8u00080kh20-8080.direct.labs.play-w
ith-docker.com/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/90.0.4430.93 Safari/537.36" "
-"
```



4. Spin up a container with image redis:6.2.2 and mount volume /mnt/redis-data of host to the /data of the container

ANS:

Using -volume:
docker run --name=ex4Volume -d -v /mnt/redis-data:/data redis:6.2.2

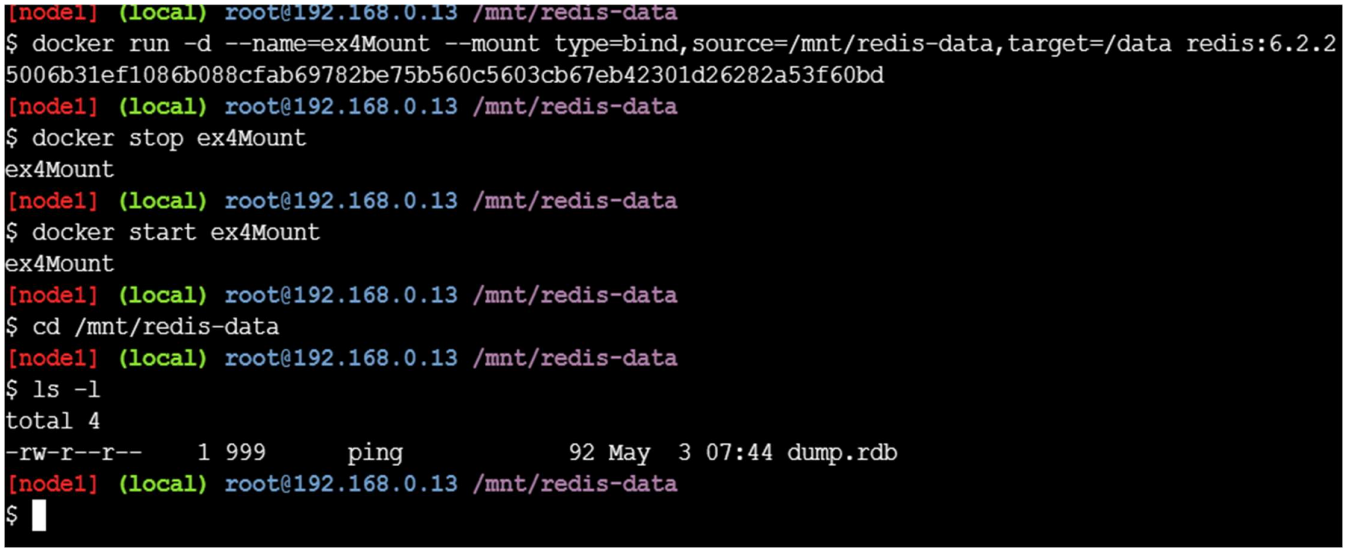
SCREENSHOT:

```
[node1] (local) root@192.168.0.13 ~
$ docker run --name=ex4Volume -d -v /mnt/redis-data:/data redis:6.2.2
Unable to find image 'redis:6.2.2' locally
6.2.2: Pulling from library/redis
f7ec5a41d630: Pull complete
a36224ca8bbd: Pull complete
7630ad34dcb2: Pull complete
c6d2a5632e6c: Pull complete
f1957981f3c1: Pull complete
42642d666cff: Pull complete
Digest: sha256:e10f55f92478715698a2cef97c2bbdc48df2a05081edd884938903aa60df6396
Status: Downloaded newer image for redis:6.2.2
dc0abdad8c5f6ce775378b617a38dbdcc43c194013a6cf1d6fafbf63081fcfea
[node1] (local) root@192.168.0.13 ~
$ cd /mnt/redis-data
[node1] (local) root@192.168.0.13 /mnt/redis-data
$ ls -l
total 0
[node1] (local) root@192.168.0.13 /mnt/redis-data
$ docker stop ex4Volume
ex4Volume
[node1] (local) root@192.168.0.13 /mnt/redis-data
$ docker start ex4Volume
ex4Volume
[node1] (local) root@192.168.0.13 /mnt/redis-data
$ ls -l
total 4
-rw-r--r--    1 999      ping            92 May  3 07:37 dump.rdb
```

Using --mount:

docker run --name=ex4Mount -d --mount type=bind,source=/mnt/redis-data,target=/data redis:6.2.2

SCREENSHOT:



5. Create a dockerfile with base image centos:7 , and build an image with any sample application file

ANS:

vi Dockerfile:

```
FROM centos:7
RUN yum -y update && yum -y install epel-release
RUN yum install -y python python-pip && pip install flask
COPY ./app.py /app.py
ENTRYPOINT FLASK_APP=/app.py flask run –host=0.0.0.0
```

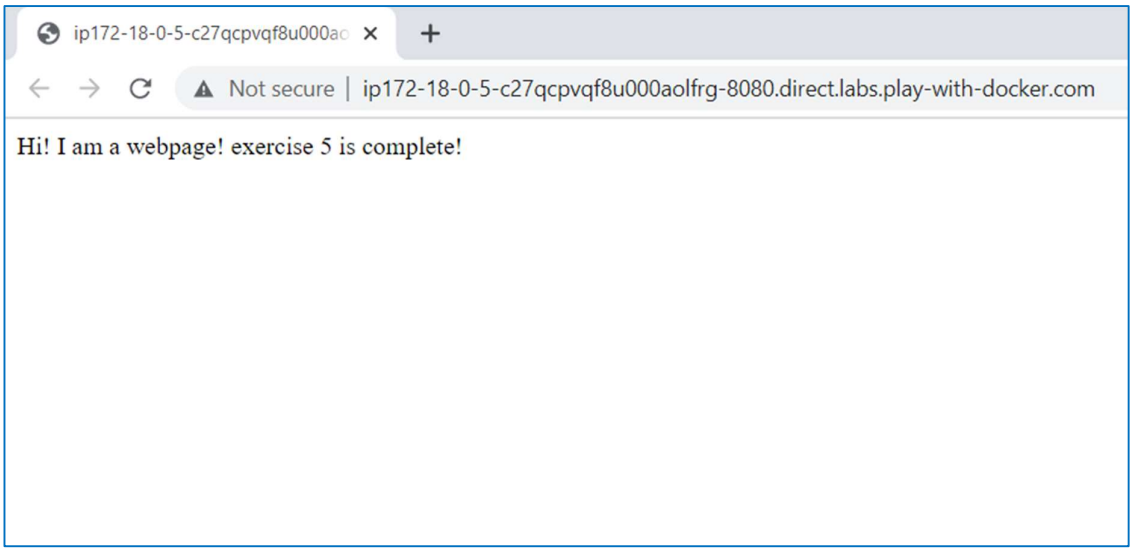
vi app.py

```
from flask import Flask
app=Flask(__name__)
@app.route("/")
def display():
    return "Hi! I am a webpage! exercise 5 is complete!"
```

```
docker build . -f Dockerfile -t ex5_im
```

```
docker run -p 8080:5000 -d --name=ex5_ct ex5_im
```

SCREENSHOTS:



```
[node1] (local) root@192.168.0.13 ~
$ more Dockerfile
FROM centos:7
RUN yum -y update && yum -y install epel-release
RUN yum install -y python python-pip && pip install flask
COPY ./app.py /app.py
ENTRYPOINT FLASK_APP=/app.py flask run --host=0.0.0.0
[node1] (local) root@192.168.0.13 ~
$ more app.py
from flask import Flask
app=Flask(__name__)
@app.route("/")
def display():
  return "Hi! I am a webpage! exercise 5 is complete!"
[node1] (local) root@192.168.0.13 ~
```

```
[node1] (local) root@192.168.0.13 ~
$ docker build . -f Dockerfile -t ex5_im
Sending build context to Docker daemon     47MB
Step 1/5 : FROM centos:7
 ---> 8652b9f0cb4c
Step 2/5 : RUN yum -y update && yum -y install epel-release
 ---> Using cache
 ---> d06e9740bb76
Step 3/5 : RUN yum install -y python python-pip && pip install flask
 ---> Using cache
 ---> 401c61018498
Step 4/5 : COPY ./app.py /app.py
 ---> Using cache
 ---> 807d93a3d985
Step 5/5 : ENTRYPOINT FLASK_APP=/app.py flask run --host=0.0.0.0
 ---> Using cache
 ---> 31b4aff5c3fd
Successfully built 31b4aff5c3fd
Successfully tagged ex5_im:latest
[node1] (local) root@192.168.0.13 ~
$ docker run -p 8080:5000 -d --name=ex5_ct ex5_im
a0d8d48aec29f830df573d84e851f02e8f8a8100fe519a84d20e7b92fdc6ca1d
[node1] (local) root@192.168.0.13 ~
```

6. Create a bridge network called test-app and spin up nginx and redis containers in that network

ANS:

docker network create --driver bridge test-app

vi docker-compose.yml

```
version: "3.3"
services:
  ngin_cont:
    image: nginx
    ports:
    - 8080:80
  redis_cont:
    image: redis:6.2.2
    ports:
    - 8081:6379
networks:
  default:
    external:
      name: test-app
```

docker-compose up -d

docker network inspect test-app

SCREENSHOTS:

```
[node1] (local) root@192.168.0.28 ~
$ docker network create --driver bridge test-app
aa19f8c3bbacc7734850820f422ee3c75b99e8a70358a35494d8ca6e331fd028
[node1] (local) root@192.168.0.28 ~
$ docker network ls
NETWORK ID     NAME       DRIVER    SCOPE
0a27a35e460d   bridge     bridge    local
3489a951cfdd   host       host      local
d8c40c6f233d   none       null      local
aa19f8c3bbac   test-app   bridge    local
[node1] (local) root@192.168.0.28 ~
```

```
[node1] (local) root@192.168.0.28 ~
$ vi docker-compose.yml
[node1] (local) root@192.168.0.28 ~
$ more docker-compose.yml
version: "3.3"
services:
  ngin_cont:
    image: nginx
    ports:
    - 8080:80
  redis_cont:
    image: redis:6.2.2
    ports:
    - 8081:6379
networks:
  default:
    external:
      name: test-app
[node1] (local) root@192.168.0.28 ~
```

```
[node1] (local) root@192.168.0.28 ~
$ docker-compose up -d
Starting root_ngin_cont_1  ... done
Starting root_redis_cont_1 ... done
[node1] (local) root@192.168.0.28 ~
$ docker network inspect test-app
[
    {
        "Name": "test-app",
        "Id": "aa19f8c3bbacc7734850820f422ee3c75b99e8a70358a35494d8ca6e331fd028",
        "Created": "2021-05-03T08:29:26.917620281Z",
        "Scope": "local",
        "Driver": "bridge",
        "EnableIPv6": false,
        "IPAM": {
            "Driver": "default",
            "Options": {},
            "Config": [
                {
                    "Subnet": "172.19.0.0/16",
                    "Gateway": "172.19.0.1"
                }
            ]
```

```
},
"ConfigOnly": false,
"Containers": {
    "4d73a542d47e6df4792ef9d2fb322da50d27e694bad420de513a997891d21e5d": {
        "Name": "root_ngin_cont_1",
        "EndpointID": "ad3cbedc0275e00a32bc3d4ad0a55d5b01e8abf67094ab05c744c9e43fc9b
        "MacAddress": "02:42:ac:13:00:02",
        "IPv4Address": "172.19.0.2/16",
        "IPv6Address": ""
    },
    "d960b072e6e6c5c5a3400f4ba7f1ee33a8699f4712f1cd21c252993041359eab": {
        "Name": "root_redis_cont_1",
        "EndpointID": "37c1169dd3bd7ef55393128871536a5ae31e41094f94f82b520fb232ea7dc
        "MacAddress": "02:42:ac:13:00:03",
        "IPv4Address": "172.19.0.3/16",
        "IPv6Address": ""
    }
},
"Options": {},
"Labels": {}
```


Result:

The exercises using docker are successfully executed.